Computer Algebra Independent Integration Tests

Summer 2023 edition

1-Algebraic-functions/1.2-Trinomial-products/1.2.2-Quartic/42-1.2.2.5-P-x-a+b-x^2+c-x^4-^p

Nasser M. Abbasi

September 6, 2023 Compiled on September 6, 2023 at 1:13am

Contents

CHAPTER 1

INTRODUCTION

This report gives the result of running the computer algebra independent integration test.The download section in on the main webpage contains links to download the problems in plain text format used for all CAS systems. The number of integrals in this report is [111]. This is test number [42].

1.1 Listing of CAS systems tested

The following are the CAS systems tested:

1. Mathematica 13.3.1 (August 16, 2023) on windows 10.
2. Rubi 4.16.1 (Dec 19, 2018) on Mathematica 13.3 on windows 10
3. Maple 2023.1 (July, 12, 2023) on windows 10.
4. Maxima 5.47 (June 1, 2023) using Lisp SBCL 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
5. FriCAS 1.3.9 (July 8, 2023) based on sbcl 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
6. Giac/Xcas 1.9.0-57 (June 26, 2023) on Linux via sagemath 10.1 (Aug 20, 2023).
7. Sympy 1.12 (May 10, 2023) Using Python 3.11.3 on Linux.
8. Mupad using Matlab 2021a with Symbolic Math Toolbox Version 8.7 on windows 10.

Maxima and Fricas and Giac are called using Sagemath. This was done using Sagemath `integrate` command by changing the name of the algorithm to use the different CAS systems.

Sympy was run directly in Python not via sagemath.

1.2 Results

Important note: A number of problems in this test suite have no antiderivative in closed form. This means the antiderivative of these integrals can not be expressed in terms of elementary, special functions or `Hypergeometric2F1` functions. `RootSum` and `RootOf` are not allowed. If a CAS returns the above integral unevaluated within the time limit, then the result is counted as passed and assigned an A grade.

However, if CAS times out, then it is assigned an F grade even if the integral is not integrable, as this implies CAS could not determine that the integral is not integrable in the time limit.

If a CAS returns an antiderivative to such an integral, it is assigned an A grade automatically and this special result is listed in the introduction section of each individual test report to make it easy to identify as this can be important result to investigate.

The results given in in the table below reflects the above.

System	% solved	% Failed
Rubi	100.00 (111)	0.00 (0)
Mathematica	100.00 (111)	0.00 (0)
Maple	100.00 (111)	0.00 (0)
Mupad	95.50 (106)	4.50 (5)
Giac	95.50 (106)	4.50 (5)
Fricas	83.78 (93)	16.22 (18)
Maxima	74.77 (83)	25.23 (28)
Sympy	40.54 (45)	59.46 (66)

Table 1.1: Percentage solved for each CAS

The table below gives additional break down of the grading of quality of the antiderivatives generated by each CAS. The grading is given using the letters A,B,C and F with A being the best quality. The grading is accomplished by comparing the antiderivative generated with the optimal antiderivatives included in the test suite. The following table describes the meaning of these grades.

grade	description
A	Integral was solved and antiderivative is optimal in quality and leaf size.
B	Integral was solved and antiderivative is optimal in quality but leaf size is larger than twice the optimal antiderivatives leaf size.
C	Integral was solved and antiderivative is non-optimal in quality. This can be due to one or more of the following reasons 1. antiderivative contains a hypergeometric function and the optimal antiderivative does not. 2. antiderivative contains a special function and the optimal antiderivative does not. 3. antiderivative contains the imaginary unit and the optimal antiderivative does not.
F	Integral was not solved. Either the integral was returned unevaluated within the time limit, or it timed out, or CAS hanged or crashed or an exception was raised.

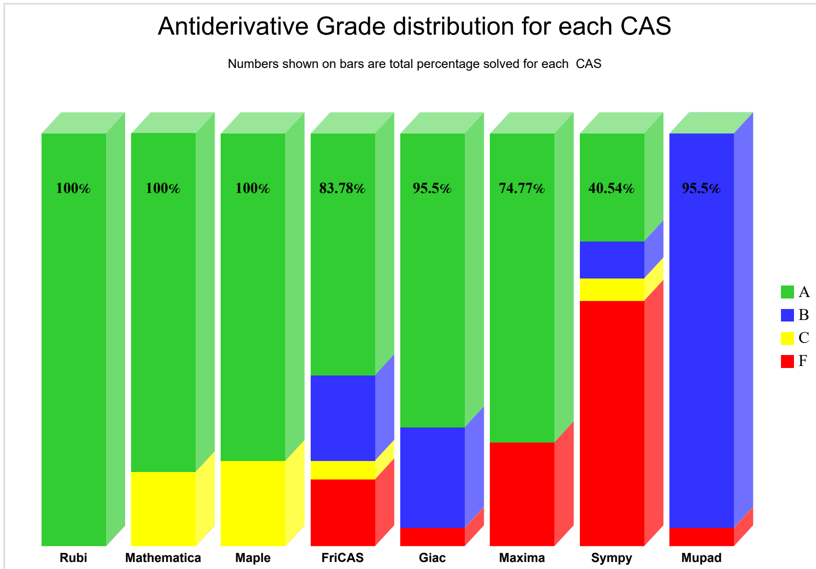
Table 1.2: Description of grading applied to integration result

Grading is implemented for all CAS systems. Based on the above, the following table summarizes the grading for this test suite.

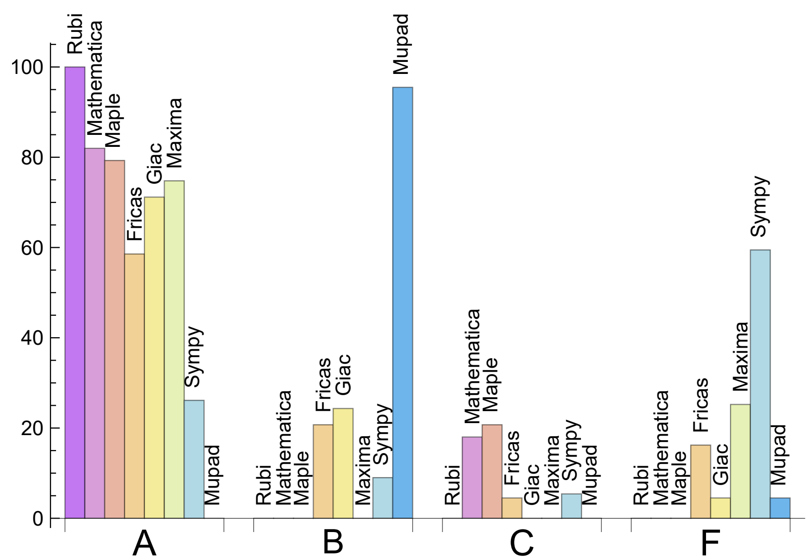
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.000	0.000	0.000	0.000
Mathematica	81.982	0.000	18.018	0.000
Maple	79.279	0.000	20.721	0.000
Maxima	74.775	0.000	0.000	25.225
Giac	71.171	24.324	0.000	4.505
Fricas	58.559	20.721	4.505	16.216
Sympy	26.126	9.009	5.405	59.459
Mupad	0.000	95.495	0.000	4.505

Table 1.3: Antiderivative Grade distribution of each CAS

The following is a Bar chart illustration of the data in the above table.



The figure below compares the grades of the CAS systems.



The following table shows the distribution of the different types of failures for each CAS. There are 3 types failures. The first is when CAS returns the input within the time limit, which means it could not solve it. This is the typical failure and given as **F**.

The second failure is due to time out. CAS could not solve the integral within the 3 minutes time limit which is assigned. This is assigned **F(-1)**.

The third is due to an exception generated, indicated as **F(-2)**. This most likely indicates an interface problem between sagemath and the CAS (applicable only to FriCAS, Maxima and Giac) or it could be an indication of an internal error in the CAS itself. This type of error requires more investigation to determine the cause.

System	Number failed	Percentage normal failure	Percentage timeout failure	Percentage exception failure
Rubi	0	0.00	0.00	0.00
Mathematica	0	0.00	0.00	0.00
Maple	0	0.00	0.00	0.00
Mupad	5	0.00	100.00	0.00
Giac	5	100.00	0.00	0.00
Fricas	18	0.00	100.00	0.00
Maxima	28	100.00	0.00	0.00
Sympy	66	12.12	87.88	0.00

Table 1.4: Failure statistics for each CAS

1.3 Time and leaf size Performance

The table below summarizes the performance of each CAS system in terms of time used and leaf size of results.

Mean size is the average leaf size produced by the CAS (before any normalization). The Normalized mean is relative to the mean size of the optimal anti-derivative given in the input files.

For example, if CAS has **Normalized mean** of 3, then the mean size of its leaf size is 3 times as large as the mean size of the optimal leaf size.

Median size is value of leaf size where half the values are larger than this and half are smaller (before any normalization). i.e. The Middle value.

Similarly the **Normalized median** is relative to the median leaf size of the optimal.

For example, if a CAS has Normalized median of 1.2, then its median is 1.2 as large as the median leaf size of the optimal.

System	Mean time (sec)
Maxima	0.21
Rubi	0.43
Giac	0.76
Maple	0.93
Mathematica	1.35
Mupad	6.07
Fricas	7.07
Sympy	10.70

Table 1.5: Time performance for each CAS

System	Mean size	Normalized mean	Median size	Normalized median
Maxima	100.89	0.88	88.00	0.87
Maple	197.41	0.90	107.00	0.93
Rubi	217.75	1.00	140.00	1.00
Mathematica	260.32	1.08	146.00	1.02
Sympy	620.93	6.32	122.00	1.19
Giac	1853.60	3.90	124.00	0.98
Mupad	5695.50	9.58	132.00	1.00
Fricas	61061.85	246.12	141.00	1.27

Table 1.6: Leaf size performance for each CAS

1.4 Performance based on number of rules Rubi used

This section shows how each CAS performed based on the number of rules Rubi needed to solve the same integral. One diagram is given for each CAS.

On the y axis is the percentage solved which Rubi itself needed the number of rules given the x axis. These plots show that as more rules are needed then most CAS system percentage of solving decreases which indicates the integral is becoming more complicated to solve.

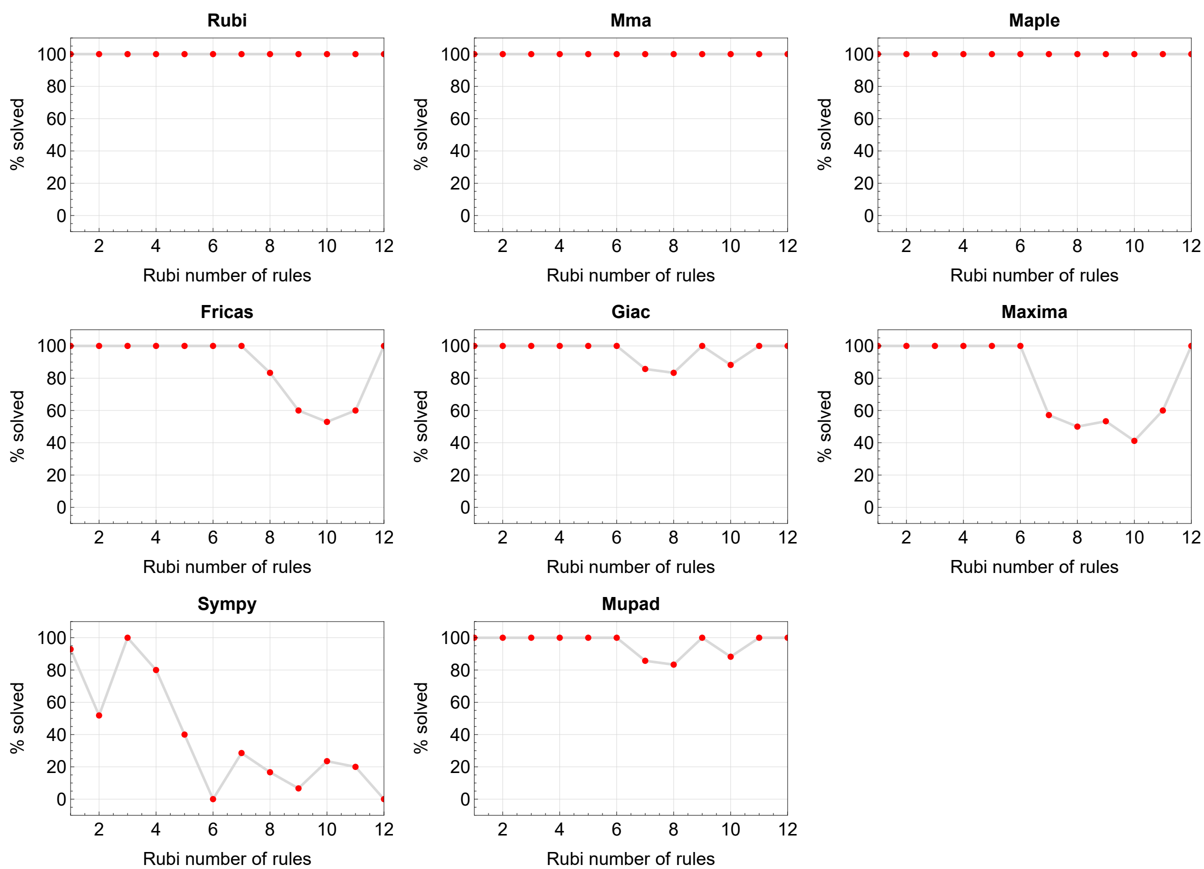


Figure 1.1: Solving statistics per number of Rubi rules used

1.5 Performance based on number of steps Rubi used

This section shows how each CAS performed based on the number of steps Rubi needed to solve the same integral. Note that the number of steps Rubi needed can be much higher than the number of rules, as the same rule could be used more than once.

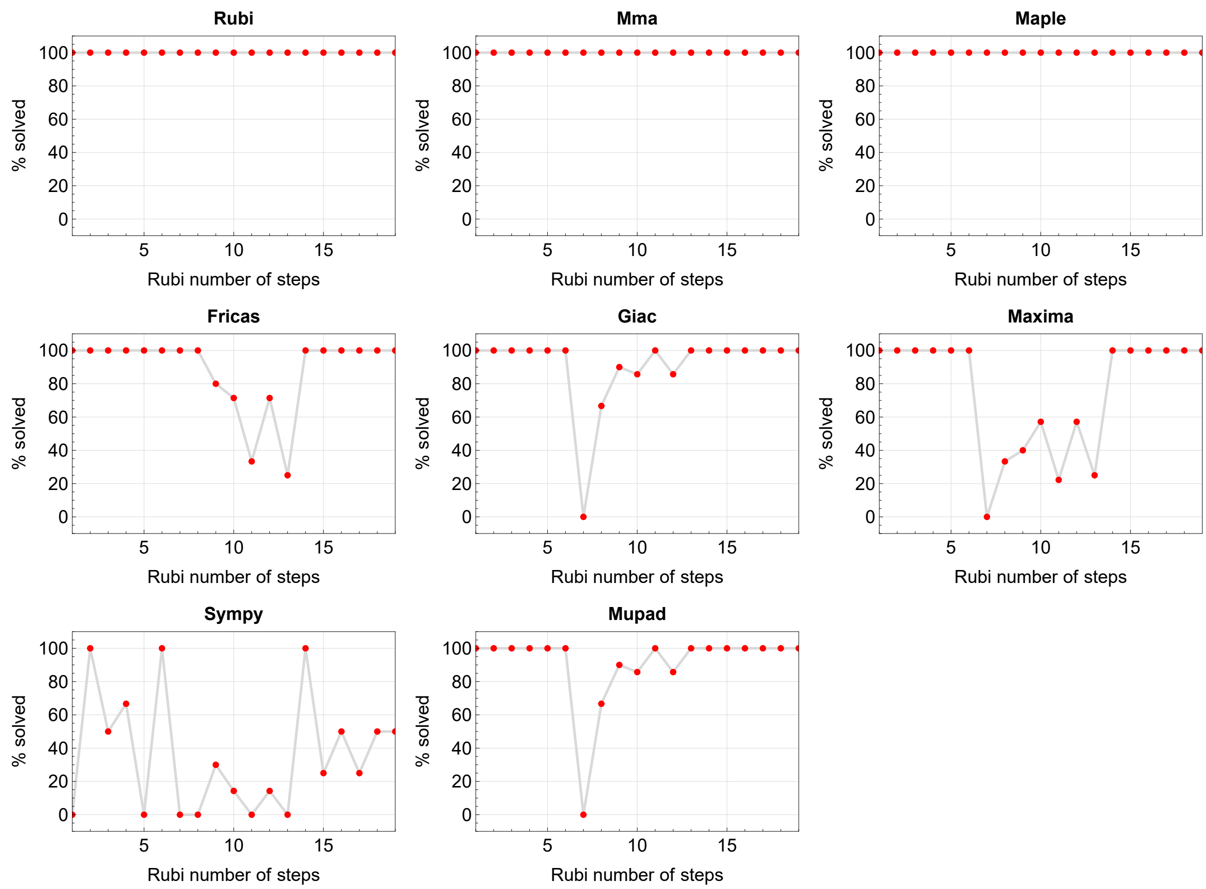


Figure 1.2: Solving statistics per number of Rubi steps used

The above diagram show that the precentage of solved intergals decreases for most CAS systems as the number of steps increases. As expected, for integrals that required less steps by Rubi, CAS systems had more success which indicates the integral was not as hard to solve. As Rubi needed more steps to solve the integral, the solved percentage decreased for most CAS systems which indicates the integral is becoming harder to solve.

1.6 Solved integrals histogram based on leaf size of result

The following shows the distribution of solved integrals for each CAS system based on leaf size of the antiderivatives produced by each CAS. It shows that most integrals solved produced leaf size less than about 100 to 150. The bin size used is 40.

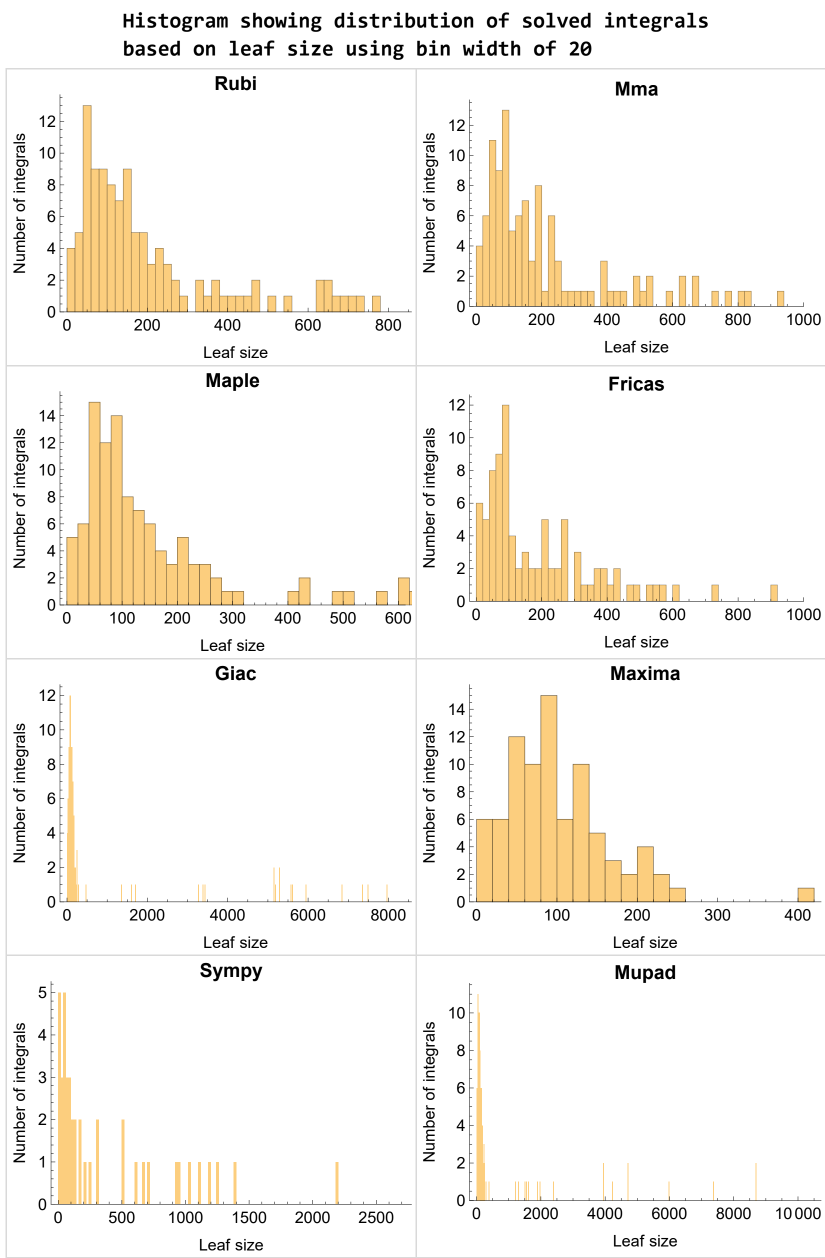


Figure 1.3: Solved integrals based on leaf size distribution

1.7 Solved integrals histogram based on CPU time used

The following shows the distribution of solved integrals for each CAS system based on CPU time used in seconds. The bin size used is 0.1 second.

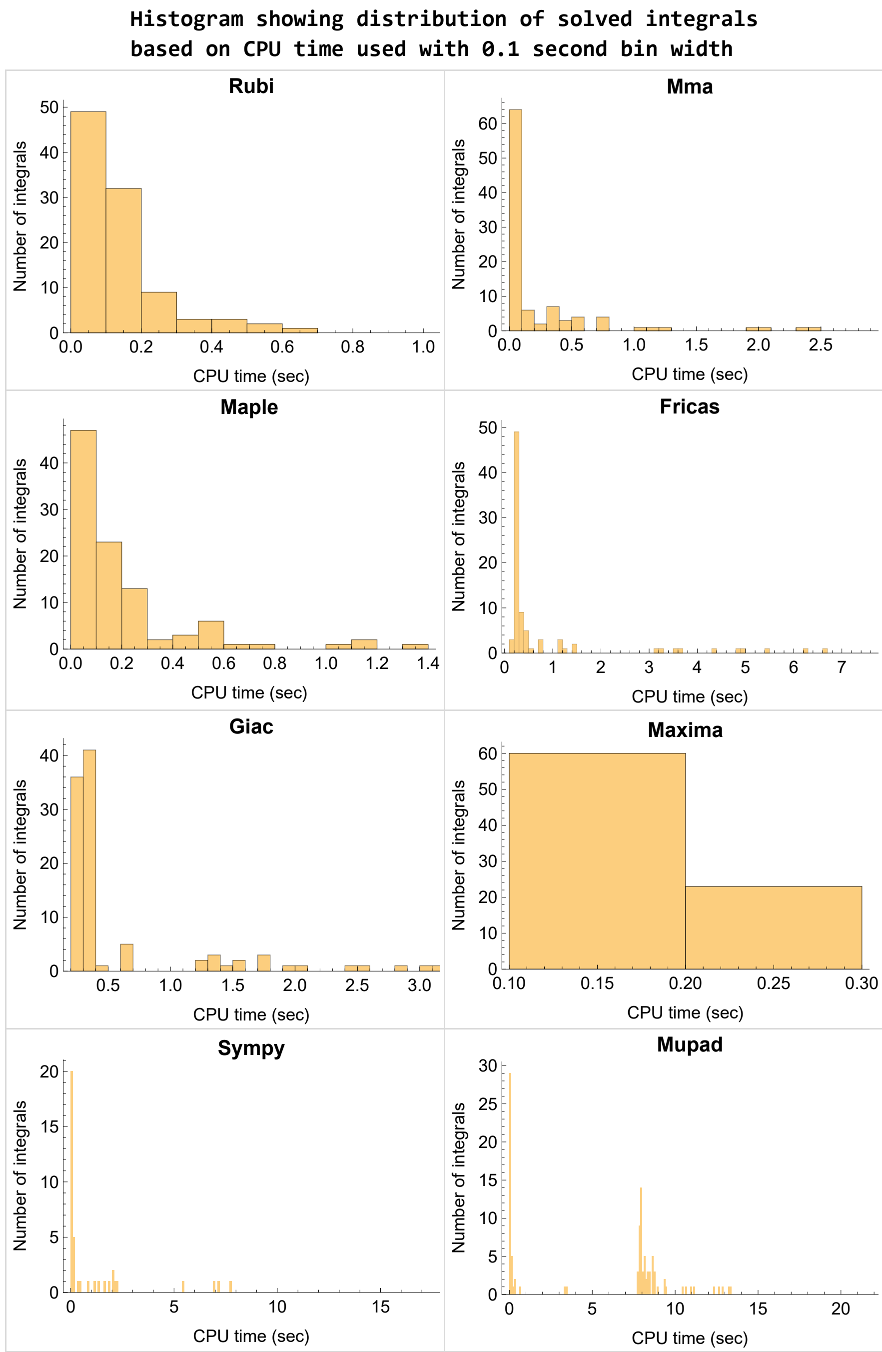


Figure 1.4: Solved integrals histogram based on CPU time used

1.8 Leaf size vs. CPU time used

The following shows the relation between the CPU time used to solve an integral and the leaf size of the antiderivative.

The result for Fricas, Maxima and Giac is shifted more to the right than the other CAS system due to the use of sagemath to call them, which causes an initial slight delay in the timing to start the integration due to overhead of starting a new process each time. This should also be taken into account when looking at the timing of these three CAS systems. Direct calls not using sagemath would result in faster timings, but current implementation uses sagemath as this makes testing much easier to do.

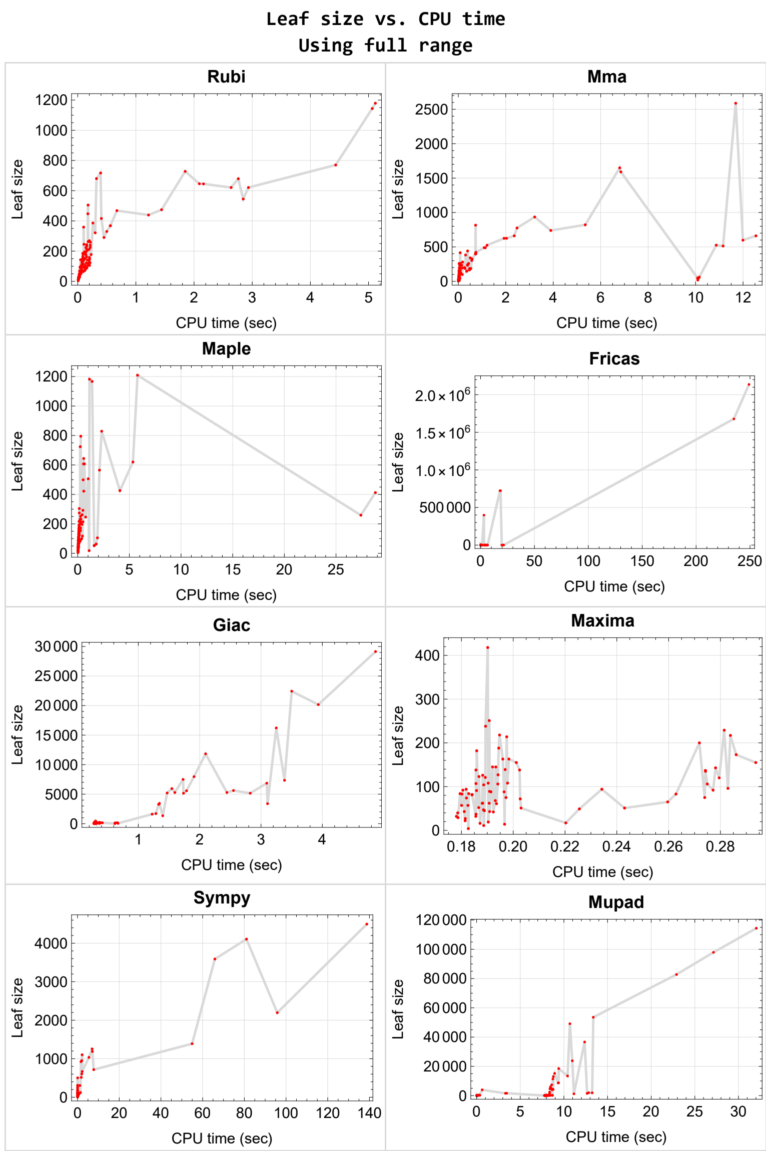


Figure 1.5: Leaf size vs. CPU time. Full range

1.9 list of integrals with no known antiderivative

{}

1.10 List of integrals solved by CAS but has no known antiderivative

Rubi {}

Mathematica {}

Maple {}

Maxima {}

Fricas {}

Sympy {}

Giac {}

Mupad {}

1.11 list of integrals solved by CAS but failed verification

The following are integrals solved by CAS but the verification phase failed to verify the anti-derivative produced is correct. This does not necessarily mean that the anti-derivative is wrong as additional methods of verification might be needed, or more time is needed (3 minutes time limit was used). These integrals are listed here to make it possible to do further investigation to determine why the result could not be verified.

Rubi {}

Mathematica {}

Maple {}

Maxima Verification phase not currently implemented.

Fricas Verification phase not currently implemented.

Sympy Verification phase not currently implemented.

Giac Verification phase not currently implemented.

Mupad Verification phase not currently implemented.

1.12 Timing

The command `AbsoluteTiming[]` was used in Mathematica to obtain the elapsed time for each integrate call. In Maple, the command `Usage` was used as in the following example

```
cpu_time := Usage(assign ('result_of_int',int(expr,x)),output='realtime'
```

For all other CAS systems, the elapsed time to complete each integral was found by taking the difference between the time after the call completed from the time before the call was made. This was done using Python's `time.time()` call.

All elapsed times shown are in seconds. A time limit of 3 CPU minutes was used for each integral. If the integrate command did not complete within this time limit, the integral was aborted and considered to have failed and assigned an F grade. The time used by failed integrals due to time out was not counted in the final statistics.

1.13 Verification

A verification phase was applied on the result of integration for `Rubi` and `Mathematica`.

Future version of this report will implement verification for the other CAS systems. For the integrals whose result was not run through a verification phase, it is assumed that the antiderivative was correct.

Verification phase also had 3 minutes time out. An integral whose result was not verified could still be correct, but further investigation is needed on those integrals. These integrals were marked in the summary table below and also in each integral separate section so they are easy to identify and locate.

1.14 Important notes about some of the results

Important note about Maxima results

Since tests were run in a batch mode, and using an automated script, then any integral where Maxima needed an interactive response from the user to answer a question during the evaluation of the integral will fail.

The exception raised is `ValueError`. Therefore Maxima results is lower than what would result if Maxima was run directly and each question was answered correctly.

The percentage of such failures were not counted for each test file, but for an example, for the `Timofeev` test file, there were about 14 such integrals out of total 705, or about 2 percent. This percentage can be higher or lower depending on the specific input test file.

Such integrals can be identified by looking at the output of the integration in each section for Maxima. The exception message will indicate the cause of error.

Maxima integrate was run using SageMath with the following settings set by default

```
'besselexpand : true'
'display2d : false'
'domain : complex'
'keepfloat : true'
'load(to_poly_solve)'
'load(simplify_sum)'
'load(abs_integrate)' 'load(diag)'
```

SageMath automatic loading of Maxima `abs_integrate` was found to cause some problems. So the following code was added to disable this effect.

```
from sage.interfaces.maxima_lib import maxima_lib
maxima_lib.set('extra_definite_integration_methods', '[]')
maxima_lib.set('extra_integration_methods', '[]')
```

See `https://ask.sagemath.org/question/43088/integrate-results-that-are-different-from-using-maxima/` for reference.

Important note about FriCAS result

There were few integrals which failed due to SageMath interface and not because FriCAS system could not do the integration.

These will fail With error `Exception raised: NotImplementedError`.

The number of such cases seems to be very small. About 1 or 2 percent of all integrals. These can be identified by looking at the exception message given in the result.

Important note about finding leaf size of antiderivative

For Mathematica, Rubi, and Maple, the builtin system function `LeafSize` was used to find the leaf size of each antiderivative.

The other CAS systems (SageMath and Sympy) do not have special builtin function for this purpose at this time. Therefore the leaf size for Fricas and Sympy antiderivative was determined using the following function, thanks to user `slelievre` at `https://ask.sagemath.org/question/57123/could-we-have-a-leaf_count-function-in-base-sagemath/`

```
def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
```

```
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)
```

For Sympy, which was called directly from Python, the following code was used to obtain the leafsize of its result

```
try:
  # 1.7 is a fudge factor since it is low side from actual leaf count
  leafCount = round(1.7*count_ops(anti))

  except Exception as ee:
        leafCount =1
```

Important note about Mupad results

Matlab's symbolic toolbox does not have a leaf count function to measure the size of the antiderivative. Maple was used to determine the leaf size of Mupad output by post processing Mupad result.

Currently no grading of the antiderivative for Mupad is implemented. If it can integrate the problem, it was assigned a B grade automatically as a placeholder. In the future, when grading function is implemented for Mupad, the tests will be rerun again.

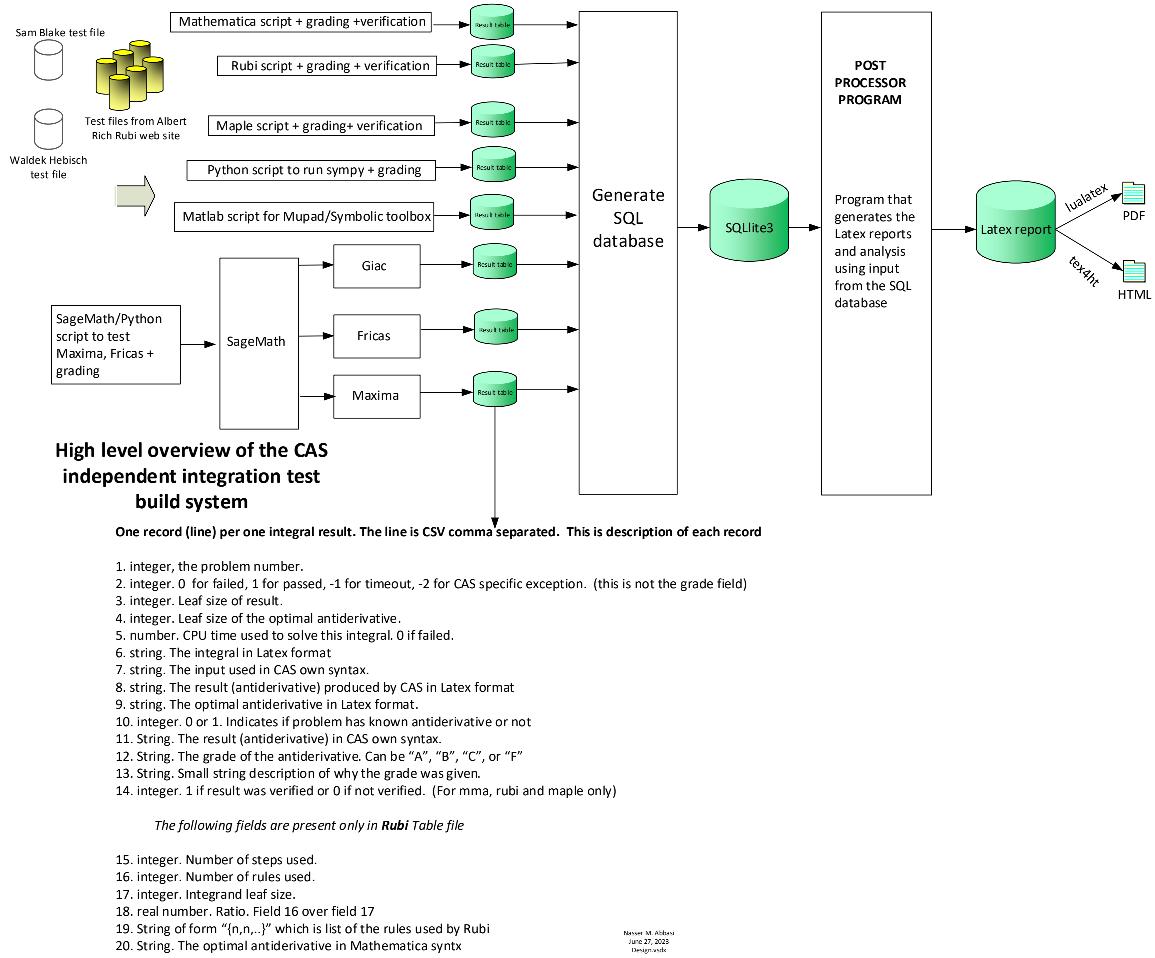
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

```
integrand = evalin(symengine,'cos(x)*sin(x)')
the_variable = evalin(symengine,'x')
anti = int(integrand,the_variable)
```

Which gives `sin(x)^2/2`

1.15 Design of the test system

The following diagram gives a high level view of the current test build system.



CHAPTER 2

DETAILED SUMMARY TABLES OF RESULTS

2.1 List of integrals sorted by grade for each CAS

Rubi

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111 }

B grade { }

C grade { }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Mma

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 108, 109, 110, 111 }

B grade { }

C grade { 15, 16, 17, 18, 19, 31, 32, 33, 34, 35, 47, 48, 49, 50, 51, 103, 104, 105, 106, 107 }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Maple

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 60, 61, 62, 63, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111 }

B grade { }

C grade { 20, 21, 22, 23, 24, 25, 36, 37, 38, 39, 40, 41, 52, 53, 54, 55, 56, 57, 58, 59, 64, 65, 66 }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Fricas

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 31, 32, 33, 34, 35, 47, 48, 49, 60, 61, 62, 63, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 103, 104, 105, 106, 108, 109, 110, 111 }

B grade { 26, 27, 28, 29, 30, 42, 43, 44, 45, 46, 50, 51, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 107 }

C grade { 20, 21, 22, 36, 64 }

F normal fail { }

F(-1) timedout fail { 23, 24, 25, 37, 38, 39, 40, 41, 52, 53, 54, 55, 56, 57, 58, 59, 65, 66 }

F(-2) exception fail { }

Maxima

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 60, 61, 62, 63, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 108, 109, 110, 111 }

B grade { }

C grade { }

F normal fail { 20, 21, 22, 23, 24, 25, 36, 37, 38, 39, 40, 41, 52, 53, 54, 55, 56, 57, 58, 59, 64, 65, 66, 103, 104, 105, 106, 107 }

F(-1) timedout fail { }

F(-2) exception fail { }

Giac

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 60, 61, 62, 63, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102 }

B grade { 20, 21, 22, 23, 24, 25, 36, 37, 38, 39, 40, 41, 52, 53, 54, 55, 56, 57, 58, 59, 64, 65, 66, 108, 109, 110, 111 }

C grade { }

F normal fail { 103, 104, 105, 106, 107 }

F(-1) timedout fail { }

F(-2) exception fail { }

Mupad

A grade { }

B grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 108, 109, 110, 111 }

C grade { }

F normal fail { }

F(-1) timedout fail { 103, 104, 105, 106, 107 }

F(-2) exception fail { }

Sympy

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 60, 61, 62, 63, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 85, 91, 97 }

B grade { 10, 11, 26, 42, 80, 81, 82, 86, 92, 98 }

C grade { 15, 16, 31, 32, 47, 48 }

F normal fail { 103, 104, 105, 106, 108, 109, 110, 111 }

F(-1) timedout fail { 12, 13, 14, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39, 40, 41, 43, 44, 45, 46, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 64, 65, 66, 83, 84, 87, 88, 89, 90, 93, 94, 95, 96, 99, 100, 101, 102, 107 }

F(-2) exception fail { }

2.2 Detailed conclusion table per each integral for all CAS systems

Detailed conclusion table per each integral is given by the table below. The elapsed time is in seconds. For failed result it is given as **F(-1)** if the failure was due to timeout. It is given as **F(-2)** if the failure was due to an exception being raised, which could indicate a bug in the system. If the failure was due to integral not being evaluated within the time limit, then it is given as **F**.

In this table, the column `N.S.` means **normalized size** and is defined as $\frac{\text{antiderivative leaf size}}{\text{optimal antiderivative leaf size}}$. To make the table fit the page, the name **`Mathematica`** was abbreviated to **`MMA`**.

Problem 1	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	50	50	50	41	40	40	46	40	40
N.S.	1	1.00	1.00	0.82	0.80	0.80	0.92	0.80	0.80
time (sec)	N/A	0.028	0.016	0.021	0.179	0.233	0.017	0.297	0.016

Problem 2	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	69	69	69	58	57	57	65	61	59
N.S.	1	1.00	1.00	0.84	0.83	0.83	0.94	0.88	0.86
time (sec)	N/A	0.032	0.022	0.067	0.180	0.239	0.018	0.361	0.019

Problem 3	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	88	88	88	75	74	74	83	82	78
N.S.	1	1.00	1.00	0.85	0.84	0.84	0.94	0.93	0.89
time (sec)	N/A	0.047	0.031	0.141	0.182	0.232	0.018	0.290	7.853

Problem 4	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	105	105	105	90	89	89	102	103	95
N.S.	1	1.00	1.00	0.86	0.85	0.85	0.97	0.98	0.90
time (sec)	N/A	0.062	0.031	0.234	0.191	0.237	0.019	0.366	7.905

Problem 5	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	122	122	122	105	104	104	121	124	112
N.S.	1	1.00	1.00	0.86	0.85	0.85	0.99	1.02	0.92
time (sec)	N/A	0.071	0.032	1.908	0.189	0.246	0.020	0.294	0.048

Problem 6	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	112	112	97	95	94	94	116	100	94
N.S.	1	1.00	0.87	0.85	0.84	0.84	1.04	0.89	0.84
time (sec)	N/A	0.086	0.039	0.099	0.182	0.243	0.024	0.369	0.047

Problem 7	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	154	154	154	139	138	138	165	151	138
N.S.	1	1.00	1.00	0.90	0.90	0.90	1.07	0.98	0.90
time (sec)	N/A	0.087	0.038	0.133	0.202	0.237	0.033	0.306	7.852

Problem 8	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	196	196	196	183	182	182	209	202	182
N.S.	1	1.00	1.00	0.93	0.93	0.93	1.07	1.03	0.93
time (sec)	N/A	0.118	0.044	0.151	0.186	0.250	0.031	0.378	7.879

Problem 9	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	234	234	234	219	218	218	258	253	220
N.S.	1	1.00	1.00	0.94	0.93	0.93	1.10	1.08	0.94
time (sec)	N/A	0.156	0.061	0.232	0.195	0.242	0.033	0.319	0.083

Problem 10	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	45	45	50	50	43	43	515	47	51
N.S.	1	1.00	1.11	1.11	0.96	0.96	11.44	1.04	1.13
time (sec)	N/A	0.023	0.017	0.050	0.181	0.262	1.809	0.287	7.813

Problem 11	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	51	51	58	62	51	51	2195	55	63
N.S.	1	1.00	1.14	1.22	1.00	1.00	43.04	1.08	1.24
time (sec)	N/A	0.036	0.022	0.058	0.203	0.289	95.810	0.364	7.763

Problem 12	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	57	57	68	74	61	61	0	65	75
N.S.	1	1.00	1.19	1.30	1.07	1.07	0.00	1.14	1.32
time (sec)	N/A	0.047	0.027	0.069	0.194	0.416	0.000	0.301	7.797

Problem 13	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	64	64	81	89	72	72	0	76	90
N.S.	1	1.00	1.27	1.39	1.12	1.12	0.00	1.19	1.41
time (sec)	N/A	0.094	0.035	0.084	0.203	1.139	0.000	0.311	7.904

Problem 14	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	76	76	98	107	88	88	0	92	108
N.S.	1	1.00	1.29	1.41	1.16	1.16	0.00	1.21	1.42
time (sec)	N/A	0.123	0.045	0.091	0.196	4.998	0.000	0.299	8.146

Problem 15	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	92	92	98	68	65	65	923	65	118
N.S.	1	1.00	1.07	0.74	0.71	0.71	10.03	0.71	1.28
time (sec)	N/A	0.051	0.169	0.122	0.260	0.267	1.662	0.304	0.128

Problem 16	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	104	104	121	82	75	75	3589	75	159
N.S.	1	1.00	1.16	0.79	0.72	0.72	34.51	0.72	1.53
time (sec)	N/A	0.060	0.110	0.226	0.274	0.289	65.868	0.297	7.978

Problem 17	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	127	127	150	90	83	83	0	83	199
N.S.	1	1.00	1.18	0.71	0.65	0.65	0.00	0.65	1.57
time (sec)	N/A	0.075	0.363	0.267	0.263	0.409	0.000	0.300	8.105

Problem 18	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	136	136	165	99	92	92	0	92	1209
N.S.	1	1.00	1.21	0.73	0.68	0.68	0.00	0.68	8.89
time (sec)	N/A	0.097	0.471	0.386	0.277	1.119	0.000	0.318	11.144

Problem 19	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	151	151	187	117	106	106	0	106	1509
N.S.	1	1.00	1.24	0.77	0.70	0.70	0.00	0.70	9.99
time (sec)	N/A	0.120	0.472	0.457	0.275	4.389	0.000	0.304	12.662

Problem 20	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	C	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	189	189	194	43	0	398481	0	1342	1308
N.S.	1	1.00	1.03	0.23	0.00	2108.37	0.00	7.10	6.92
time (sec)	N/A	0.132	0.172	0.084	0.000	3.158	0.000	1.399	8.363

Problem 21	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	C	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	211	211	234	48	0	723401	0	1714	3942
N.S.	1	1.00	1.11	0.23	0.00	3428.44	0.00	8.12	18.68
time (sec)	N/A	0.158	0.136	0.070	0.000	18.016	0.000	1.287	8.641

Problem 22	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	C	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	245	245	280	53	0	2136355	0	3270	15179
N.S.	1	1.00	1.14	0.22	0.00	8719.82	0.00	13.35	61.96
time (sec)	N/A	0.107	0.181	0.067	0.000	249.076	0.000	1.331	8.942

Problem 23	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	290	290	383	78	0	0	0	5199	5981
N.S.	1	1.00	1.32	0.27	0.00	0.00	0.00	17.93	20.62
time (sec)	N/A	0.452	0.308	0.102	0.000	0.000	0.000	1.470	8.465

Problem 24	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	321	321	441	99	0	0	0	5941	11383
N.S.	1	1.00	1.37	0.31	0.00	0.00	0.00	18.51	35.46
time (sec)	N/A	0.301	0.392	0.096	0.000	0.000	0.000	1.547	8.741

Problem 25	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	545	545	816	246	0	0	0	11830	49150
N.S.	1	1.00	1.50	0.45	0.00	0.00	0.00	21.71	90.18
time (sec)	N/A	2.845	0.729	0.762	0.000	0.000	0.000	2.097	10.688

Problem 26	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	94	94	90	83	83	169	604	87	84
N.S.	1	1.00	0.96	0.88	0.88	1.80	6.43	0.93	0.89
time (sec)	N/A	0.039	0.038	0.081	0.180	0.278	2.041	0.298	0.053

Problem 27	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	115	115	112	105	106	217	0	109	107
N.S.	1	1.00	0.97	0.91	0.92	1.89	0.00	0.95	0.93
time (sec)	N/A	0.099	0.054	0.088	0.194	0.315	0.000	0.315	0.060

Problem 28	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	138	138	134	125	127	262	0	130	128
N.S.	1	1.00	0.97	0.91	0.92	1.90	0.00	0.94	0.93
time (sec)	N/A	0.100	0.039	0.102	0.194	0.486	0.000	0.295	7.876

Problem 29	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	150	150	159	143	145	304	0	152	146
N.S.	1	1.00	1.06	0.95	0.97	2.03	0.00	1.01	0.97
time (sec)	N/A	0.127	0.049	0.103	0.193	1.405	0.000	0.302	7.980

Problem 30	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	162	162	185	161	163	346	0	173	164
N.S.	1	1.00	1.14	0.99	1.01	2.14	0.00	1.07	1.01
time (sec)	N/A	0.152	0.059	0.123	0.198	6.239	0.000	0.306	0.347

Problem 31	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	140	140	146	124	96	154	952	96	149
N.S.	1	1.00	1.04	0.89	0.69	1.10	6.80	0.69	1.06
time (sec)	N/A	0.067	0.351	0.153	0.283	0.277	2.014	0.303	0.145

Problem 32	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	165	165	186	154	120	212	4106	124	201
N.S.	1	1.00	1.13	0.93	0.73	1.28	24.88	0.75	1.22
time (sec)	N/A	0.088	0.282	0.268	0.280	0.301	81.025	0.324	0.180

Problem 33	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	179	179	200	172	135	239	0	138	237
N.S.	1	1.00	1.12	0.96	0.75	1.34	0.00	0.77	1.32
time (sec)	N/A	0.096	0.267	0.276	0.274	0.464	0.000	0.331	8.138

Problem 34	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	187	187	234	196	143	255	0	151	1547
N.S.	1	1.00	1.25	1.05	0.76	1.36	0.00	0.81	8.27
time (sec)	N/A	0.114	0.387	0.412	0.278	1.101	0.000	0.415	3.309

Problem 35	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	194	194	243	214	155	279	0	165	1894
N.S.	1	1.00	1.25	1.10	0.80	1.44	0.00	0.85	9.76
time (sec)	N/A	0.140	0.388	0.506	0.294	4.866	0.000	0.298	13.239

Problem 36	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	C	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	330	330	341	209	0	1678440	0	3426	2382
N.S.	1	1.00	1.03	0.63	0.00	5086.18	0.00	10.38	7.22
time (sec)	N/A	0.500	0.503	0.247	0.000	235.031	0.000	1.345	8.365

Problem 37	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	368	368	398	232	0	0	0	5156	4707
N.S.	1	1.00	1.08	0.63	0.00	0.00	0.00	14.01	12.79
time (sec)	N/A	0.562	0.720	0.260	0.000	0.000	0.000	1.737	8.403

Problem 38	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	386	386	421	252	0	0	0	5573	7373
N.S.	1	1.00	1.09	0.65	0.00	0.00	0.00	14.44	19.10
time (sec)	N/A	0.261	0.741	0.279	0.000	0.000	0.000	1.787	8.609

Problem 39	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	439	439	489	274	0	0	0	7495	13024
N.S.	1	1.00	1.11	0.62	0.00	0.00	0.00	17.07	29.67
time (sec)	N/A	1.218	1.093	0.154	0.000	0.000	0.000	1.730	8.795

Problem 40	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	468	468	524	304	0	0	0	7962	18449
N.S.	1	1.00	1.12	0.65	0.00	0.00	0.00	17.01	39.42
time (sec)	N/A	0.674	1.211	0.154	0.000	0.000	0.000	1.908	9.411

Problem 41	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	770	770	935	506	0	0	0	20159	82785
N.S.	1	1.00	1.21	0.66	0.00	0.00	0.00	26.18	107.51
time (sec)	N/A	4.435	3.220	1.013	0.000	0.000	0.000	3.934	22.902

Problem 42	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	143	143	128	107	121	307	668	115	118
N.S.	1	1.00	0.90	0.75	0.85	2.15	4.67	0.80	0.83
time (sec)	N/A	0.049	0.068	0.090	0.189	0.291	2.230	0.313	0.053

Problem 43	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	175	175	161	139	155	389	0	149	151
N.S.	1	1.00	0.92	0.79	0.89	2.22	0.00	0.85	0.86
time (sec)	N/A	0.141	0.088	0.117	0.201	0.316	0.000	0.320	0.063

Problem 44	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	204	204	193	169	188	470	0	182	182
N.S.	1	1.00	0.95	0.83	0.92	2.30	0.00	0.89	0.89
time (sec)	N/A	0.162	0.062	0.109	0.194	0.505	0.000	0.310	7.862

Problem 45	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	224	224	231	193	214	544	0	216	209
N.S.	1	1.00	1.03	0.86	0.96	2.43	0.00	0.96	0.93
time (sec)	N/A	0.215	0.087	0.124	0.197	1.450	0.000	0.364	0.151

Problem 46	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	239	239	261	217	238	616	0	249	233
N.S.	1	1.00	1.09	0.91	1.00	2.58	0.00	1.04	0.97
time (sec)	N/A	0.216	0.085	0.131	0.189	6.601	0.000	0.282	0.372

Problem 47	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	185	185	186	158	137	278	1103	125	185
N.S.	1	1.00	1.01	0.85	0.74	1.50	5.96	0.68	1.00
time (sec)	N/A	0.086	0.530	0.183	0.274	0.294	2.187	0.287	0.153

Problem 48	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	223	223	235	202	173	384	4496	165	249
N.S.	1	1.00	1.05	0.91	0.78	1.72	20.16	0.74	1.12
time (sec)	N/A	0.141	0.390	0.277	0.286	0.309	138.761	0.325	7.974

Problem 49	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	243	243	259	232	200	435	0	192	295
N.S.	1	1.00	1.07	0.95	0.82	1.79	0.00	0.79	1.21
time (sec)	N/A	0.153	0.435	0.304	0.272	0.487	0.000	0.307	8.066

Problem 50	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	263	263	303	262	217	485	0	222	1611
N.S.	1	1.00	1.15	1.00	0.83	1.84	0.00	0.84	6.13
time (sec)	N/A	0.179	0.568	0.432	0.284	1.221	0.000	0.307	3.443

Problem 51	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	269	269	325	292	229	521	0	249	1963
N.S.	1	1.00	1.21	1.09	0.85	1.94	0.00	0.93	7.30
time (sec)	N/A	0.200	0.580	0.505	0.281	5.488	0.000	0.295	12.838

Problem 52	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	474	474	488	499	0	0	0	3389	4225
N.S.	1	1.00	1.03	1.05	0.00	0.00	0.00	7.15	8.91
time (sec)	N/A	1.442	1.137	0.528	0.000	0.000	0.000	3.107	8.771

Problem 53	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	621	621	625	607	0	0	0	5284	8689
N.S.	1	1.00	1.01	0.98	0.00	0.00	0.00	8.51	13.99
time (sec)	N/A	2.935	1.929	0.648	0.000	0.000	0.000	2.444	9.384

Problem 54	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	646	646	661	644	0	0	0	5619	13431
N.S.	1	1.00	1.02	1.00	0.00	0.00	0.00	8.70	20.79
time (sec)	N/A	2.091	2.357	0.579	0.000	0.000	0.000	2.552	10.415

Problem 55	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	679	679	739	724	0	0	0	6854	23811
N.S.	1	1.00	1.09	1.07	0.00	0.00	0.00	10.09	35.07
time (sec)	N/A	2.761	3.899	0.240	0.000	0.000	0.000	3.094	10.961

Problem 56	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	728	728	821	795	0	0	0	7340	36653
N.S.	1	1.00	1.13	1.09	0.00	0.00	0.00	10.08	50.35
time (sec)	N/A	1.848	5.350	0.296	0.000	0.000	0.000	3.383	12.374

Problem 57	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	1150	1144	1590	1167	0	0	0	22429	114377
N.S.	1	0.99	1.38	1.01	0.00	0.00	0.00	19.50	99.46
time (sec)	N/A	5.063	6.849	1.392	0.000	0.000	0.000	3.501	32.044

Problem 58	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	645	645	775	422	0	0	0	16214	53538
N.S.	1	1.00	1.20	0.65	0.00	0.00	0.00	25.14	83.00
time (sec)	N/A	2.162	2.474	0.583	0.000	0.000	0.000	3.249	13.368

Problem 59	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	1177	1179	1649	1182	0	0	0	29142	97905
N.S.	1	1.00	1.40	1.00	0.00	0.00	0.00	24.76	83.18
time (sec)	N/A	5.117	6.801	1.140	0.000	0.000	0.000	4.868	27.137

Problem 60	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	416	416	416	412	418	418	503	463	398
N.S.	1	1.00	1.00	0.99	1.00	1.00	1.21	1.11	0.96
time (sec)	N/A	0.407	0.078	28.809	0.190	0.258	0.053	0.304	0.298

Problem 61	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	259	259	259	259	251	251	309	285	246
N.S.	1	1.00	1.00	1.00	0.97	0.97	1.19	1.10	0.95
time (sec)	N/A	0.218	0.035	27.405	0.191	0.270	0.041	0.298	8.113

Problem 62	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	154	154	154	141	138	138	165	151	138
N.S.	1	1.00	1.00	0.92	0.90	0.90	1.07	0.98	0.90
time (sec)	N/A	0.100	0.027	0.134	0.186	0.276	0.027	0.387	0.069

Problem 63	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	20	20	20	17	16	16	15	16	16
N.S.	1	1.00	1.00	0.85	0.80	0.80	0.75	0.80	0.80
time (sec)	N/A	0.021	0.004	0.025	0.187	0.258	0.028	0.619	0.016

Problem 64	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	C	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	211	211	234	48	0	723401	0	1616	3942
N.S.	1	1.00	1.11	0.23	0.00	3428.44	0.00	7.66	18.68
time (sec)	N/A	0.195	0.143	0.063	0.000	18.527	0.000	1.225	0.637

Problem 65	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	368	368	398	232	0	0	0	5159	4707
N.S.	1	1.00	1.08	0.63	0.00	0.00	0.00	14.02	12.79
time (sec)	N/A	0.560	0.738	0.243	0.000	0.000	0.000	2.825	8.620

Problem 66	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	C	F	F(-1)	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	621	621	625	607	0	0	0	5280	8689
N.S.	1	1.00	1.01	0.98	0.00	0.00	0.00	8.50	13.99
time (sec)	N/A	2.638	2.034	0.558	0.000	0.000	0.000	1.596	9.343

Problem 67	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	4	4	4	5	4	4	3	5	4
N.S.	1	1.00	1.00	1.25	1.00	1.00	0.75	1.25	1.00
time (sec)	N/A	0.009	0.002	0.025	0.183	0.269	0.024	0.279	0.012

Problem 68	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	14	14	16	15	14	14	12	15	14
N.S.	1	1.00	1.14	1.07	1.00	1.00	0.86	1.07	1.00
time (sec)	N/A	0.017	0.017	0.033	0.197	0.262	0.057	0.315	7.856

Problem 69	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	31	31	30	28	27	27	26	28	27
N.S.	1	1.00	0.97	0.90	0.87	0.87	0.84	0.90	0.87
time (sec)	N/A	0.034	0.023	0.034	0.181	0.262	0.077	0.306	0.022

Problem 70	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	51	51	45	45	43	43	41	47	44
N.S.	1	1.00	0.88	0.88	0.84	0.84	0.80	0.92	0.86
time (sec)	N/A	0.056	0.022	0.039	0.191	0.261	0.082	0.299	0.022

Problem 71	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	68	68	68	65	62	62	63	72	64
N.S.	1	1.00	1.00	0.96	0.91	0.91	0.93	1.06	0.94
time (sec)	N/A	0.079	0.023	0.042	0.188	0.244	0.099	0.298	0.020

Problem 72	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	92	92	92	88	84	84	88	103	87
N.S.	1	1.00	1.00	0.96	0.91	0.91	0.96	1.12	0.95
time (sec)	N/A	0.111	0.042	0.045	0.183	0.251	0.126	0.295	0.023

Problem 73	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	11	11	11	12	11	11	8	13	8
N.S.	1	1.00	1.00	1.09	1.00	1.00	0.73	1.18	0.73
time (sec)	N/A	0.007	0.045	0.032	0.189	0.254	0.044	0.309	0.051

Problem 74	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	22	22	23	24	22	22	29	24	22
N.S.	1	1.00	1.05	1.09	1.00	1.00	1.32	1.09	1.00
time (sec)	N/A	0.012	0.016	0.038	0.181	0.247	0.165	0.290	7.893

Problem 75	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	29	29	30	31	29	29	44	31	29
N.S.	1	1.00	1.03	1.07	1.00	1.00	1.52	1.07	1.00
time (sec)	N/A	0.033	0.042	0.044	0.179	0.239	0.301	0.304	0.043

Problem 76	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	47	47	44	47	45	45	66	47	45
N.S.	1	1.00	0.94	1.00	0.96	0.96	1.40	1.00	0.96
time (sec)	N/A	0.047	0.070	0.052	0.189	0.253	0.480	0.302	0.039

Problem 77	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	66	66	67	65	62	62	94	67	63
N.S.	1	1.00	1.02	0.98	0.94	0.94	1.42	1.02	0.95
time (sec)	N/A	0.057	0.039	0.056	0.191	0.256	0.850	0.308	0.043

Problem 78	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	90	90	91	88	84	84	122	95	86
N.S.	1	1.00	1.01	0.98	0.93	0.93	1.36	1.06	0.96
time (sec)	N/A	0.072	0.029	0.061	0.179	0.249	1.394	0.299	0.051

Problem 79	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	29	29	29	20	19	19	19	22	19
N.S.	1	1.00	1.00	0.69	0.66	0.66	0.66	0.76	0.66
time (sec)	N/A	0.013	0.007	0.038	0.190	0.251	0.063	0.285	0.041

Problem 80	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	42	42	39	38	32	32	304	35	38
N.S.	1	1.00	0.93	0.90	0.76	0.76	7.24	0.83	0.90
time (sec)	N/A	0.032	0.014	0.046	0.178	0.261	1.109	0.274	7.785

Problem 81	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	47	47	44	47	37	37	716	40	47
N.S.	1	1.00	0.94	1.00	0.79	0.79	15.23	0.85	1.00
time (sec)	N/A	0.041	0.017	0.056	0.186	0.250	7.753	0.306	0.066

Problem 82	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	57	57	55	59	47	47	1389	50	59
N.S.	1	1.00	0.96	1.04	0.82	0.82	24.37	0.88	1.04
time (sec)	N/A	0.055	0.031	0.059	0.189	0.268	54.944	0.301	7.872

Problem 83	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	74	74	71	78	62	62	0	65	78
N.S.	1	1.00	0.96	1.05	0.84	0.84	0.00	0.88	1.05
time (sec)	N/A	0.072	0.028	0.073	0.188	0.284	0.000	0.286	8.034

Problem 84	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	96	96	91	99	82	82	0	87	99
N.S.	1	1.00	0.95	1.03	0.85	0.85	0.00	0.91	1.03
time (sec)	N/A	0.093	0.034	0.076	0.184	0.315	0.000	0.302	7.928

Problem 85	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	46	46	42	33	32	45	34	36	32
N.S.	1	1.00	0.91	0.72	0.70	0.98	0.74	0.78	0.70
time (sec)	N/A	0.034	0.017	0.060	0.186	0.266	0.138	0.293	0.030

Problem 86	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	71	71	66	64	57	93	1188	61	64
N.S.	1	1.00	0.93	0.90	0.80	1.31	16.73	0.86	0.90
time (sec)	N/A	0.122	0.034	0.069	0.183	0.279	7.105	0.299	7.948

Problem 87	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	82	82	77	79	68	116	0	72	79
N.S.	1	1.00	0.94	0.96	0.83	1.41	0.00	0.88	0.96
time (sec)	N/A	0.138	0.043	0.076	0.193	0.340	0.000	0.301	7.902

Problem 88	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	95	95	90	94	81	141	0	85	94
N.S.	1	1.00	0.95	0.99	0.85	1.48	0.00	0.89	0.99
time (sec)	N/A	0.154	0.041	0.083	0.184	0.722	0.000	0.286	7.943

Problem 89	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	106	106	102	109	92	164	0	96	108
N.S.	1	1.00	0.96	1.03	0.87	1.55	0.00	0.91	1.02
time (sec)	N/A	0.185	0.045	0.098	0.181	3.260	0.000	0.291	8.113

Problem 90	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	122	122	118	127	108	200	0	112	127
N.S.	1	1.00	0.97	1.04	0.89	1.64	0.00	0.92	1.04
time (sec)	N/A	0.211	0.050	0.116	0.190	19.688	0.000	0.297	8.369

Problem 91	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	56	56	48	40	42	72	46	46	42
N.S.	1	1.00	0.86	0.71	0.75	1.29	0.82	0.82	0.75
time (sec)	N/A	0.045	0.019	0.064	0.192	0.254	0.149	0.313	0.027

Problem 92	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	89	89	80	78	75	153	1255	79	79
N.S.	1	1.00	0.90	0.88	0.84	1.72	14.10	0.89	0.89
time (sec)	N/A	0.159	0.035	0.073	0.197	0.270	6.979	0.290	0.057

Problem 93	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	105	105	97	96	91	191	0	95	97
N.S.	1	1.00	0.92	0.91	0.87	1.82	0.00	0.90	0.92
time (sec)	N/A	0.202	0.055	0.081	0.186	0.340	0.000	0.304	0.077

Problem 94	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	117	117	114	114	107	229	0	111	115
N.S.	1	1.00	0.97	0.97	0.91	1.96	0.00	0.95	0.98
time (sec)	N/A	0.174	0.043	0.100	0.186	0.725	0.000	0.300	7.978

Problem 95	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	131	131	136	132	123	267	0	127	133
N.S.	1	1.00	1.04	1.01	0.94	2.04	0.00	0.97	1.02
time (sec)	N/A	0.198	0.060	0.124	0.187	3.518	0.000	0.290	8.236

Problem 96	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	147	147	153	150	139	305	0	143	151
N.S.	1	1.00	1.04	1.02	0.95	2.07	0.00	0.97	1.03
time (sec)	N/A	0.205	0.063	0.125	0.197	21.463	0.000	0.297	8.666

Problem 97	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	68	68	60	47	52	103	53	56	52
N.S.	1	1.00	0.88	0.69	0.76	1.51	0.78	0.82	0.76
time (sec)	N/A	0.038	0.021	0.067	0.187	0.251	0.157	0.295	0.026

Problem 98	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	105	105	97	92	88	211	1034	92	90
N.S.	1	1.00	0.92	0.88	0.84	2.01	9.85	0.88	0.86
time (sec)	N/A	0.134	0.060	0.077	0.191	0.279	5.457	0.300	0.058

Problem 99	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	122	122	121	113	108	267	0	112	113
N.S.	1	1.00	0.99	0.93	0.89	2.19	0.00	0.92	0.93
time (sec)	N/A	0.148	0.042	0.080	0.198	0.361	0.000	0.285	7.901

Problem 100	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	141	141	144	134	126	321	0	130	131
N.S.	1	1.00	1.02	0.95	0.89	2.28	0.00	0.92	0.93
time (sec)	N/A	0.167	0.052	0.088	0.188	0.777	0.000	0.288	8.211

Problem 101	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	158	158	169	155	145	376	0	149	152
N.S.	1	1.00	1.07	0.98	0.92	2.38	0.00	0.94	0.96
time (sec)	N/A	0.201	0.062	0.111	0.192	3.629	0.000	0.293	8.431

Problem 102	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	B	F(-1)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	177	177	195	176	163	430	0	167	170
N.S.	1	1.00	1.10	0.99	0.92	2.43	0.00	0.94	0.96
time (sec)	N/A	0.231	0.071	0.129	0.196	20.839	0.000	0.299	8.687

Problem 103	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	F	A	F	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	717	717	2588	1209	0	911	0	0	0
N.S.	1	1.00	3.61	1.69	0.00	1.27	0.00	0.00	0.00
time (sec)	N/A	0.393	11.689	5.791	0.000	0.313	0.000	0.000	0.000

Problem 104	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	F	A	F	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	505	505	661	620	0	574	0	0	0
N.S.	1	1.00	1.31	1.23	0.00	1.14	0.00	0.00	0.00
time (sec)	N/A	0.178	12.543	5.344	0.000	0.255	0.000	0.000	0.000

Problem 105	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	F	A	F	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	359	359	525	426	0	376	0	0	0
N.S.	1	1.00	1.46	1.19	0.00	1.05	0.00	0.00	0.00
time (sec)	N/A	0.102	10.869	4.082	0.000	0.189	0.000	0.000	0.000

Problem 106	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	F	A	F	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	447	447	513	565	0	723	0	0	0
N.S.	1	1.00	1.15	1.26	0.00	1.62	0.00	0.00	0.00
time (sec)	N/A	0.174	11.157	2.107	0.000	0.115	0.000	0.000	0.000

Problem 107	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	F	B	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	680	680	598	829	0	1948	0	0	0
N.S.	1	1.00	0.88	1.22	0.00	2.86	0.00	0.00	0.00
time (sec)	N/A	0.323	11.995	2.319	0.000	0.145	0.000	0.000	0.000

Problem 108	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	19	19	19	18	17	17	0	60	17
N.S.	1	1.00	1.00	0.95	0.89	0.89	0.00	3.16	0.89
time (sec)	N/A	0.014	10.093	1.102	0.220	0.270	0.000	0.669	8.012

Problem 109	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	57	57	51	52	51	82	0	138	51
N.S.	1	1.00	0.89	0.91	0.89	1.44	0.00	2.42	0.89
time (sec)	N/A	0.041	10.135	1.599	0.243	0.275	0.000	0.662	7.918

Problem 110	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	57	57	48	53	49	80	0	136	51
N.S.	1	1.00	0.84	0.93	0.86	1.40	0.00	2.39	0.89
time (sec)	N/A	0.055	10.083	1.596	0.226	0.276	0.000	0.622	7.940

Problem 111	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	F	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	69	69	61	63	94	92	0	164	62
N.S.	1	1.00	0.88	0.91	1.36	1.33	0.00	2.38	0.90
time (sec)	N/A	0.058	10.155	1.790	0.234	0.286	0.000	0.656	7.995

2.3 Detailed conclusion table specific for Rubi results

The following table is specific to Rubi only. It gives additional statistics for each integral. the column **steps** is the number of steps used by Rubi to obtain the antiderivative. The **rules** column is the number of unique rules used. The **integrand size** column is the leaf size of the integrand. Finally the ratio $\frac{\text{number of rules}}{\text{integrand size}}$ is also given. The larger this ratio is, the harder the integral is to solve. In this test file, problem number [47] had the largest ratio of [.6875000000000000000]

Table 2.1: Rubi specific breakdown of results for each integral

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
1	A	2	1	1.00	18	0.056
2	A	2	1	1.00	23	0.043
3	A	2	1	1.00	28	0.036
4	A	2	1	1.00	33	0.030
5	A	2	1	1.00	38	0.026
6	A	2	1	1.00	20	0.050
7	A	2	1	1.00	25	0.040
8	A	2	1	1.00	30	0.033
9	A	2	1	1.00	35	0.029
10	A	10	7	1.00	18	0.389
11	A	9	7	1.00	23	0.304
12	A	8	6	1.00	28	0.214
13	A	10	7	1.00	33	0.212
14	A	12	8	1.00	38	0.210
15	A	15	8	1.00	16	0.500
16	A	14	8	1.00	21	0.381
17	A	15	7	1.00	26	0.269
18	A	17	8	1.00	31	0.258
19	A	19	9	1.00	36	0.250
20	A	9	7	1.00	20	0.350
21	A	8	7	1.00	25	0.280
22	A	9	8	1.00	30	0.267
23	A	11	9	1.00	35	0.257
24	A	13	10	1.00	40	0.250

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
25	A	13	10	1.00	55	0.182
26	A	12	9	1.00	18	0.500
27	A	11	9	1.00	23	0.391
28	A	10	8	1.00	28	0.286
29	A	10	8	1.00	33	0.242
30	A	11	9	1.00	38	0.237
31	A	17	10	1.00	16	0.625
32	A	16	10	1.00	21	0.476
33	A	15	9	1.00	26	0.346
34	A	15	9	1.00	31	0.290
35	A	16	10	1.00	36	0.278
36	A	11	9	1.00	20	0.450
37	A	10	9	1.00	25	0.360
38	A	9	8	1.00	30	0.267
39	A	9	8	1.00	35	0.229
40	A	10	9	1.00	40	0.225
41	A	13	11	1.00	55	0.200
42	A	14	10	1.00	18	0.556
43	A	13	9	1.00	23	0.391
44	A	12	9	1.00	28	0.321
45	A	12	10	1.00	33	0.303
46	A	13	11	1.00	38	0.290
47	A	19	11	1.00	16	0.688
48	A	18	10	1.00	21	0.476
49	A	17	10	1.00	26	0.385
50	A	17	11	1.00	31	0.355
51	A	18	12	1.00	36	0.333
52	A	13	10	1.00	20	0.500
53	A	12	9	1.00	25	0.360
54	A	11	9	1.00	30	0.300
55	A	11	10	1.00	35	0.286
56	A	12	11	1.00	40	0.275
57	A	11	9	0.99	55	0.164
58	A	11	10	1.00	50	0.200
59	A	13	10	1.00	50	0.200

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
60	A	2	1	1.00	63	0.016
61	A	2	1	1.00	63	0.016
62	A	2	1	1.00	61	0.016
63	A	2	1	1.00	63	0.016
64	A	9	8	1.00	63	0.127
65	A	11	10	1.00	63	0.159
66	A	13	10	1.00	63	0.159
67	A	2	2	1.00	26	0.077
68	A	3	2	1.00	31	0.065
69	A	3	2	1.00	36	0.056
70	A	3	2	1.00	41	0.049
71	A	3	2	1.00	46	0.043
72	A	3	2	1.00	51	0.039
73	A	4	3	1.00	21	0.143
74	A	4	3	1.00	26	0.115
75	A	6	4	1.00	31	0.129
76	A	6	4	1.00	36	0.111
77	A	6	4	1.00	41	0.098
78	A	6	4	1.00	46	0.087
79	A	3	2	1.00	16	0.125
80	A	3	2	1.00	21	0.095
81	A	3	2	1.00	26	0.077
82	A	3	2	1.00	31	0.065
83	A	3	2	1.00	36	0.056
84	A	3	2	1.00	41	0.049
85	A	3	2	1.00	26	0.077
86	A	3	2	1.00	31	0.065
87	A	3	2	1.00	36	0.056
88	A	3	2	1.00	41	0.049
89	A	3	2	1.00	46	0.043
90	A	3	2	1.00	51	0.039
91	A	9	5	1.00	21	0.238
92	A	9	5	1.00	26	0.192
93	A	9	5	1.00	31	0.161
94	A	3	2	1.00	36	0.056

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
95	A	3	2	1.00	41	0.049
96	A	3	2	1.00	46	0.043
97	A	3	2	1.00	16	0.125
98	A	3	2	1.00	21	0.095
99	A	3	2	1.00	26	0.077
100	A	3	2	1.00	31	0.065
101	A	3	2	1.00	36	0.056
102	A	3	2	1.00	41	0.049
103	A	12	10	1.00	32	0.312
104	A	10	10	1.00	32	0.312
105	A	8	8	1.00	32	0.250
106	A	7	7	1.00	32	0.219
107	A	9	8	1.00	32	0.250
108	A	1	1	1.00	28	0.036
109	A	5	5	1.00	31	0.161
110	A	5	5	1.00	33	0.152
111	A	4	4	1.00	36	0.111

CHAPTER 3

LISTING OF INTEGRALS

3.1 $\int (d+ex)\left(a+bx^2+cx^4\right) dx$

Optimal result

Integrand size = 18, antiderivative size = 50

$$\int (d+ex)\left(a+bx^2+cx^4\right) dx = adx+\frac{1}{2}aex^2+\frac{1}{3}bdx^3+\frac{1}{4}bex^4+\frac{1}{5}cdx^5+\frac{1}{6}cex^6$$

```
[Out] a*d*x+1/2*a*e*x^2+1/3*b*d*x^3+1/4*b*e*x^4+1/5*c*d*x^5+1/6*c*e*x^6
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.056, Rules used = {1685}

$$\int (d+ex)\left(a+bx^2+cx^4\right) dx = adx+\frac{1}{2}aex^2+\frac{1}{3}bdx^3+\frac{1}{4}bex^4+\frac{1}{5}cdx^5+\frac{1}{6}cex^6$$

```
[In] Int[(d + e*x)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + (b*d*x^3)/3 + (b*e*x^4)/4 + (c*d*x^5)/5 + (c*e*x^6)/6
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(ad+aex+bdx^2+bex^3+cdx^4+cex^5\right) dx \\ &= adx+\frac{1}{2}aex^2+\frac{1}{3}bdx^3+\frac{1}{4}bex^4+\frac{1}{5}cdx^5+\frac{1}{6}cex^6\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.00

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=adx+\frac{1}{2}aex^2+\frac{1}{3}bdx^3+\frac{1}{4}bex^4+\frac{1}{5}cdx^5+\frac{1}{6}cex^6$$

```
[In] Integrate[(d + e*x)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + (b*d*x^3)/3 + (b*e*x^4)/4 + (c*d*x^5)/5 + (c*e*x^6)/6
```

Maple [A] (verified)

Time = 0.02 (sec) , antiderivative size = 41, normalized size of antiderivative = 0.82

method	result	size
gosper	$adx+\frac{1}{2}ae\,x^2+\frac{1}{3}x^3bd+\frac{1}{4}be\,x^4+\frac{1}{5}cd\,x^5+\frac{1}{6}ce\,x^6$	41
default	$adx+\frac{1}{2}ae\,x^2+\frac{1}{3}x^3bd+\frac{1}{4}be\,x^4+\frac{1}{5}cd\,x^5+\frac{1}{6}ce\,x^6$	41
norman	$adx+\frac{1}{2}ae\,x^2+\frac{1}{3}x^3bd+\frac{1}{4}be\,x^4+\frac{1}{5}cd\,x^5+\frac{1}{6}ce\,x^6$	41
risch	$adx+\frac{1}{2}ae\,x^2+\frac{1}{3}x^3bd+\frac{1}{4}be\,x^4+\frac{1}{5}cd\,x^5+\frac{1}{6}ce\,x^6$	41
parallelrisch	$adx+\frac{1}{2}ae\,x^2+\frac{1}{3}x^3bd+\frac{1}{4}be\,x^4+\frac{1}{5}cd\,x^5+\frac{1}{6}ce\,x^6$	41

```
[In] int((e*x+d)*(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)
```

```
[Out] a*d*x+1/2*a*e*x^2+1/3*x^3*b*d+1/4*b*e*x^4+1/5*c*d*x^5+1/6*c*e*x^6
```

Fricas [A] (verification not implemented)

none

Time = 0.23 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.80

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=\frac{1}{6}\,cex^6+\frac{1}{5}\,cdx^5+\frac{1}{4}\,bex^4+\frac{1}{3}\,bdx^3+\frac{1}{2}\,aex^2+adx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a),x, algorithm="fricas")
```

```
[Out] 1/6*c*e*x^6 + 1/5*c*d*x^5 + 1/4*b*e*x^4 + 1/3*b*d*x^3 + 1/2*a*e*x^2 + a*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 46, normalized size of antiderivative = 0.92

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=adx+\frac{aex^2}{2}+\frac{bdx^3}{3}+\frac{bex^4}{4}+\frac{cdx^5}{5}+\frac{cex^6}{6}$$

```
[In] integrate((e*x+d)*(c*x**4+b*x**2+a),x)
[Out] a*d*x + a*e*x**2/2 + b*d*x**3/3 + b*e*x**4/4 + c*d*x**5/5 + c*e*x**6/6
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.80

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=\frac{1}{6}\,cex^6+\frac{1}{5}\,cdx^5+\frac{1}{4}\,bex^4+\frac{1}{3}\,bdx^3+\frac{1}{2}\,aex^2+adx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] 1/6*c*e*x^6 + 1/5*c*d*x^5 + 1/4*b*e*x^4 + 1/3*b*d*x^3 + 1/2*a*e*x^2 + a*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.80

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=\frac{1}{6}\,cex^6+\frac{1}{5}\,cdx^5+\frac{1}{4}\,bex^4+\frac{1}{3}\,bdx^3+\frac{1}{2}\,aex^2+adx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a),x, algorithm="giac")
[Out] 1/6*c*e*x^6 + 1/5*c*d*x^5 + 1/4*b*e*x^4 + 1/3*b*d*x^3 + 1/2*a*e*x^2 + a*d*x
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.80

$$\int (d+ex)\left(a+bx^2+cx^4\right)\,dx=\frac{c\,e\,x^6}{6}+\frac{c\,d\,x^5}{5}+\frac{b\,e\,x^4}{4}+\frac{b\,d\,x^3}{3}+\frac{a\,e\,x^2}{2}+a\,d\,x$$

```
[In] int((d + e*x)*(a + b*x^2 + c*x^4),x)
[Out] a*d*x + (a*e*x^2)/2 + (b*d*x^3)/3 + (b*e*x^4)/4 + (c*d*x^5)/5 + (c*e*x^6)/6
```

3.2 $\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx$

Optimal result

Integrand size = 23, antiderivative size = 69

$$\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx = adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}bex^4+\frac{1}{5}(cd+bf)x^5+\frac{1}{6}cex^6+\frac{1}{7}cfx^7$$

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*b*e*x^4+1/5*(b*f+c*d)*x^5+1/6*c*e*x
^6+1/7*c*f*x^7
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.043, Rules used = {1671}

$$\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx = \frac{1}{3}x^3(af+bd)+adx+\frac{1}{2}aex^2+\frac{1}{5}x^5(bf+cd)+\frac{1}{4}bex^4+\frac{1}{6}cex^6+\frac{1}{7}cfx^7$$

```
[In] Int[(d + e*x + f*x^2)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + (b*e*x^4)/4 + ((c*d + b*f)*x^5)
/5 + (c*e*x^6)/6 + (c*f*x^7)/7
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$\text{integral} = \int \left(ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6\right)\,dx$$
$$= adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}bex^4 + \frac{1}{5}(cd + bf)x^5 + \frac{1}{6}cex^6 + \frac{1}{7}cfx^7$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.00

$$\int \left(d + ex + fx^2\right)\left(a + bx^2 + cx^4\right)\,dx = adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}bex^4 + \frac{1}{5}(cd + bf)x^5 + \frac{1}{6}cex^6 + \frac{1}{7}cfx^7$$

```
[In] Integrate[(d + e*x + f*x^2)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + (b*e*x^4)/4 + ((c*d + b*f)*x^5)
/5 + (c*e*x^6)/6 + (c*f*x^7)/7
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 58, normalized size of antiderivative = 0.84

method	result	size
default	$adx + \frac{ae\,x^2}{2} + \frac{(af+bd)x^3}{3} + \frac{be\,x^4}{4} + \frac{(bf+cd)x^5}{5} + \frac{ce\,x^6}{6} + \frac{cf\,x^7}{7}$	58
norman	$\frac{cf\,x^7}{7} + \frac{ce\,x^6}{6} + \left(\frac{bf}{5} + \frac{cd}{5}\right)x^5 + \frac{be\,x^4}{4} + \left(\frac{af}{3} + \frac{bd}{3}\right)x^3 + \frac{ae\,x^2}{2} + adx$	60
gosper	$\frac{1}{7}cf\,x^7 + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}be\,x^4 + \frac{1}{3}x^3af + \frac{1}{3}x^3bd + \frac{1}{2}ae\,x^2 + adx$	62
risch	$\frac{1}{7}cf\,x^7 + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}be\,x^4 + \frac{1}{3}x^3af + \frac{1}{3}x^3bd + \frac{1}{2}ae\,x^2 + adx$	62
parallelrisch	$\frac{1}{7}cf\,x^7 + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}be\,x^4 + \frac{1}{3}x^3af + \frac{1}{3}x^3bd + \frac{1}{2}ae\,x^2 + adx$	62

```
[In] int((f*x^2+e*x+d)*(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)
```

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*b*e*x^4+1/5*(b*f+c*d)*x^5+1/6*c*e*x
^6+1/7*c*f*x^7
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 57, normalized size of antiderivative = 0.83

$$\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx = \frac{1}{7}cfx^7 + \frac{1}{6}cex^6 + \frac{1}{4}bex^4 + \frac{1}{5}(cd+bf)x^5 + \frac{1}{2}aex^2 + \frac{1}{3}(bd+af)x^3 + adx$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="fricas")
[Out] 1/7*c*f*x^7 + 1/6*c*e*x^6 + 1/4*b*e*x^4 + 1/5*(c*d + b*f)*x^5 + 1/2*a*e*x^2
 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.94

$$\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx = adx + \frac{aex^2}{2} + \frac{bex^4}{4} + \frac{cex^6}{6} + \frac{cfx^7}{7} + x^5\left(\frac{bf}{5} + \frac{cd}{5}\right) + x^3\left(\frac{af}{3} + \frac{bd}{3}\right)$$

```
[In] integrate((f*x**2+e*x+d)*(c*x**4+b*x**2+a),x)
[Out] a*d*x + a*e*x**2/2 + b*e*x**4/4 + c*e*x**6/6 + c*f*x**7/7 + x**5*(b*f/5 + c
*d/5) + x**3*(a*f/3 + b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 57, normalized size of antiderivative = 0.83

$$\int (d+ex+fx^2)(a+bx^2+cx^4)\,dx = \frac{1}{7}cfx^7 + \frac{1}{6}cex^6 + \frac{1}{4}bex^4 + \frac{1}{5}(cd+bf)x^5 + \frac{1}{2}aex^2 + \frac{1}{3}(bd+af)x^3 + adx$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] 1/7*c*f*x^7 + 1/6*c*e*x^6 + 1/4*b*e*x^4 + 1/5*(c*d + b*f)*x^5 + 1/2*a*e*x^2
 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.36 (sec) , antiderivative size = 61, normalized size of antiderivative = 0.88

$$\int (d + ex + fx^2) (a + bx^2 + cx^4) \, dx = \frac{1}{7} c f x^7 + \frac{1}{6} c e x^6 + \frac{1}{5} c d x^5 + \frac{1}{5} b f x^5 + \frac{1}{4} b e x^4 + \frac{1}{3} b d x^3 + \frac{1}{3} a f x^3 + \frac{1}{2} a e x^2 + a d x$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="giac")
```

```
[Out] 1/7*c*f*x^7 + 1/6*c*e*x^6 + 1/5*c*d*x^5 + 1/5*b*f*x^5 + 1/4*b*e*x^4 + 1/3*b
*d*x^3 + 1/3*a*f*x^3 + 1/2*a*e*x^2 + a*d*x
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 59, normalized size of antiderivative = 0.86

$$\int (d + ex + fx^2) (a + bx^2 + cx^4) \, dx = \frac{c f x^7}{7} + \frac{c e x^6}{6} + \left(\frac{c d}{5} + \frac{b f}{5}\right) x^5 + \frac{b e x^4}{4} + \left(\frac{b d}{3} + \frac{a f}{3}\right) x^3 + \frac{a e x^2}{2} + a d x$$

```
[In] int((d + e*x + f*x^2)*(a + b*x^2 + c*x^4),x)
```

```
[Out] x^3*((b*d)/3 + (a*f)/3) + x^5*((c*d)/5 + (b*f)/5) + a*d*x + (a*e*x^2)/2 + (
b*e*x^4)/4 + (c*e*x^6)/6 + (c*f*x^7)/7
```

3.3 $\int (d+ex+fx^2+gx^3)(a+bx^2+cx^4)\,dx$

Optimal result

Integrand size = 28, antiderivative size = 88

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)\,dx = adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}(be+ag)x^4+\frac{1}{5}(cd+bf)x^5+\frac{1}{6}(ce+bg)x^6+\frac{1}{7}cfx^7+\frac{1}{8}cgx^8$$

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(b*f+c*d)*x^5+1/6
*(b*g+c*e)*x^6+1/7*c*f*x^7+1/8*c*g*x^8
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.036, Rules used = {1685}

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)\,dx = \frac{1}{3}x^3(af+bd)+\frac{1}{4}x^4(ag+be)+adx+\frac{1}{2}aex^2+\frac{1}{5}x^5(bf+cd)+\frac{1}{6}x^6(bg+ce)+\frac{1}{7}cfx^7+\frac{1}{8}cgx^8$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f)*x^5)/5 + ((c*e + b*g)*x^6)/6 + (c*f*x^7)/7 + (c*g*x^8)/8
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(ad+aex+(bd+af)x^2+(be+ag)x^3+(cd+bf)x^4+(ce+bg)x^5+cfx^6+cgx^7\right) dx \\ &= adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}(be+ag)x^4+\frac{1}{5}(cd+bf)x^5+\frac{1}{6}(ce+bg)x^6+\frac{1}{7}cfx^7 \\ &\quad +\frac{1}{8}cgx^8\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.00

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right) dx = adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}(be+ag)x^4+\frac{1}{5}(cd+bf)x^5+\frac{1}{6}(ce+bg)x^6+\frac{1}{7}cfx^7+\frac{1}{8}cgx^8$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f)*x^5)/5 + ((c*e + b*g)*x^6)/6 + (c*f*x^7)/7 + (c*g*x^8)/8
```

Maple [A] (verified)

Time = 0.14 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.85

method	result
default	$adx+\frac{ae\,x^2}{2}+\frac{(af+bd)x^3}{3}+\frac{(ag+be)x^4}{4}+\frac{(bf+cd)x^5}{5}+\frac{(bg+ec)x^6}{6}+\frac{cf\,x^7}{7}+\frac{cg\,x^8}{8}$
norman	$\frac{cg\,x^8}{8}+\frac{cf\,x^7}{7}+\left(\frac{bg}{6}+\frac{ec}{6}\right)x^6+\left(\frac{bf}{5}+\frac{cd}{5}\right)x^5+\left(\frac{ag}{4}+\frac{be}{4}\right)x^4+\left(\frac{af}{3}+\frac{bd}{3}\right)x^3+\frac{ae\,x^2}{2}+adx$
gosper	$\frac{1}{8}cg\,x^8+\frac{1}{7}cf\,x^7+\frac{1}{6}x^6bg+\frac{1}{6}ce\,x^6+\frac{1}{5}x^5bf+\frac{1}{5}cd\,x^5+\frac{1}{4}x^4ag+\frac{1}{4}be\,x^4+\frac{1}{3}x^3af+\frac{1}{3}x^3bd+$
risch	$\frac{1}{8}cg\,x^8+\frac{1}{7}cf\,x^7+\frac{1}{6}x^6bg+\frac{1}{6}ce\,x^6+\frac{1}{5}x^5bf+\frac{1}{5}cd\,x^5+\frac{1}{4}x^4ag+\frac{1}{4}be\,x^4+\frac{1}{3}x^3af+\frac{1}{3}x^3bd+$
parallelrisch	$\frac{1}{8}cg\,x^8+\frac{1}{7}cf\,x^7+\frac{1}{6}x^6bg+\frac{1}{6}ce\,x^6+\frac{1}{5}x^5bf+\frac{1}{5}cd\,x^5+\frac{1}{4}x^4ag+\frac{1}{4}be\,x^4+\frac{1}{3}x^3af+\frac{1}{3}x^3bd+$

```
[In] int((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)
```

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(b*f+c*d)*x^5+1/6
*(b*g+c*e)*x^6+1/7*c*f*x^7+1/8*c*g*x^8
```

Fricas [A] (verification not implemented)

none

Time = 0.23 (sec) , antiderivative size = 74, normalized size of antiderivative = 0.84

$$\int (d+ex+fx^2+gx^3)\,(a+bx^2+cx^4)\;dx = \frac{1}{8}\,cgx^8+\frac{1}{7}\,cfx^7+\frac{1}{6}\,(ce+bg)x^6 + \frac{1}{5}\,(cd+bf)x^5+\frac{1}{4}\,(be+ag)x^4 + \frac{1}{2}\,aex^2+\frac{1}{3}\,(bd+af)x^3+adx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="fricas")
[Out] 1/8*c*g*x^8 + 1/7*c*f*x^7 + 1/6*(c*e + b*g)*x^6 + 1/5*(c*d + b*f)*x^5 + 1/4
*(b*e + a*g)*x^4 + 1/2*a*e*x^2 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.94

$$\int (d+ex+fx^2+gx^3)\,(a+bx^2+cx^4)\;dx = adx+\frac{aex^2}{2}+\frac{cfx^7}{7}+\frac{cgx^8}{8} + x^6\left(\frac{bg}{6}+\frac{ce}{6}\right)+x^5\left(\frac{bf}{5}+\frac{cd}{5}\right) + x^4\left(\frac{ag}{4}+\frac{be}{4}\right)+x^3\left(\frac{af}{3}+\frac{bd}{3}\right)$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)*(c*x**4+b*x**2+a),x)
[Out] a*d*x + a*e*x**2/2 + c*f*x**7/7 + c*g*x**8/8 + x**6*(b*g/6 + c*e/6) + x**5*
(b*f/5 + c*d/5) + x**4*(a*g/4 + b*e/4) + x**3*(a*f/3 + b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 74, normalized size of antiderivative = 0.84

$$\int (d+ex+fx^2+gx^3)\,(a+bx^2+cx^4)\;dx = \frac{1}{8}\,cgx^8+\frac{1}{7}\,cfx^7+\frac{1}{6}\,(ce+bg)x^6 + \frac{1}{5}\,(cd+bf)x^5+\frac{1}{4}\,(be+ag)x^4 + \frac{1}{2}\,aex^2+\frac{1}{3}\,(bd+af)x^3+adx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] 1/8*c*g*x^8 + 1/7*c*f*x^7 + 1/6*(c*e + b*g)*x^6 + 1/5*(c*d + b*f)*x^5 + 1/4
*(b*e + a*g)*x^4 + 1/2*a*e*x^2 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.93

$$\int (d + ex + fx^2 + gx^3)(a + bx^2 + cx^4)\,dx = \frac{1}{8}cgx^8 + \frac{1}{7}cfx^7 + \frac{1}{6}cex^6 + \frac{1}{6}bgx^6 + \frac{1}{5}cdx^5 + \frac{1}{5}bfx^5 + \frac{1}{4}bex^4 + \frac{1}{4}agx^4 + \frac{1}{3}bdx^3 + \frac{1}{3}afx^3 + \frac{1}{2}aex^2 + adx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a),x, algorithm="giac")
```

```
[Out] 1/8*c*g*x^8 + 1/7*c*f*x^7 + 1/6*c*e*x^6 + 1/6*b*g*x^6 + 1/5*c*d*x^5 + 1/5*b
*f*x^5 + 1/4*b*e*x^4 + 1/4*a*g*x^4 + 1/3*b*d*x^3 + 1/3*a*f*x^3 + 1/2*a*e*x^
2 + a*d*x
```

Mupad [B] (verification not implemented)

Time = 7.85 (sec) , antiderivative size = 78, normalized size of antiderivative = 0.89

$$\int (d + ex + fx^2 + gx^3)(a + bx^2 + cx^4)\,dx = \frac{c\,g\,x^8}{8} + \frac{c\,f\,x^7}{7} + \left(\frac{c\,e}{6} + \frac{b\,g}{6}\right)x^6 + \left(\frac{c\,d}{5} + \frac{b\,f}{5}\right)x^5 + \left(\frac{b\,e}{4} + \frac{a\,g}{4}\right)x^4 + \left(\frac{b\,d}{3} + \frac{a\,f}{3}\right)x^3 + \frac{a\,e\,x^2}{2} + a\,d\,x$$

```
[In] int((a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3),x)
```

```
[Out] x^3*((b*d)/3 + (a*f)/3) + x^4*((b*e)/4 + (a*g)/4) + x^5*((c*d)/5 + (b*f)/5)
 + x^6*((c*e)/6 + (b*g)/6) + (c*g*x^8)/8 + a*d*x + (a*e*x^2)/2 + (c*f*x^7)/
7
```

3.4 $\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx$

Optimal result

Integrand size = 33, antiderivative size = 105

$$\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx = adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}(be+ag)x^4+\frac{1}{5}(cd+bf+ah)x^5+\frac{1}{6}(ce+bg)x^6+\frac{1}{7}(cf+bh)x^7+\frac{1}{8}cgx^8+\frac{1}{9}chx^9$$

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(a*h+b*f+c*d)*x^5
+1/6*(b*g+c*e)*x^6+1/7*(b*h+c*f)*x^7+1/8*c*g*x^8+1/9*c*h*x^9
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 105, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.030, Rules used = {1685}

$$\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx = \frac{1}{5}x^5(ah+bf+cd)+\frac{1}{3}x^3(af+bd)+\frac{1}{4}x^4(ag+be)+adx+\frac{1}{2}aex^2+\frac{1}{6}x^6(bg+ce)+\frac{1}{7}x^7(bh+cf)+\frac{1}{8}cgx^8+\frac{1}{9}chx^9$$

```
[In] Int[(a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f + a*h)*x^5)/5 + ((c*e + b*g)*x^6)/6 + ((c*f + b*h)*x^7)/7 + (c*g*x^8)/8
+ (c*h*x^9)/9
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(ad + aex + (bd + af)x^2 + (be + ag)x^3 + (cd + bf + ah)x^4 + (ce + bg)x^5 \right. \\ &\qquad \left. + (cf + bh)x^6 + cgx^7 + chx^8\right) dx \\ &= adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}(be + ag)x^4 + \frac{1}{5}(cd + bf + ah)x^5 \\ &\quad + \frac{1}{6}(ce + bg)x^6 + \frac{1}{7}(cf + bh)x^7 + \frac{1}{8}cgx^8 + \frac{1}{9}chx^9\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 105, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \left(a + bx^2 + cx^4\right)\left(d + ex + fx^2 + gx^3 + hx^4\right) dx \\ &= adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}(be + ag)x^4 + \frac{1}{5}(cd + bf + ah)x^5 \\ &\quad + \frac{1}{6}(ce + bg)x^6 + \frac{1}{7}(cf + bh)x^7 + \frac{1}{8}cgx^8 + \frac{1}{9}chx^9\end{aligned}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4),x]
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f + a*h)*x^5)/5 + ((c*e + b*g)*x^6)/6 + ((c*f + b*h)*x^7)/7 + (c*g*x^8)/8
+ (c*h*x^9)/9
```

Maple [A] (verified)

Time = 0.23 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.86

method	result
default	$adx + \frac{ae\,x^2}{2} + \frac{(af+bd)x^3}{3} + \frac{(ag+be)x^4}{4} + \frac{(ah+bf+cd)x^5}{5} + \frac{(bg+ec)x^6}{6} + \frac{(bh+cf)x^7}{7} + \frac{cg\,x^8}{8} + \frac{ch\,x^9}{9}$
norman	$\frac{ch\,x^9}{9} + \frac{cg\,x^8}{8} + \left(\frac{bh}{7} + \frac{cf}{7}\right)x^7 + \left(\frac{bg}{6} + \frac{ec}{6}\right)x^6 + \left(\frac{ah}{5} + \frac{bf}{5} + \frac{cd}{5}\right)x^5 + \left(\frac{ag}{4} + \frac{be}{4}\right)x^4 + \left(\frac{af}{3} + \frac{bd}{3}\right)x^3$
gosper	$\frac{1}{9}ch\,x^9 + \frac{1}{8}cg\,x^8 + \frac{1}{7}x^7bh + \frac{1}{7}cf\,x^7 + \frac{1}{6}x^6bg + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5ah + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}x^4ag +$
risch	$\frac{1}{9}ch\,x^9 + \frac{1}{8}cg\,x^8 + \frac{1}{7}x^7bh + \frac{1}{7}cf\,x^7 + \frac{1}{6}x^6bg + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5ah + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}x^4ag +$
parallelrisch	$\frac{1}{9}ch\,x^9 + \frac{1}{8}cg\,x^8 + \frac{1}{7}x^7bh + \frac{1}{7}cf\,x^7 + \frac{1}{6}x^6bg + \frac{1}{6}ce\,x^6 + \frac{1}{5}x^5ah + \frac{1}{5}x^5bf + \frac{1}{5}cd\,x^5 + \frac{1}{4}x^4ag +$

```
[In] int((c*x^4+b*x^2+a)*(h*x^4+g*x^3+f*x^2+e*x+d),x,method=_RETURNVERBOSE)
```

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(a*h+b*f+c*d)*x^5
+1/6*(b*g+c*e)*x^6+1/7*(b*h+c*f)*x^7+1/8*c*g*x^8+1/9*c*h*x^9
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 89, normalized size of antiderivative = 0.85

$$\begin{aligned}
&\int \left(a+bx^2+cx^4\right)\left(d+ex+fx^2+gx^3+hx^4\right)\,dx \\
&= \frac{1}{9}\,chx^9+\frac{1}{8}\,cgx^8+\frac{1}{7}\,(cf+bh)x^7+\frac{1}{6}\,(ce+bg)x^6+\frac{1}{5}\,(cd+bf+ah)x^5 \\
&\quad+\frac{1}{4}\,(be+ag)x^4+\frac{1}{2}\,aex^2+\frac{1}{3}\,(bd+af)x^3+adx
\end{aligned}$$

```
[In] integrate((c*x^4+b*x^2+a)*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="fricas")
```

```
[Out] 1/9*c*h*x^9 + 1/8*c*g*x^8 + 1/7*(c*f + b*h)*x^7 + 1/6*(c*e + b*g)*x^6 + 1/5
*(c*d + b*f + a*h)*x^5 + 1/4*(b*e + a*g)*x^4 + 1/2*a*e*x^2 + 1/3*(b*d + a*f
)*x^3 + a*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 102, normalized size of antiderivative = 0.97

$$\begin{aligned}
&\int \left(a+bx^2+cx^4\right)\left(d+ex+fx^2+gx^3+hx^4\right)\,dx \\
&= adx+\frac{aex^2}{2}+\frac{cgx^8}{8}+\frac{chx^9}{9}+x^7\left(\frac{bh}{7}+\frac{cf}{7}\right)+x^6\left(\frac{bg}{6}+\frac{ce}{6}\right) \\
&\quad+x^5\left(\frac{ah}{5}+\frac{bf}{5}+\frac{cd}{5}\right)+x^4\left(\frac{ag}{4}+\frac{be}{4}\right)+x^3\left(\frac{af}{3}+\frac{bd}{3}\right)
\end{aligned}$$

```
[In] integrate((c*x**4+b*x**2+a)*(h*x**4+g*x**3+f*x**2+e*x+d),x)
```

```
[Out] a*d*x + a*e*x**2/2 + c*g*x**8/8 + c*h*x**9/9 + x**7*(b*h/7 + c*f/7) + x**6*
(b*g/6 + c*e/6) + x**5*(a*h/5 + b*f/5 + c*d/5) + x**4*(a*g/4 + b*e/4) + x**
3*(a*f/3 + b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 89, normalized size of antiderivative = 0.85

$$\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx = \frac{1}{9}chx^9+\frac{1}{8}cgx^8+\frac{1}{7}(cf+bh)x^7+\frac{1}{6}(ce+bg)x^6+\frac{1}{5}(cd+bf+ah)x^5 + \frac{1}{4}(be+ag)x^4+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+adx$$

```
[In] integrate((c*x^4+b*x^2+a)*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="maxima")
```

```
[Out] 1/9*c*h*x^9 + 1/8*c*g*x^8 + 1/7*(c*f + b*h)*x^7 + 1/6*(c*e + b*g)*x^6 + 1/5
*(c*d + b*f + a*h)*x^5 + 1/4*(b*e + a*g)*x^4 + 1/2*a*e*x^2 + 1/3*(b*d + a*f
)*x^3 + a*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 103, normalized size of antiderivative = 0.98

$$\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx = \frac{1}{9}chx^9+\frac{1}{8}cgx^8+\frac{1}{7}cfx^7+\frac{1}{7}bhx^7+\frac{1}{6}cex^6+\frac{1}{6}bgx^6+\frac{1}{5}cdx^5+\frac{1}{5}bfx^5 + \frac{1}{5}ahx^5+\frac{1}{4}bex^4+\frac{1}{4}agx^4+\frac{1}{3}bdx^3+\frac{1}{3}afx^3+\frac{1}{2}aex^2+adx$$

```
[In] integrate((c*x^4+b*x^2+a)*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="giac")
```

```
[Out] 1/9*c*h*x^9 + 1/8*c*g*x^8 + 1/7*c*f*x^7 + 1/7*b*h*x^7 + 1/6*c*e*x^6 + 1/6*b
*g*x^6 + 1/5*c*d*x^5 + 1/5*b*f*x^5 + 1/5*a*h*x^5 + 1/4*b*e*x^4 + 1/4*a*g*x^
4 + 1/3*b*d*x^3 + 1/3*a*f*x^3 + 1/2*a*e*x^2 + a*d*x
```

Mupad [B] (verification not implemented)

Time = 7.90 (sec) , antiderivative size = 95, normalized size of antiderivative = 0.90

$$\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4)\,dx = \frac{c\,h\,x^9}{9}+\frac{c\,g\,x^8}{8}+\left(\frac{c\,f}{7}+\frac{b\,h}{7}\right)x^7+\left(\frac{c\,e}{6}+\frac{b\,g}{6}\right)x^6+\left(\frac{c\,d}{5}+\frac{b\,f}{5}+\frac{a\,h}{5}\right)x^5 + \left(\frac{b\,e}{4}+\frac{a\,g}{4}\right)x^4+\left(\frac{b\,d}{3}+\frac{a\,f}{3}\right)x^3+\frac{a\,e\,x^2}{2}+a\,d\,x$$

```
[In] int((a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4),x)
```

```
[Out] x^5*((c*d)/5 + (b*f)/5 + (a*h)/5) + x^3*((b*d)/3 + (a*f)/3) + x^4*((b*e)/4
+ (a*g)/4) + x^6*((c*e)/6 + (b*g)/6) + x^7*((c*f)/7 + (b*h)/7) + (c*g*x^8)/
8 + (c*h*x^9)/9 + a*d*x + (a*e*x^2)/2
```

3.5 $\int (a+bx^2+cx^4)(d+ex+fx^2+gx^3+hx^4+ix^5)\,dx$

Optimal result

Integrand size = 38, antiderivative size = 122

$$\begin{aligned}\int \left(a+bx^2+cx^4\right)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)\,dx &= adx+\frac{1}{2}aex^2+\frac{1}{3}(bd+af)x^3+\frac{1}{4}(be+ag)x^4+\frac{1}{5}(cd+bf+ah)x^5\\ &\quad+\frac{1}{6}(ce+bg+ai)x^6+\frac{1}{7}(cf+bh)x^7+\frac{1}{8}(cg+bi)x^8+\frac{1}{9}chx^9+\frac{1}{10}cix^{10}\end{aligned}$$

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(a*h+b*f+c*d)*x^5
+1/6*(a*i+b*g+c*e)*x^6+1/7*(b*h+c*f)*x^7+1/8*(b*i+c*g)*x^8+1/9*c*h*x^9+1/10
*c*i*x^10
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.026$, Rules used = {1685}

$$\begin{aligned}\int \left(a+bx^2+cx^4\right)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)\,dx &= \frac{1}{5}x^5(ah+bf+cd)+\frac{1}{6}x^6(ai+bg+ce)+\frac{1}{3}x^3(af+bd)+\frac{1}{4}x^4(ag+be)\\ &\quad+adx+\frac{1}{2}aex^2+\frac{1}{7}x^7(bh+cf)+\frac{1}{8}x^8(bi+cg)+\frac{1}{9}chx^9+\frac{1}{10}cix^{10}\end{aligned}$$

```
[In] Int[(a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5),x]
```

```
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f + a*h)*x^5)/5 + ((c*e + b*g + a*i)*x^6)/6 + ((c*f + b*h)*x^7)/7 + ((c*g
+ b*i)*x^8)/8 + (c*h*x^9)/9 + (c*i*x^10)/10
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(ad + aex + (bd + af)x^2 + (be + ag)x^3 + (cd + bf + ah)x^4 + (ce + bg + ai)x^5 \right. \\ &\qquad \left. + (cf + bh)x^6 + (cg + bi)x^7 + chx^8 + cix^9\right) dx \\ &= adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}(be + ag)x^4 + \frac{1}{5}(cd + bf + ah)x^5 \\ &\quad + \frac{1}{6}(ce + bg + ai)x^6 + \frac{1}{7}(cf + bh)x^7 + \frac{1}{8}(cg + bi)x^8 + \frac{1}{9}chx^9 + \frac{1}{10}cix^{10}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \left(a + bx^2 + cx^4\right)\left(d + ex + fx^2 + gx^3 + hx^4 + ix^5\right) dx \\ &= adx + \frac{1}{2}aex^2 + \frac{1}{3}(bd + af)x^3 + \frac{1}{4}(be + ag)x^4 + \frac{1}{5}(cd + bf + ah)x^5 \\ &\quad + \frac{1}{6}(ce + bg + ai)x^6 + \frac{1}{7}(cf + bh)x^7 + \frac{1}{8}(cg + bi)x^8 + \frac{1}{9}chx^9 + \frac{1}{10}cix^{10}\end{aligned}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5),x]
[Out] a*d*x + (a*e*x^2)/2 + ((b*d + a*f)*x^3)/3 + ((b*e + a*g)*x^4)/4 + ((c*d + b
*f + a*h)*x^5)/5 + ((c*e + b*g + a*i)*x^6)/6 + ((c*f + b*h)*x^7)/7 + ((c*g
+ b*i)*x^8)/8 + (c*h*x^9)/9 + (c*i*x^10)/10
```

Maple [A] (verified)

Time = 1.91 (sec) , antiderivative size = 105, normalized size of antiderivative = 0.86

method	result
default	$adx+\frac{ae\,x^{2}}{2}+\frac{(af+bd)x^{3}}{3}+\frac{(ag+be)x^{4}}{4}+\frac{(ah+bf+cd)x^{5}}{5}+\frac{(ai+bg+ec)x^{6}}{6}+\frac{(bh+cf)x^{7}}{7}+\frac{(bi+gc)x^{8}}{8}+\frac{ch\,x^{9}}{9}$
norman	$\frac{ci\,x^{10}}{10}+\frac{ch\,x^{9}}{9}+\left(\frac{bi}{8}+\frac{gc}{8}\right)x^{8}+\left(\frac{bh}{7}+\frac{cf}{7}\right)x^{7}+\left(\frac{ai}{6}+\frac{bg}{6}+\frac{ec}{6}\right)x^{6}+\left(\frac{ah}{5}+\frac{bf}{5}+\frac{cd}{5}\right)x^{5}+\left(\frac{ag}{4}+\right.$
gosper	$\frac{1}{10}ci\,x^{10}+\frac{1}{9}ch\,x^{9}+\frac{1}{8}x^{8}bi+\frac{1}{8}cg\,x^{8}+\frac{1}{7}x^{7}bh+\frac{1}{7}cf\,x^{7}+\frac{1}{6}x^{6}ai+\frac{1}{6}x^{6}bg+\frac{1}{6}ce\,x^{6}+\frac{1}{5}x^{5}ah+$
risch	$\frac{1}{10}ci\,x^{10}+\frac{1}{9}ch\,x^{9}+\frac{1}{8}x^{8}bi+\frac{1}{8}cg\,x^{8}+\frac{1}{7}x^{7}bh+\frac{1}{7}cf\,x^{7}+\frac{1}{6}x^{6}ai+\frac{1}{6}x^{6}bg+\frac{1}{6}ce\,x^{6}+\frac{1}{5}x^{5}ah+$
parallelrisch	$\frac{1}{10}ci\,x^{10}+\frac{1}{9}ch\,x^{9}+\frac{1}{8}x^{8}bi+\frac{1}{8}cg\,x^{8}+\frac{1}{7}x^{7}bh+\frac{1}{7}cf\,x^{7}+\frac{1}{6}x^{6}ai+\frac{1}{6}x^{6}bg+\frac{1}{6}ce\,x^{6}+\frac{1}{5}x^{5}ah+$

```
[In] int((c*x^4+b*x^2+a)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d),x,method=_RETURNVERBOSE
)
```

```
[Out] a*d*x+1/2*a*e*x^2+1/3*(a*f+b*d)*x^3+1/4*(a*g+b*e)*x^4+1/5*(a*h+b*f+c*d)*x^5
+1/6*(a*i+b*g+c*e)*x^6+1/7*(b*h+c*f)*x^7+1/8*(b*i+c*g)*x^8+1/9*c*h*x^9+1/10
*c*i*x^10
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 104, normalized size of antiderivative = 0.85

$$\int (a+bx^2+cx^4)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)\,dx = \frac{1}{10}\,cix^{10}+\frac{1}{9}\,chx^{9}+\frac{1}{8}\,(cg+bi)x^{8}+\frac{1}{7}\,(cf+bh)x^{7}+\frac{1}{6}\,(ce+bg+ai)x^{6} + \frac{1}{5}\,(cd+bf+ah)x^{5}+\frac{1}{4}\,(be+ag)x^{4}+\frac{1}{2}\,aex^{2}+\frac{1}{3}\,(bd+af)x^{3}+adx$$

```
[In] integrate((c*x^4+b*x^2+a)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="fri
cas")
```

```
[Out] 1/10*c*i*x^10 + 1/9*c*h*x^9 + 1/8*(c*g + b*i)*x^8 + 1/7*(c*f + b*h)*x^7 + 1
/6*(c*e + b*g + a*i)*x^6 + 1/5*(c*d + b*f + a*h)*x^5 + 1/4*(b*e + a*g)*x^4
+ 1/2*a*e*x^2 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 121, normalized size of antiderivative = 0.99

$$\int (a+bx^2+cx^4)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)\,dx = adx+\frac{aex^2}{2}+\frac{chx^9}{9}+\frac{cix^{10}}{10}+x^8\left(\frac{bi}{8}+\frac{cg}{8}\right)+x^7\left(\frac{bh}{7}+\frac{cf}{7}\right) + x^6\left(\frac{ai}{6}+\frac{bg}{6}+\frac{ce}{6}\right)+x^5\left(\frac{ah}{5}+\frac{bf}{5}+\frac{cd}{5}\right)+x^4\left(\frac{ag}{4}+\frac{be}{4}\right)+x^3\left(\frac{af}{3}+\frac{bd}{3}\right)$$

```
[In] integrate((c*x**4+b*x**2+a)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d),x)
```

```
[Out] a*d*x + a*e*x**2/2 + c*h*x**9/9 + c*i*x**10/10 + x**8*(b*i/8 + c*g/8) + x**
7*(b*h/7 + c*f/7) + x**6*(a*i/6 + b*g/6 + c*e/6) + x**5*(a*h/5 + b*f/5 + c*
d/5) + x**4*(a*g/4 + b*e/4) + x**3*(a*f/3 + b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 104, normalized size of antiderivative = 0.85

$$\int (a + bx^2 + cx^4) (d + ex + fx^2 + gx^3 + hx^4 + ix^5) \, dx$$
$$= \frac{1}{10} cix^{10} + \frac{1}{9} chx^9 + \frac{1}{8} (cg + bi)x^8 + \frac{1}{7} (cf + bh)x^7 + \frac{1}{6} (ce + bg + ai)x^6 + \frac{1}{5} (cd + bf + ah)x^5 + \frac{1}{4} (be + ag)x^4 + \frac{1}{2} aex^2 + \frac{1}{3} (bd + af)x^3 + adx$$

```
[In] integrate((c*x^4+b*x^2+a)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="max
ima")
```

```
[Out] 1/10*c*i*x^10 + 1/9*c*h*x^9 + 1/8*(c*g + b*i)*x^8 + 1/7*(c*f + b*h)*x^7 + 1
/6*(c*e + b*g + a*i)*x^6 + 1/5*(c*d + b*f + a*h)*x^5 + 1/4*(b*e + a*g)*x^4
+ 1/2*a*e*x^2 + 1/3*(b*d + a*f)*x^3 + a*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 124, normalized size of antiderivative = 1.02

$$\int (a + bx^2 + cx^4) (d + ex + fx^2 + gx^3 + hx^4 + ix^5) \, dx$$
$$= \frac{1}{10} cix^{10} + \frac{1}{9} chx^9 + \frac{1}{8} cgx^8 + \frac{1}{8} bix^8 + \frac{1}{7} cfx^7 + \frac{1}{7} bhx^7 + \frac{1}{6} cex^6 + \frac{1}{6} bgx^6 + \frac{1}{6} aix^6 + \frac{1}{5} cdx^5 + \frac{1}{5} bfx^5 + \frac{1}{5} ahx^5 + \frac{1}{4} bex^4 + \frac{1}{4} agx^4 + \frac{1}{3} bdx^3 + \frac{1}{3} afx^3 + \frac{1}{2} aex^2 + adx$$

```
[In] integrate((c*x^4+b*x^2+a)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="gia
c")
```

```
[Out] 1/10*c*i*x^10 + 1/9*c*h*x^9 + 1/8*c*g*x^8 + 1/8*b*i*x^8 + 1/7*c*f*x^7 + 1/7
*b*h*x^7 + 1/6*c*e*x^6 + 1/6*b*g*x^6 + 1/6*a*i*x^6 + 1/5*c*d*x^5 + 1/5*b*f*
x^5 + 1/5*a*h*x^5 + 1/4*b*e*x^4 + 1/4*a*g*x^4 + 1/3*b*d*x^3 + 1/3*a*f*x^3 +
 1/2*a*e*x^2 + a*d*x
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 112, normalized size of antiderivative = 0.92

$$\int (a+bx^2+cx^4)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)\,dx = \frac{c\,i\,x^{10}}{10}+\frac{c\,h\,x^9}{9}+\left(\frac{c\,g}{8}+\frac{b\,i}{8}\right)x^8+\left(\frac{c\,f}{7}+\frac{b\,h}{7}\right)x^7+\left(\frac{c\,e}{6}+\frac{b\,g}{6}+\frac{a\,i}{6}\right)x^6 +\left(\frac{c\,d}{5}+\frac{b\,f}{5}+\frac{a\,h}{5}\right)x^5+\left(\frac{b\,e}{4}+\frac{a\,g}{4}\right)x^4+\left(\frac{b\,d}{3}+\frac{a\,f}{3}\right)x^3+\frac{a\,e\,x^2}{2}+a\,d\,x$$

```
[In] int((a + b*x^2 + c*x^4)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5),x)
```

```
[Out] x^5*((c*d)/5 + (b*f)/5 + (a*h)/5) + x^6*((c*e)/6 + (b*g)/6 + (a*i)/6) + x^3
*((b*d)/3 + (a*f)/3) + x^4*((b*e)/4 + (a*g)/4) + x^7*((c*f)/7 + (b*h)/7) +
x^8*((c*g)/8 + (b*i)/8) + (c*h*x^9)/9 + (c*i*x^10)/10 + a*d*x + (a*e*x^2)/2
```

3.6 $\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx$

Optimal result

Integrand size = 20, antiderivative size = 112

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx=a^2dx+\frac{1}{2}a^2ex^2+\frac{2}{3}abdx^3+\frac{1}{2}abex^4+\frac{1}{5}\left(b^2+2ac\right)dx^5+\frac{1}{6}\left(b^2+2ac\right)ex^6+\frac{2}{7}bcdx^7+\frac{1}{4}bcex^8+\frac{1}{9}c^2dx^9+\frac{1}{10}c^2ex^{10}$$

```
[Out] a^2*d*x+1/2*a^2*e*x^2+2/3*a*b*d*x^3+1/2*a*b*e*x^4+1/5*(2*a*c+b^2)*d*x^5+1/6
*(2*a*c+b^2)*e*x^6+2/7*b*c*d*x^7+1/4*b*c*e*x^8+1/9*c^2*d*x^9+1/10*c^2*e*x^1
0
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 112, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.050, Rules used = {1685}

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx=a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{5}dx^5\left(2ac+b^2\right)+\frac{1}{6}ex^6\left(2ac+b^2\right)+\frac{2}{3}abdx^3+\frac{1}{2}abex^4+\frac{2}{7}bcdx^7+\frac{1}{4}bcex^8+\frac{1}{9}c^2dx^9+\frac{1}{10}c^2ex^{10}$$

```
[In] Int[(d + e*x)*(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (2*a*b*d*x^3)/3 + (a*b*e*x^4)/2 + ((b^2 + 2*a*c)*
d*x^5)/5 + ((b^2 + 2*a*c)*e*x^6)/6 + (2*b*c*d*x^7)/7 + (b*c*e*x^8)/4 + (c^2
*d*x^9)/9 + (c^2*e*x^10)/10
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
```

```
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\text{integral} = \int \left(a^2d + a^2ex + 2abdx^2 + 2abex^3 + \left(b^2+2ac\right)dx^4 + \left(b^2+2ac\right)ex^5 + 2bcdx^6 + 2bcex^7 + c^2dx^8 + c^2ex^9\right)\,dx$$

$$= a^2dx + \frac{1}{2}a^2ex^2 + \frac{2}{3}abdx^3 + \frac{1}{2}abex^4 + \frac{1}{5}\left(b^2+2ac\right)dx^5 + \frac{1}{6}\left(b^2+2ac\right)ex^6 + \frac{2}{7}bcdx^7 + \frac{1}{4}bcex^8 + \frac{1}{9}c^2dx^9 + \frac{1}{10}c^2ex^{10}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 97, normalized size of antiderivative = 0.87

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx = \frac{630a^2x(2d+ex)+42b^2x^5(6d+5ex)+45bcx^7(8d+7ex)+14c^2x^9(10d+9ex)+42a(5bx^3(4d+3ex)+2}{1260}$$

```
[In] Integrate[(d + e*x)*(a + b*x^2 + c*x^4)^2,x]
[Out] (630*a^2*x*(2*d + e*x) + 42*b^2*x^5*(6*d + 5*e*x) + 45*b*c*x^7*(8*d + 7*e*x
) + 14*c^2*x^9*(10*d + 9*e*x) + 42*a*(5*b*x^3*(4*d + 3*e*x) + 2*c*x^5*(6*d
+ 5*e*x)))/1260
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 95, normalized size of antiderivative = 0.85

method	result
default	$a^2dx + \frac{a^2e\,x^2}{2} + \frac{2x^3dab}{3} + \frac{abe\,x^4}{2} + \frac{(2ac+b^2)d\,x^5}{5} + \frac{(2ac+b^2)e\,x^6}{6} + \frac{2x^7bcd}{7} + \frac{bce\,x^8}{4} + \frac{c^2d\,x^9}{9} + \frac{c^2e\,x^{10}}{10}$
norman	$\frac{c^2e\,x^{10}}{10} + \frac{c^2d\,x^9}{9} + \frac{bce\,x^8}{4} + \frac{2x^7bcd}{7} + \left(\frac{1}{3}ace + \frac{1}{6}b^2e\right)x^6 + \left(\frac{2}{5}acd + \frac{1}{5}b^2d\right)x^5 + \frac{abe\,x^4}{2} + \frac{2x^3dab}{3} +$
gosper	$\frac{1}{10}c^2e\,x^{10} + \frac{1}{9}c^2d\,x^9 + \frac{1}{4}bce\,x^8 + \frac{2}{7}x^7bcd + \frac{1}{3}x^6ace + \frac{1}{6}x^6b^2e + \frac{2}{5}acd\,x^5 + \frac{1}{5}x^5b^2d + \frac{1}{2}abe\,x^4 +$
risch	$\frac{1}{10}c^2e\,x^{10} + \frac{1}{9}c^2d\,x^9 + \frac{1}{4}bce\,x^8 + \frac{2}{7}x^7bcd + \frac{1}{3}x^6ace + \frac{1}{6}x^6b^2e + \frac{2}{5}acd\,x^5 + \frac{1}{5}x^5b^2d + \frac{1}{2}abe\,x^4 +$
parallelrisch	$\frac{1}{10}c^2e\,x^{10} + \frac{1}{9}c^2d\,x^9 + \frac{1}{4}bce\,x^8 + \frac{2}{7}x^7bcd + \frac{1}{3}x^6ace + \frac{1}{6}x^6b^2e + \frac{2}{5}acd\,x^5 + \frac{1}{5}x^5b^2d + \frac{1}{2}abe\,x^4 +$

```
[In] int((e*x+d)*(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] a^2*d*x+1/2*a^2*e*x^2+2/3*x^3*d*a*b+1/2*a*b*e*x^4+1/5*(2*a*c+b^2)*d*x^5+1/6
*(2*a*c+b^2)*e*x^6+2/7*x^7*b*c*d+1/4*b*c*e*x^8+1/9*c^2*d*x^9+1/10*c^2*e*x^1
0
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 94, normalized size of antiderivative = 0.84

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx = \frac{1}{10}\,c^2ex^{10}+\frac{1}{9}\,c^2dx^9+\frac{1}{4}\,bcex^8+\frac{2}{7}\,bcdx^7+\frac{1}{6}\left(b^2+2\,ac\right)ex^6+\frac{1}{2}\,abex^4+\frac{1}{5}\left(b^2+2\,ac\right)dx^5+\frac{2}{3}\,abdx^3+\frac{1}{2}\,a^2ex^2+a^2dx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] 1/10*c^2*e*x^10 + 1/9*c^2*d*x^9 + 1/4*b*c*e*x^8 + 2/7*b*c*d*x^7 + 1/6*(b^2
+ 2*a*c)*e*x^6 + 1/2*a*b*e*x^4 + 1/5*(b^2 + 2*a*c)*d*x^5 + 2/3*a*b*d*x^3 +
1/2*a^2*e*x^2 + a^2*d*x
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 116, normalized size of antiderivative = 1.04

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx+\frac{a^2ex^2}{2}+\frac{2abdx^3}{3}+\frac{abex^4}{2}+\frac{2bcdx^7}{7}+\frac{bcex^8}{4}+\frac{c^2dx^9}{9}+\frac{c^2ex^{10}}{10}+x^6\left(\frac{ace}{3}+\frac{b^2e}{6}\right)+x^5\cdot\left(\frac{2acd}{5}+\frac{b^2d}{5}\right)$$

```
[In] integrate((e*x+d)*(c*x**4+b*x**2+a)**2,x)
```

```
[Out] a**2*d*x + a**2*e*x**2/2 + 2*a*b*d*x**3/3 + a*b*e*x**4/2 + 2*b*c*d*x**7/7 +
 b*c*e*x**8/4 + c**2*d*x**9/9 + c**2*e*x**10/10 + x**6*(a*c*e/3 + b**2*e/6)
 + x**5*(2*a*c*d/5 + b**2*d/5)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 94, normalized size of antiderivative = 0.84

$$\int (d+ex)\left(a+bx^2+cx^4\right)^2\,dx = \frac{1}{10}\,c^2ex^{10}+\frac{1}{9}\,c^2dx^9+\frac{1}{4}\,bcex^8+\frac{2}{7}\,bcdx^7+\frac{1}{6}\left(b^2+2\,ac\right)ex^6+\frac{1}{2}\,abex^4+\frac{1}{5}\left(b^2+2\,ac\right)dx^5+\frac{2}{3}\,abdx^3+\frac{1}{2}\,a^2ex^2+a^2dx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] 1/10*c^2*e*x^10 + 1/9*c^2*d*x^9 + 1/4*b*c*e*x^8 + 2/7*b*c*d*x^7 + 1/6*(b^2
+ 2*a*c)*e*x^6 + 1/2*a*b*e*x^4 + 1/5*(b^2 + 2*a*c)*d*x^5 + 2/3*a*b*d*x^3 +
1/2*a^2*e*x^2 + a^2*d*x
```

Giac [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 100, normalized size of antiderivative = 0.89

$$\int (d+ex) \left(a+bx^2+cx^4\right)^2 \, dx = \frac{1}{10}\,c^2ex^{10}+\frac{1}{9}\,c^2dx^9+\frac{1}{4}\,bcex^8+\frac{2}{7}\,bcdx^7+\frac{1}{6}\,b^2ex^6+\frac{1}{3}\,acex^6 +\frac{1}{5}\,b^2dx^5+\frac{2}{5}\,acdx^5+\frac{1}{2}\,abex^4+\frac{2}{3}\,abdx^3+\frac{1}{2}\,a^2ex^2+a^2dx$$

```
[In] integrate((e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/10*c^2*e*x^10 + 1/9*c^2*d*x^9 + 1/4*b*c*e*x^8 + 2/7*b*c*d*x^7 + 1/6*b^2*e
*x^6 + 1/3*a*c*e*x^6 + 1/5*b^2*d*x^5 + 2/5*a*c*d*x^5 + 1/2*a*b*e*x^4 + 2/3*
a*b*d*x^3 + 1/2*a^2*e*x^2 + a^2*d*x
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 94, normalized size of antiderivative = 0.84

$$\int (d+ex) \left(a+bx^2+cx^4\right)^2 \, dx = \frac{a^2\,e\,x^2}{2}+\frac{c^2\,d\,x^9}{9}+\frac{c^2\,e\,x^{10}}{10}+\frac{d\,x^5\,\left(b^2+2\,a\,c\right)}{5}+\frac{e\,x^6\,\left(b^2+2\,a\,c\right)}{6} +a^2\,d\,x+\frac{2\,a\,b\,d\,x^3}{3}+\frac{a\,b\,e\,x^4}{2}+\frac{2\,b\,c\,d\,x^7}{7}+\frac{b\,c\,e\,x^8}{4}$$

```
[In] int((d + e*x)*(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] (a^2*e*x^2)/2 + (c^2*d*x^9)/9 + (c^2*e*x^10)/10 + (d*x^5*(2*a*c + b^2))/5 +
 (e*x^6*(2*a*c + b^2))/6 + a^2*d*x + (2*a*b*d*x^3)/3 + (a*b*e*x^4)/2 + (2*b
*c*d*x^7)/7 + (b*c*e*x^8)/4
```

3.7 $\int \left(d+ex+fx^2\right)\left(a+bx^2+cx^4\right)^2\,dx$

Optimal result

Integrand size = 25, antiderivative size = 154

$$\int \left(d+ex+fx^2\right)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4 +\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6 +\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8 +\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}$$

```
[Out] a^2*d*x+1/2*a^2*e*x^2+1/3*a*(a*f+2*b*d)*x^3+1/2*a*b*e*x^4+1/5*(2*a*b*f+2*a*
c*d+b^2*d)*x^5+1/6*(2*a*c+b^2)*e*x^6+1/7*(2*a*c*f+b^2*f+2*b*c*d)*x^7+1/4*b*
c*e*x^8+1/9*c*(2*b*f+c*d)*x^9+1/10*c^2*e*x^10+1/11*c^2*f*x^11
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.040, Rules used = {1671}

$$\int \left(d+ex+fx^2\right)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{7}x^7\left(2acf+b^2f+2bcd\right) +\frac{1}{5}x^5\left(2abf+2acd+b^2d\right)+\frac{1}{6}ex^6\left(2ac+b^2\right) +\frac{1}{3}ax^3(af+2bd)+\frac{1}{2}abex^4+\frac{1}{9}cx^9(2bf+cd) +\frac{1}{4}bcex^8+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}$$

```
[In] Int[(d + e*x + f*x^2)*(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*b*e*x^4)/2 + ((b^2*d
 + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2 + 2*a*c)*e*x^6)/6 + ((2*b*c*d + b^2*f
+ 2*a*c*f)*x^7)/7 + (b*c*e*x^8)/4 + (c*(c*d + 2*b*f)*x^9)/9 + (c^2*e*x^10)/
10 + (c^2*f*x^11)/11
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(a^2d+a^2ex+a(2bd+af)x^2+2abex^3+\left(b^2d+2acd+2abf\right)x^4+\left(b^2+2ac\right)ex^5\right.\\&\qquad\left.+\left(2bcd+b^2f+2acf\right)x^6+2bcex^7+c(cd+2bf)x^8+c^2ex^9+c^2fx^{10}\right)\,dx\\&=a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6\\&\quad+\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00

$$\begin{aligned}\int \left(d+ex+fx^2\right)\left(a+bx^2+cx^4\right)^2\,dx &= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4\\&\quad+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6\\&\quad+\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8\\&\quad+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}\end{aligned}$$

```
[In] Integrate[(d + e*x + f*x^2)*(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*b*e*x^4)/2 + ((b^2*d
 + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2 + 2*a*c)*e*x^6)/6 + ((2*b*c*d + b^2*f
+ 2*a*c*f)*x^7)/7 + (b*c*e*x^8)/4 + (c*(c*d + 2*b*f)*x^9)/9 + (c^2*e*x^10)/
10 + (c^2*f*x^11)/11
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 139, normalized size of antiderivative = 0.90

method	result
default	$\frac{c^2 f x^{11}}{11}+\frac{c^2 e x^{10}}{10}+\frac{(2fbc+c^2d)x^9}{9}+\frac{bce\,x^8}{4}+\frac{(2bcd+f(2ac+b^2))x^7}{7}+\frac{(2ac+b^2)e\,x^6}{6}+\frac{(d(2ac+b^2)+2abf)x^5}{5}+ab$
norman	$\frac{c^2 f x^{11}}{11}+\frac{c^2 e x^{10}}{10}+\left(\frac{2}{9}fbc+\frac{1}{9}c^2d\right)x^9+\frac{bce\,x^8}{4}+\left(\frac{2}{7}acf+\frac{1}{7}b^2f+\frac{2}{7}bcd\right)x^7+\left(\frac{1}{3}ace+\frac{1}{6}b^2e\right)x^6+$
gosper	$\frac{1}{11}c^2f\,x^{11}+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}bce\,x^8+\frac{2}{7}x^7acf+\frac{1}{7}x^7b^2f+\frac{2}{7}x^7bcd+\frac{1}{3}x^6ace+$
risch	$\frac{1}{11}c^2f\,x^{11}+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}bce\,x^8+\frac{2}{7}x^7acf+\frac{1}{7}x^7b^2f+\frac{2}{7}x^7bcd+\frac{1}{3}x^6ace+$
parallelrisch	$\frac{1}{11}c^2f\,x^{11}+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}bce\,x^8+\frac{2}{7}x^7acf+\frac{1}{7}x^7b^2f+\frac{2}{7}x^7bcd+\frac{1}{3}x^6ace+$

```
[In] int((f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] 1/11*c^2*f*x^11+1/10*c^2*e*x^10+1/9*(2*b*c*f+c^2*d)*x^9+1/4*b*c*e*x^8+1/7*(
2*b*c*d+f*(2*a*c+b^2))*x^7+1/6*(2*a*c+b^2)*e*x^6+1/5*(d*(2*a*c+b^2)+2*a*b*f
)*x^5+1/2*a*b*e*x^4+1/3*(a^2*f+2*a*b*d)*x^3+1/2*a^2*e*x^2+a^2*d*x
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\begin{aligned}\int (d+ex+fx^2)\left(a+bx^2+cx^4\right)^2\,dx &= \frac{1}{11}c^2fx^{11}+\frac{1}{10}c^2ex^{10}+\frac{1}{4}bcex^8+\frac{1}{9}\left(c^2d+2\,bcf\right)x^9\\ &\quad+\frac{1}{6}\left(b^2+2\,ac\right)ex^6+\frac{1}{7}\left(2\,bcd+\left(b^2+2\,ac\right)f\right)x^7\\ &\quad+\frac{1}{2}abex^4+\frac{1}{5}\left(2\,abf+\left(b^2+2\,ac\right)d\right)x^5\\ &\quad+\frac{1}{2}a^2ex^2+a^2dx+\frac{1}{3}\left(2\,abd+a^2f\right)x^3\end{aligned}$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/4*b*c*e*x^8 + 1/9*(c^2*d + 2*b*c*f)*x
^9 + 1/6*(b^2 + 2*a*c)*e*x^6 + 1/7*(2*b*c*d + (b^2 + 2*a*c)*f)*x^7 + 1/2*a*
b*e*x^4 + 1/5*(2*a*b*f + (b^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + a^2*d*x + 1
/3*(2*a*b*d + a^2*f)*x^3
```

Sympy [A] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.07

$$\int (d+ex+fx^2)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx + \frac{a^2ex^2}{2} + \frac{abex^4}{2} + \frac{bcex^8}{4} + \frac{c^2ex^{10}}{10} + \frac{c^2fx^{11}}{11} + x^9\cdot\left(\frac{2bcf}{9} + \frac{c^2d}{9}\right) + x^7\cdot\left(\frac{2acf}{7} + \frac{b^2f}{7} + \frac{2bcd}{7}\right) + x^6\left(\frac{ace}{3} + \frac{b^2e}{6}\right) + x^5\cdot\left(\frac{2abf}{5} + \frac{2acd}{5} + \frac{b^2d}{5}\right) + x^3\left(\frac{a^2f}{3} + \frac{2abd}{3}\right)$$

```
[In] integrate((f*x**2+e*x+d)*(c*x**4+b*x**2+a)**2,x)
```

```
[Out] a**2*d*x + a**2*e*x**2/2 + a*b*e*x**4/2 + b*c*e*x**8/4 + c**2*e*x**10/10 +
c**2*f*x**11/11 + x**9*(2*b*c*f/9 + c**2*d/9) + x**7*(2*a*c*f/7 + b**2*f/7
+ 2*b*c*d/7) + x**6*(a*c*e/3 + b**2*e/6) + x**5*(2*a*b*f/5 + 2*a*c*d/5 + b*
*2*d/5) + x**3*(a**2*f/3 + 2*a*b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\int (d+ex+fx^2)\left(a+bx^2+cx^4\right)^2\,dx = \frac{1}{11}\,c^2fx^{11} + \frac{1}{10}\,c^2ex^{10} + \frac{1}{4}\,bcex^8 + \frac{1}{9}\,(c^2d + 2\,bcf)x^9 + \frac{1}{6}\,(b^2 + 2\,ac)ex^6 + \frac{1}{7}\,(2\,bcd + (b^2 + 2\,ac)f)x^7 + \frac{1}{2}\,abex^4 + \frac{1}{5}\,(2\,abf + (b^2 + 2\,ac)d)x^5 + \frac{1}{2}\,a^2ex^2 + a^2dx + \frac{1}{3}\,(2\,abd + a^2f)x^3$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/4*b*c*e*x^8 + 1/9*(c^2*d + 2*b*c*f)*x
^9 + 1/6*(b^2 + 2*a*c)*e*x^6 + 1/7*(2*b*c*d + (b^2 + 2*a*c)*f)*x^7 + 1/2*a*
b*e*x^4 + 1/5*(2*a*b*f + (b^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + a^2*d*x + 1
/3*(2*a*b*d + a^2*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 151, normalized size of antiderivative = 0.98

$$\int (d+ex+fx^2)(a+bx^2+cx^4)^2\,dx = \frac{1}{11}c^2fx^{11} + \frac{1}{10}c^2ex^{10} + \frac{1}{9}c^2dx^9 + \frac{2}{9}bcfx^9 + \frac{1}{4}bcex^8 + \frac{2}{7}bcdx^7 + \frac{1}{7}b^2fx^7 + \frac{2}{7}acfx^7 + \frac{1}{6}b^2ex^6 + \frac{1}{3}acex^6 + \frac{1}{5}b^2dx^5 + \frac{2}{5}acdx^5 + \frac{2}{5}abfx^5 + \frac{1}{2}abex^4 + \frac{2}{3}abdx^3 + \frac{1}{3}a^2fx^3 + \frac{1}{2}a^2ex^2 + a^2dx$$

```
[In] integrate((f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/9*c^2*d*x^9 + 2/9*b*c*f*x^9 + 1/4*b*c
*e*x^8 + 2/7*b*c*d*x^7 + 1/7*b^2*f*x^7 + 2/7*a*c*f*x^7 + 1/6*b^2*e*x^6 + 1/
3*a*c*e*x^6 + 1/5*b^2*d*x^5 + 2/5*a*c*d*x^5 + 2/5*a*b*f*x^5 + 1/2*a*b*e*x^4
 + 2/3*a*b*d*x^3 + 1/3*a^2*f*x^3 + 1/2*a^2*e*x^2 + a^2*d*x
```

Mupad [B] (verification not implemented)

Time = 7.85 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\int (d+ex+fx^2)(a+bx^2+cx^4)^2\,dx = x^5\left(\frac{d\,b^2}{5} + \frac{2\,a\,f\,b}{5} + \frac{2\,a\,c\,d}{5}\right) + x^7\left(\frac{f\,b^2}{7} + \frac{2\,c\,d\,b}{7} + \frac{2\,a\,c\,f}{7}\right) + x^3\left(\frac{f\,a^2}{3} + \frac{2\,b\,d\,a}{3}\right) + x^9\left(\frac{d\,c^2}{9} + \frac{2\,b\,f\,c}{9}\right) + \frac{a^2\,e\,x^2}{2} + \frac{c^2\,e\,x^{10}}{10} + \frac{c^2\,f\,x^{11}}{11} + \frac{e\,x^6\,(b^2+2\,a\,c)}{6} + a^2\,d\,x + \frac{a\,b\,e\,x^4}{2} + \frac{b\,c\,e\,x^8}{4}$$

```
[In] int((d + e*x + f*x^2)*(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] x^5*((b^2*d)/5 + (2*a*c*d)/5 + (2*a*b*f)/5) + x^7*((b^2*f)/7 + (2*b*c*d)/7
+ (2*a*c*f)/7) + x^3*((a^2*f)/3 + (2*a*b*d)/3) + x^9*((c^2*d)/9 + (2*b*c*f)
/9) + (a^2*e*x^2)/2 + (c^2*e*x^10)/10 + (c^2*f*x^11)/11 + (e*x^6*(2*a*c + b
^2))/6 + a^2*d*x + (a*b*e*x^4)/2 + (b*c*e*x^8)/4
```

3.8 $\int (d+ex+fx^2+gx^3) (a+bx^2+cx^4)^2 \, dx$

Optimal result

Integrand size = 30, antiderivative size = 196

$$\int (d+ex+fx^2+gx^3) \left(a+bx^2+cx^4\right)^2 \, dx = a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3 +\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5 +\frac{1}{6}\left(b^2e+2ace+2abg\right)x^6 +\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7 +\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8+\frac{1}{9}c(cd+2bf)x^9 +\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c^2fx^{11}+\frac{1}{12}c^2gx^{12}$$

```
[Out] a^2*d*x+1/2*a^2*e*x^2+1/3*a*(a*f+2*b*d)*x^3+1/4*a*(a*g+2*b*e)*x^4+1/5*(2*a*
b*f+2*a*c*d+b^2*d)*x^5+1/6*(2*a*b*g+2*a*c*e+b^2*e)*x^6+1/7*(2*a*c*f+b^2*f+2
*b*c*d)*x^7+1/8*(2*a*c*g+b^2*g+2*b*c*e)*x^8+1/9*c*(2*b*f+c*d)*x^9+1/10*c*(2
*b*g+c*e)*x^10+1/11*c^2*f*x^11+1/12*c^2*g*x^12
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 196, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.033$, Rules used

$= \{1685\}$

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{7}x^7\left(2acf+b^2f+2bcd\right)+\frac{1}{5}x^5\left(2abf+2acd+b^2d\right)+\frac{1}{8}x^8\left(2acg+b^2g+2bce\right)+\frac{1}{6}x^6\left(2abg+2ace+b^2e\right)+\frac{1}{3}ax^3(af+2bd)+\frac{1}{4}ax^4(ag+2be)+\frac{1}{9}cx^9(2bf+cd)+\frac{1}{10}cx^{10}(2bg+ce)+\frac{1}{11}c^2fx^{11}+\frac{1}{12}c^2gx^{12}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*(2*b*e + a*g)*x^4)/4
 + ((b^2*d + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2*e + 2*a*c*e + 2*a*b*g)*x^6)/
6 + ((2*b*c*d + b^2*f + 2*a*c*f)*x^7)/7 + ((2*b*c*e + b^2*g + 2*a*c*g)*x^8)
/8 + (c*(c*d + 2*b*f)*x^9)/9 + (c*(c*e + 2*b*g)*x^10)/10 + (c^2*f*x^11)/11
+ (c^2*g*x^12)/12
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(a^2d+a^2ex+a(2bd+af)x^2+a(2be+ag)x^3+\left(b^2d+2acd+2abf\right)x^4+\left(b^2e+2ace+2abg\right)x^5+\left(2bcd+b^2f+2acf\right)x^6+\left(2bce+b^2g+2acg\right)x^7+c(cd+2bf)x^8+c(ce+2bg)x^9+c^2fx^{10}+c^2gx^{11}\right)dx\\ &= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2e+2ace+2abg\right)x^6+\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c^2fx^{11}+\frac{1}{12}c^2gx^{12}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 196, normalized size of antiderivative = 1.00

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^2\,dx = a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3 +\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5 +\frac{1}{6}\left(b^2e+2ace+2abg\right)x^6 +\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7 +\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8+\frac{1}{9}c(cd+2bf)x^9 +\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c^2fx^{11}+\frac{1}{12}c^2gx^{12}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*(2*b*e + a*g)*x^4)/4
 + ((b^2*d + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2*e + 2*a*c*e + 2*a*b*g)*x^6)/
6 + ((2*b*c*d + b^2*f + 2*a*c*f)*x^7)/7 + ((2*b*c*e + b^2*g + 2*a*c*g)*x^8)
/8 + (c*(c*d + 2*b*f)*x^9)/9 + (c*(c*e + 2*b*g)*x^10)/10 + (c^2*f*x^11)/11
+ (c^2*g*x^12)/12
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 183, normalized size of antiderivative = 0.93

method	result
default	$\frac{c^2 g\,x^{12}}{12}+\frac{c^2 f\,x^{11}}{11}+\frac{(2gbc+e\,c^2)x^{10}}{10}+\frac{(2fbc+c^2d)x^9}{9}+\frac{(2ebc+g(2ac+b^2))x^8}{8}+\frac{(2bcd+f(2ac+b^2))x^7}{7}+$ (e(2ac+b
norman	$\frac{c^2 g\,x^{12}}{12}+\frac{c^2 f\,x^{11}}{11}+\left(\frac{1}{5}gbc+\frac{1}{10}e\,c^2\right)x^{10}+\left(\frac{2}{9}fbc+\frac{1}{9}c^2d\right)x^9+\left(\frac{1}{4}acg+\frac{1}{8}b^2g+\frac{1}{4}ebc\right)x^8+$ (2/7 ac
gosper	$\frac{1}{12}c^2g\,x^{12}+\frac{1}{11}c^2f\,x^{11}+\frac{1}{5}x^{10}gbc+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}x^8acg+\frac{1}{8}x^8b^2g+\frac{1}{4}bce$
risch	$\frac{1}{12}c^2g\,x^{12}+\frac{1}{11}c^2f\,x^{11}+\frac{1}{5}x^{10}gbc+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}x^8acg+\frac{1}{8}x^8b^2g+\frac{1}{4}bce$
parallelrisch	$\frac{1}{12}c^2g\,x^{12}+\frac{1}{11}c^2f\,x^{11}+\frac{1}{5}x^{10}gbc+\frac{1}{10}c^2e\,x^{10}+\frac{2}{9}x^9fbc+\frac{1}{9}c^2d\,x^9+\frac{1}{4}x^8acg+\frac{1}{8}x^8b^2g+\frac{1}{4}bce$

```
[In] int((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] 1/12*c^2*g*x^12+1/11*c^2*f*x^11+1/10*(2*b*c*g+c^2*e)*x^10+1/9*(2*b*c*f+c^2*
d)*x^9+1/8*(2*e*b*c+g*(2*a*c+b^2))*x^8+1/7*(2*b*c*d+f*(2*a*c+b^2))*x^7+1/6*
(e*(2*a*c+b^2)+2*a*b*g)*x^6+1/5*(d*(2*a*c+b^2)+2*a*b*f)*x^5+1/4*(a^2*g+2*a*
b*e)*x^4+1/3*(a^2*f+2*a*b*d)*x^3+1/2*a^2*e*x^2+a^2*d*x
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 182, normalized size of antiderivative = 0.93

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)^2\,dx$$
$$=\frac{1}{12}c^2gx^{12}+\frac{1}{11}c^2fx^{11}+\frac{1}{10}\left(c^2e+2\,bcg\right)x^{10}+\frac{1}{9}\left(c^2d+2\,bcf\right)x^9$$
$$+\frac{1}{8}\left(2\,bce+\left(b^2+2\,ac\right)g\right)x^8+\frac{1}{7}\left(2\,bcd+\left(b^2+2\,ac\right)f\right)x^7+\frac{1}{6}\left(2\,abg+\left(b^2+2\,ac\right)e\right)x^6$$
$$+\frac{1}{5}\left(2\,abf+\left(b^2+2\,ac\right)d\right)x^5+\frac{1}{2}a^2ex^2+\frac{1}{4}\left(2\,abe+a^2g\right)x^4+a^2dx+\frac{1}{3}\left(2\,abd+a^2f\right)x^3$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] 1/12*c^2*g*x^12 + 1/11*c^2*f*x^11 + 1/10*(c^2*e + 2*b*c*g)*x^10 + 1/9*(c^2*
d + 2*b*c*f)*x^9 + 1/8*(2*b*c*e + (b^2 + 2*a*c)*g)*x^8 + 1/7*(2*b*c*d + (b^
2 + 2*a*c)*f)*x^7 + 1/6*(2*a*b*g + (b^2 + 2*a*c)*e)*x^6 + 1/5*(2*a*b*f + (b
^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + 1/4*(2*a*b*e + a^2*g)*x^4 + a^2*d*x +
1/3*(2*a*b*d + a^2*f)*x^3
```

Sympy [A] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 209, normalized size of antiderivative = 1.07

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)^2\,dx=a^2dx+\frac{a^2ex^2}{2}+\frac{c^2fx^{11}}{11}+\frac{c^2gx^{12}}{12}$$
$$+x^{10}\left(\frac{bcg}{5}+\frac{c^2e}{10}\right)+x^9\cdot\left(\frac{2bcf}{9}+\frac{c^2d}{9}\right)$$
$$+x^8\left(\frac{acg}{4}+\frac{b^2g}{8}+\frac{bce}{4}\right)$$
$$+x^7\cdot\left(\frac{2acf}{7}+\frac{b^2f}{7}+\frac{2bcd}{7}\right)$$
$$+x^6\left(\frac{abg}{3}+\frac{ace}{3}+\frac{b^2e}{6}\right)$$
$$+x^5\cdot\left(\frac{2abf}{5}+\frac{2acd}{5}+\frac{b^2d}{5}\right)$$
$$+x^4\left(\frac{a^2g}{4}+\frac{abe}{2}\right)+x^3\left(\frac{a^2f}{3}+\frac{2abd}{3}\right)$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)*(c*x**4+b*x**2+a)**2,x)
```

```
[Out] a**2*d*x + a**2*e*x**2/2 + c**2*f*x**11/11 + c**2*g*x**12/12 + x**10*(b*c*g
/5 + c**2*e/10) + x**9*(2*b*c*f/9 + c**2*d/9) + x**8*(a*c*g/4 + b**2*g/8 +
```

```
b*c*e/4) + x**7*(2*a*c*f/7 + b**2*f/7 + 2*b*c*d/7) + x**6*(a*b*g/3 + a*c*e/
3 + b**2*e/6) + x**5*(2*a*b*f/5 + 2*a*c*d/5 + b**2*d/5) + x**4*(a**2*g/4 +
a*b*e/2) + x**3*(a**2*f/3 + 2*a*b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 182, normalized size of antiderivative = 0.93

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^2\,dx = \frac{1}{12}\,c^2gx^{12}+\frac{1}{11}\,c^2fx^{11}+\frac{1}{10}\,(c^2e+2\,bcg)x^{10}+\frac{1}{9}\,(c^2d+2\,bcf)x^9 + \frac{1}{8}\,(2\,bce+(b^2+2\,ac)g)x^8+\frac{1}{7}\,(2\,bcd+(b^2+2\,ac)f)x^7+\frac{1}{6}\,(2\,abg+(b^2+2\,ac)e)x^6 + \frac{1}{5}\,(2\,abf+(b^2+2\,ac)d)x^5+\frac{1}{2}\,a^2ex^2+\frac{1}{4}\,(2\,abe+a^2g)x^4+a^2dx+\frac{1}{3}\,(2\,abd+a^2f)x^3$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] 1/12*c^2*g*x^12 + 1/11*c^2*f*x^11 + 1/10*(c^2*e + 2*b*c*g)*x^10 + 1/9*(c^2*
d + 2*b*c*f)*x^9 + 1/8*(2*b*c*e + (b^2 + 2*a*c)*g)*x^8 + 1/7*(2*b*c*d + (b^
2 + 2*a*c)*f)*x^7 + 1/6*(2*a*b*g + (b^2 + 2*a*c)*e)*x^6 + 1/5*(2*a*b*f + (b
^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + 1/4*(2*a*b*e + a^2*g)*x^4 + a^2*d*x +
1/3*(2*a*b*d + a^2*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 202, normalized size of antiderivative = 1.03

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^2\,dx = \frac{1}{12}\,c^2gx^{12}+\frac{1}{11}\,c^2fx^{11}+\frac{1}{10}\,c^2ex^{10}+\frac{1}{5}\,bcgx^{10} + \frac{1}{9}\,c^2dx^9+\frac{2}{9}\,bcfx^9+\frac{1}{4}\,bcex^8+\frac{1}{8}\,b^2gx^8 + \frac{1}{4}\,acgx^8+\frac{2}{7}\,bcdx^7+\frac{1}{7}\,b^2fx^7+\frac{2}{7}\,acfx^7 + \frac{1}{6}\,b^2ex^6+\frac{1}{3}\,acex^6+\frac{1}{3}\,abgx^6+\frac{1}{5}\,b^2dx^5 + \frac{2}{5}\,acdx^5+\frac{2}{5}\,abfx^5+\frac{1}{2}\,abex^4+\frac{1}{4}\,a^2gx^4 + \frac{2}{3}\,abdx^3+\frac{1}{3}\,a^2fx^3+\frac{1}{2}\,a^2ex^2+a^2dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/12*c^2*g*x^12 + 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/5*b*c*g*x^10 + 1/9*
c^2*d*x^9 + 2/9*b*c*f*x^9 + 1/4*b*c*e*x^8 + 1/8*b^2*g*x^8 + 1/4*a*c*g*x^8 +
 2/7*b*c*d*x^7 + 1/7*b^2*f*x^7 + 2/7*a*c*f*x^7 + 1/6*b^2*e*x^6 + 1/3*a*c*e*
x^6 + 1/3*a*b*g*x^6 + 1/5*b^2*d*x^5 + 2/5*a*c*d*x^5 + 2/5*a*b*f*x^5 + 1/2*a
*b*e*x^4 + 1/4*a^2*g*x^4 + 2/3*a*b*d*x^3 + 1/3*a^2*f*x^3 + 1/2*a^2*e*x^2 +
a^2*d*x
```

Mupad [B] (verification not implemented)

Time = 7.88 (sec) , antiderivative size = 182, normalized size of antiderivative = 0.93

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)^2\,dx$$
$$=x^5\left(\frac{d\,b^2}{5}+\frac{2\,a\,f\,b}{5}+\frac{2\,a\,c\,d}{5}\right)+x^6\left(\frac{e\,b^2}{6}+\frac{a\,g\,b}{3}+\frac{a\,c\,e}{3}\right)+x^7\left(\frac{f\,b^2}{7}+\frac{2\,c\,d\,b}{7}+\frac{2\,a\,c\,f}{7}\right)$$
$$+x^8\left(\frac{g\,b^2}{8}+\frac{c\,e\,b}{4}+\frac{a\,c\,g}{4}\right)+x^3\left(\frac{f\,a^2}{3}+\frac{2\,b\,d\,a}{3}\right)+x^4\left(\frac{g\,a^2}{4}+\frac{b\,e\,a}{2}\right)$$
$$+x^9\left(\frac{d\,c^2}{9}+\frac{2\,b\,f\,c}{9}\right)+x^{10}\left(\frac{e\,c^2}{10}+\frac{b\,g\,c}{5}\right)+\frac{a^2\,e\,x^2}{2}+\frac{c^2\,f\,x^{11}}{11}+\frac{c^2\,g\,x^{12}}{12}+a^2\,d\,x$$

```
[In] int((a + b*x^2 + c*x^4)^2*(d + e*x + f*x^2 + g*x^3),x)
```

```
[Out] x^5*((b^2*d)/5 + (2*a*c*d)/5 + (2*a*b*f)/5) + x^6*((b^2*e)/6 + (a*c*e)/3 +
(a*b*g)/3) + x^7*((b^2*f)/7 + (2*b*c*d)/7 + (2*a*c*f)/7) + x^8*((b^2*g)/8 +
 (b*c*e)/4 + (a*c*g)/4) + x^3*((a^2*f)/3 + (2*a*b*d)/3) + x^4*((a^2*g)/4 +
(a*b*e)/2) + x^9*((c^2*d)/9 + (2*b*c*f)/9) + x^10*((c^2*e)/10 + (b*c*g)/5)
+ (a^2*e*x^2)/2 + (c^2*f*x^11)/11 + (c^2*g*x^12)/12 + a^2*d*x
```

3.9 $\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right) dx$

Optimal result

Integrand size = 35, antiderivative size = 234

$$
\begin{aligned}
&\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx \\
&= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2abf+a(2cd+ah)\right)x^5 \\
&\quad+\frac{1}{6}\left(b^2e+2ace+2abg\right)x^6+\frac{1}{7}\left(b^2f+2acf+2b(cd+ah)\right)x^7 \\
&\quad+\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8+\frac{1}{9}\left(c^2d+b^2h+2c(bf+ah)\right)x^9 \\
&\quad+\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c(cf+2bh)x^{11}+\frac{1}{12}c^2gx^{12}+\frac{1}{13}c^2hx^{13}
\end{aligned}
$$

```
[Out] a^2*d*x+1/2*a^2*e*x^2+1/3*a*(a*f+2*b*d)*x^3+1/4*a*(a*g+2*b*e)*x^4+1/5*(b^2*
d+2*a*b*f+a*(a*h+2*c*d))*x^5+1/6*(2*a*b*g+2*a*c*e+b^2*e)*x^6+1/7*(b^2*f+2*a
*c*f+2*b*(a*h+c*d))*x^7+1/8*(2*a*c*g+b^2*g+2*b*c*e)*x^8+1/9*(c^2*d+b^2*h+2*
c*(a*h+b*f))*x^9+1/10*c*(2*b*g+c*e)*x^10+1/11*c*(2*b*h+c*f)*x^11+1/12*c^2*g
*x^12+1/13*c^2*h*x^13
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.029$, Rules used

$= \{1685\}$

$$\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx$$
$$= a^2dx + \frac{1}{2}a^2ex^2 + \frac{1}{9}x^9\left(2c(ah+bf)+b^2h+c^2d\right) + \frac{1}{7}x^7\left(2b(ah+cd)+2acf+b^2f\right)$$
$$+ \frac{1}{5}x^5\left(2abf+a(ah+2cd)+b^2d\right) + \frac{1}{8}x^8\left(2acg+b^2g+2bce\right)$$
$$+ \frac{1}{6}x^6\left(2abg+2ace+b^2e\right) + \frac{1}{3}ax^3(af+2bd) + \frac{1}{4}ax^4(ag+2be)$$
$$+ \frac{1}{10}cx^{10}(2bg+ce) + \frac{1}{11}cx^{11}(2bh+cf) + \frac{1}{12}c^2gx^{12} + \frac{1}{13}c^2hx^{13}$$

```
[In] Int[(a + b*x^2 + c*x^4)^2*(d + e*x + f*x^2 + g*x^3 + h*x^4),x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*(2*b*e + a*g)*x^4)/4
 + ((b^2*d + 2*a*b*f + a*(2*c*d + a*h))*x^5)/5 + ((b^2*e + 2*a*c*e + 2*a*b*
g)*x^6)/6 + ((b^2*f + 2*a*c*f + 2*b*(c*d + a*h))*x^7)/7 + ((2*b*c*e + b^2*g
 + 2*a*c*g)*x^8)/8 + ((c^2*d + b^2*h + 2*c*(b*f + a*h))*x^9)/9 + (c*(c*e +
2*b*g)*x^10)/10 + (c*(c*f + 2*b*h)*x^11)/11 + (c^2*g*x^12)/12 + (c^2*h*x^13
)/13
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\text{integral} = \int \left(a^2d+a^2ex+a(2bd+af)x^2+a(2be+ag)x^3+\left(b^2d+2abf+a(2cd+ah)\right)x^4 + \left(b^2e+2ace+2abg\right)x^5+\left(b^2f+2acf+2b(cd+ah)\right)x^6+\left(2bce+b^2g+2acg\right)x^7 + \left(c^2d+b^2h+2c(bf+ah)\right)x^8+c(ce+2bg)x^9+c(cf+2bh)x^{10}+c^2gx^{11}+c^2hx^{12}\right)dx$$
$$= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2abf+a(2cd+ah)\right)x^5 + \frac{1}{6}\left(b^2e+2ace+2abg\right)x^6+\frac{1}{7}\left(b^2f+2acf+2b(cd+ah)\right)x^7+\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8 + \frac{1}{9}\left(c^2d+b^2h+2c(bf+ah)\right)x^9+\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c(cf+2bh)x^{11}+\frac{1}{12}c^2gx^{12} + \frac{1}{13}c^2hx^{13}$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.00

$$\begin{aligned} &\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right) \, dx \\ &= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{4}a(2be+ag)x^4+\frac{1}{5}\left(b^2d+2acd+2abf+a^2h\right)x^5 \\ &\quad +\frac{1}{6}\left(b^2e+2ace+2abg\right)x^6+\frac{1}{7}\left(2bcd+b^2f+2acf+2abh\right)x^7 \\ &\quad +\frac{1}{8}\left(2bce+b^2g+2acg\right)x^8+\frac{1}{9}\left(c^2d+2bcf+b^2h+2ach\right)x^9 \\ &\quad +\frac{1}{10}c(ce+2bg)x^{10}+\frac{1}{11}c(cf+2bh)x^{11}+\frac{1}{12}c^2gx^{12}+\frac{1}{13}c^2hx^{13} \end{aligned}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)^2*(d + e*x + f*x^2 + g*x^3 + h*x^4),x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*(2*b*e + a*g)*x^4)/4
 + ((b^2*d + 2*a*c*d + 2*a*b*f + a^2*h)*x^5)/5 + ((b^2*e + 2*a*c*e + 2*a*b*
g)*x^6)/6 + ((2*b*c*d + b^2*f + 2*a*c*f + 2*a*b*h)*x^7)/7 + ((2*b*c*e + b^2
*g + 2*a*c*g)*x^8)/8 + ((c^2*d + 2*b*c*f + b^2*h + 2*a*c*h)*x^9)/9 + (c*(c*
e + 2*b*g)*x^10)/10 + (c*(c*f + 2*b*h)*x^11)/11 + (c^2*g*x^12)/12 + (c^2*h*
x^13)/13
```

Maple [A] (verified)

Time = 0.23 (sec) , antiderivative size = 219, normalized size of antiderivative = 0.94

method	result
default	$\frac{c^2 h\,x^{13}}{13}+\frac{c^2 g\,x^{12}}{12}+\frac{\left(2bch+f\,c^2\right)x^{11}}{11}+\frac{\left(2gbc+e\,c^2\right)x^{10}}{10}+\frac{\left(\left(2ac+b^2\right)h+2fbc+c^2 d\right)x^9}{9}+\frac{\left(2ebc+g\left(2ac+b^2\right)\right)x^8}{8}+$
norman	$\frac{c^2 h\,x^{13}}{13}+\frac{c^2 g\,x^{12}}{12}+\left(\frac{2}{11}bch+\frac{1}{11}f\,c^2\right)x^{11}+\left(\frac{1}{5}gbc+\frac{1}{10}e\,c^2\right)x^{10}+\left(\frac{2}{9}ach+\frac{1}{9}b^2h+\frac{2}{9}fbc+\frac{1}{9}c^2d\right.$
gosper	$\frac{1}{9}x^9b^2h+\frac{1}{5}x^5a^2h+\frac{1}{8}x^8b^2g+\frac{1}{4}x^4g\,a^2+\frac{2}{9}x^9fbc+\frac{2}{7}x^7acf+\frac{2}{5}x^5abf+\frac{1}{3}x^3f\,a^2+\frac{1}{7}x^7b^2f+$
risch	$\frac{1}{9}x^9b^2h+\frac{1}{5}x^5a^2h+\frac{1}{8}x^8b^2g+\frac{1}{4}x^4g\,a^2+\frac{2}{9}x^9fbc+\frac{2}{7}x^7acf+\frac{2}{5}x^5abf+\frac{1}{3}x^3f\,a^2+\frac{1}{7}x^7b^2f+$
parallelrisch	$\frac{1}{9}x^9b^2h+\frac{1}{5}x^5a^2h+\frac{1}{8}x^8b^2g+\frac{1}{4}x^4g\,a^2+\frac{2}{9}x^9fbc+\frac{2}{7}x^7acf+\frac{2}{5}x^5abf+\frac{1}{3}x^3f\,a^2+\frac{1}{7}x^7b^2f+$

```
[In] int((c*x^4+b*x^2+a)^2*(h*x^4+g*x^3+f*x^2+e*x+d),x,method=_RETURNVERBOSE)
```

```
[Out] 1/13*c^2*h*x^13+1/12*c^2*g*x^12+1/11*(2*b*c*h+c^2*f)*x^11+1/10*(2*b*c*g+c^2
*e)*x^10+1/9*((2*a*c+b^2)*h+2*f*b*c+c^2*d)*x^9+1/8*(2*e*b*c+g*(2*a*c+b^2))*
x^8+1/7*(2*a*b*h+f*(2*a*c+b^2)+2*b*c*d)*x^7+1/6*(e*(2*a*c+b^2)+2*a*b*g)*x^6
+1/5*(a^2*h+2*a*b*f+d*(2*a*c+b^2))*x^5+1/4*(a^2*g+2*a*b*e)*x^4+1/3*(a^2*f+2
*a*b*d)*x^3+1/2*a^2*e*x^2+a^2*d*x
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 218, normalized size of antiderivative = 0.93

$$\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx = \frac{1}{13}\,c^2hx^{13}+\frac{1}{12}\,c^2gx^{12}+\frac{1}{11}\,(c^2f+2\,bch)x^{11}+\frac{1}{10}\,(c^2e+2\,bcg)x^{10} + \frac{1}{9}\,\left(c^2d+2\,bcf+\left(b^2+2\,ac\right)h\right)x^9+\frac{1}{8}\,\left(2\,bce+\left(b^2+2\,ac\right)g\right)x^8 + \frac{1}{7}\,\left(2\,bcd+2\,abh+\left(b^2+2\,ac\right)f\right)x^7+\frac{1}{6}\,\left(2\,abg+\left(b^2+2\,ac\right)e\right)x^6 + \frac{1}{5}\,\left(2\,abf+a^2h+\left(b^2+2\,ac\right)d\right)x^5+\frac{1}{2}\,a^2ex^2 + \frac{1}{4}\,(2\,abe+a^2g)x^4+a^2dx+\frac{1}{3}\,(2\,abd+a^2f)x^3$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="fricas"
)
```

```
[Out] 1/13*c^2*h*x^13 + 1/12*c^2*g*x^12 + 1/11*(c^2*f + 2*b*c*h)*x^11 + 1/10*(c^2
*e + 2*b*c*g)*x^10 + 1/9*(c^2*d + 2*b*c*f + (b^2 + 2*a*c)*h)*x^9 + 1/8*(2*b
*c*e + (b^2 + 2*a*c)*g)*x^8 + 1/7*(2*b*c*d + 2*a*b*h + (b^2 + 2*a*c)*f)*x^7
 + 1/6*(2*a*b*g + (b^2 + 2*a*c)*e)*x^6 + 1/5*(2*a*b*f + a^2*h + (b^2 + 2*a*
c)*d)*x^5 + 1/2*a^2*e*x^2 + 1/4*(2*a*b*e + a^2*g)*x^4 + a^2*d*x + 1/3*(2*a*
b*d + a^2*f)*x^3
```

Sympy [A] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 258, normalized size of antiderivative = 1.10

$$\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx = a^2dx+\frac{a^2ex^2}{2}+\frac{c^2gx^{12}}{12}+\frac{c^2hx^{13}}{13}+x^{11}\cdot\left(\frac{2bch}{11}+\frac{c^2f}{11}\right)+x^{10}\left(\frac{bcg}{5}+\frac{c^2e}{10}\right) + x^9\cdot\left(\frac{2ach}{9}+\frac{b^2h}{9}+\frac{2bcf}{9}+\frac{c^2d}{9}\right)+x^8\left(\frac{acg}{4}+\frac{b^2g}{8}+\frac{bce}{4}\right) + x^7\cdot\left(\frac{2abh}{7}+\frac{2acf}{7}+\frac{b^2f}{7}+\frac{2bcd}{7}\right)+x^6\left(\frac{abg}{3}+\frac{ace}{3}+\frac{b^2e}{6}\right) + x^5\left(\frac{a^2h}{5}+\frac{2abf}{5}+\frac{2acd}{5}+\frac{b^2d}{5}\right)+x^4\left(\frac{a^2g}{4}+\frac{abe}{2}\right)+x^3\left(\frac{a^2f}{3}+\frac{2abd}{3}\right)$$

```
[In] integrate((c*x**4+b*x**2+a)**2*(h*x**4+g*x**3+f*x**2+e*x+d),x)
```

```
[Out] a**2*d*x + a**2*e*x**2/2 + c**2*g*x**12/12 + c**2*h*x**13/13 + x**11*(2*b*c
*h/11 + c**2*f/11) + x**10*(b*c*g/5 + c**2*e/10) + x**9*(2*a*c*h/9 + b**2*h
/9 + 2*b*c*f/9 + c**2*d/9) + x**8*(a*c*g/4 + b**2*g/8 + b*c*e/4) + x**7*(2*
a*b*h/7 + 2*a*c*f/7 + b**2*f/7 + 2*b*c*d/7) + x**6*(a*b*g/3 + a*c*e/3 + b**
2*e/6) + x**5*(a**2*h/5 + 2*a*b*f/5 + 2*a*c*d/5 + b**2*d/5) + x**4*(a**2*g/
4 + a*b*e/2) + x**3*(a**2*f/3 + 2*a*b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 218, normalized size of antiderivative = 0.93

$$\begin{aligned}
&\int \left(a + bx^2 + cx^4\right)^2 \left(d + ex + fx^2 + gx^3 + hx^4\right) dx \\
&= \frac{1}{13} c^2 h x^{13} + \frac{1}{12} c^2 g x^{12} + \frac{1}{11} \left(c^2 f + 2\,bch\right) x^{11} + \frac{1}{10} \left(c^2 e + 2\,bcg\right) x^{10} \\
&\quad + \frac{1}{9} \left(c^2 d + 2\,bcf + \left(b^2 + 2\,ac\right) h\right) x^9 + \frac{1}{8} \left(2\,bce + \left(b^2 + 2\,ac\right) g\right) x^8 \\
&\quad + \frac{1}{7} \left(2\,bcd + 2\,abh + \left(b^2 + 2\,ac\right) f\right) x^7 + \frac{1}{6} \left(2\,abg + \left(b^2 + 2\,ac\right) e\right) x^6 \\
&\quad + \frac{1}{5} \left(2\,abf + a^2 h + \left(b^2 + 2\,ac\right) d\right) x^5 + \frac{1}{2} a^2 e x^2 \\
&\quad + \frac{1}{4} \left(2\,abe + a^2 g\right) x^4 + a^2 d x + \frac{1}{3} \left(2\,abd + a^2 f\right) x^3
\end{aligned}$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="maxima"
)
```

```
[Out] 1/13*c^2*h*x^13 + 1/12*c^2*g*x^12 + 1/11*(c^2*f + 2*b*c*h)*x^11 + 1/10*(c^2
*e + 2*b*c*g)*x^10 + 1/9*(c^2*d + 2*b*c*f + (b^2 + 2*a*c)*h)*x^9 + 1/8*(2*b
*c*e + (b^2 + 2*a*c)*g)*x^8 + 1/7*(2*b*c*d + 2*a*b*h + (b^2 + 2*a*c)*f)*x^7
 + 1/6*(2*a*b*g + (b^2 + 2*a*c)*e)*x^6 + 1/5*(2*a*b*f + a^2*h + (b^2 + 2*a*
c)*d)*x^5 + 1/2*a^2*e*x^2 + 1/4*(2*a*b*e + a^2*g)*x^4 + a^2*d*x + 1/3*(2*a*
b*d + a^2*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 253, normalized size of antiderivative = 1.08

$$\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx$$
$$= \frac{1}{13}c^2hx^{13}+\frac{1}{12}c^2gx^{12}+\frac{1}{11}c^2fx^{11}+\frac{2}{11}bchx^{11}+\frac{1}{10}c^2ex^{10}+\frac{1}{5}bcgx^{10}+\frac{1}{9}c^2dx^9 +\frac{2}{9}bcfx^9+\frac{1}{9}b^2hx^9+\frac{2}{9}achx^9+\frac{1}{4}bcex^8+\frac{1}{8}b^2gx^8+\frac{1}{4}acgx^8+\frac{2}{7}bcdx^7+\frac{1}{7}b^2fx^7 +\frac{2}{7}acfx^7+\frac{2}{7}abhx^7+\frac{1}{6}b^2ex^6+\frac{1}{3}acex^6+\frac{1}{3}abgx^6+\frac{1}{5}b^2dx^5+\frac{2}{5}acdx^5 +\frac{2}{5}abfx^5+\frac{1}{5}a^2hx^5+\frac{1}{2}abex^4+\frac{1}{4}a^2gx^4+\frac{2}{3}abdx^3+\frac{1}{3}a^2fx^3+\frac{1}{2}a^2ex^2+a^2dx$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(h*x^4+g*x^3+f*x^2+e*x+d),x, algorithm="giac")
```

```
[Out] 1/13*c^2*h*x^13 + 1/12*c^2*g*x^12 + 1/11*c^2*f*x^11 + 2/11*b*c*h*x^11 + 1/1
0*c^2*e*x^10 + 1/5*b*c*g*x^10 + 1/9*c^2*d*x^9 + 2/9*b*c*f*x^9 + 1/9*b^2*h*x
^9 + 2/9*a*c*h*x^9 + 1/4*b*c*e*x^8 + 1/8*b^2*g*x^8 + 1/4*a*c*g*x^8 + 2/7*b*
c*d*x^7 + 1/7*b^2*f*x^7 + 2/7*a*c*f*x^7 + 2/7*a*b*h*x^7 + 1/6*b^2*e*x^6 + 1
/3*a*c*e*x^6 + 1/3*a*b*g*x^6 + 1/5*b^2*d*x^5 + 2/5*a*c*d*x^5 + 2/5*a*b*f*x^
5 + 1/5*a^2*h*x^5 + 1/2*a*b*e*x^4 + 1/4*a^2*g*x^4 + 2/3*a*b*d*x^3 + 1/3*a^2
*f*x^3 + 1/2*a^2*e*x^2 + a^2*d*x
```

Mupad [B] (verification not implemented)

Time = 0.08 (sec) , antiderivative size = 220, normalized size of antiderivative = 0.94

$$\int \left(a+bx^2+cx^4\right)^2 \left(d+ex+fx^2+gx^3+hx^4\right)\,dx$$
$$= x^6\left(\frac{e\,b^2}{6}+\frac{a\,g\,b}{3}+\frac{a\,c\,e}{3}\right)+x^8\left(\frac{g\,b^2}{8}+\frac{c\,e\,b}{4}+\frac{a\,c\,g}{4}\right)+x^3\left(\frac{f\,a^2}{3}+\frac{2\,b\,d\,a}{3}\right) +x^4\left(\frac{g\,a^2}{4}+\frac{b\,e\,a}{2}\right)+x^{10}\left(\frac{e\,c^2}{10}+\frac{b\,g\,c}{5}\right)+x^{11}\left(\frac{f\,c^2}{11}+\frac{2\,b\,h\,c}{11}\right) +x^5\left(\frac{h\,a^2}{5}+\frac{2\,f\,a\,b}{5}+\frac{2\,c\,d\,a}{5}+\frac{d\,b^2}{5}\right)+x^7\left(\frac{b^2\,f}{7}+\frac{2\,b\,c\,d}{7}+\frac{2\,a\,c\,f}{7}+\frac{2\,a\,b\,h}{7}\right) +x^9\left(\frac{h\,b^2}{9}+\frac{2\,f\,b\,c}{9}+\frac{d\,c^2}{9}+\frac{2\,a\,h\,c}{9}\right)+\frac{a^2\,e\,x^2}{2}+\frac{c^2\,g\,x^{12}}{12}+\frac{c^2\,h\,x^{13}}{13}+a^2\,d\,x$$

```
[In] int((a + b*x^2 + c*x^4)^2*(d + e*x + f*x^2 + g*x^3 + h*x^4),x)
```

```
[Out] x^6*((b^2*e)/6 + (a*c*e)/3 + (a*b*g)/3) + x^8*((b^2*g)/8 + (b*c*e)/4 + (a*c
*g)/4) + x^3*((a^2*f)/3 + (2*a*b*d)/3) + x^4*((a^2*g)/4 + (a*b*e)/2) + x^10
*((c^2*e)/10 + (b*c*g)/5) + x^11*((c^2*f)/11 + (2*b*c*h)/11) + x^5*((b^2*d)
/5 + (a^2*h)/5 + (2*a*c*d)/5 + (2*a*b*f)/5) + x^7*((b^2*f)/7 + (2*b*c*d)/7
+ (2*a*c*f)/7 + (2*a*b*h)/7) + x^9*((c^2*d)/9 + (b^2*h)/9 + (2*b*c*f)/9 + (
2*a*c*h)/9) + (a^2*e*x^2)/2 + (c^2*g*x^12)/12 + (c^2*h*x^13)/13 + a^2*d*x
```

3.10 $\int \frac{d+ex}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 18, antiderivative size = 45

$$\int \frac{d+ex}{4-5x^2+x^4} \, dx = -\frac{1}{6}d\text{arctanh}\left(\frac{x}{2}\right)+\frac{1}{3}d\text{arctanh}(x)-\frac{1}{6}e\log\left(1-x^2\right)+\frac{1}{6}e\log\left(4-x^2\right)$$

```
[Out] -1/6*d*arctanh(1/2*x)+1/3*d*arctanh(x)-1/6*e*ln(-x^2+1)+1/6*e*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.02 (sec) , antiderivative size = 45, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.389, Rules used = {1687, 12, 1107, 213, 1121, 630, 31}

$$\int \frac{d+ex}{4-5x^2+x^4} \, dx = -\frac{1}{6}d\text{arctanh}\left(\frac{x}{2}\right)+\frac{1}{3}d\text{arctanh}(x)-\frac{1}{6}e\log\left(1-x^2\right)+\frac{1}{6}e\log\left(4-x^2\right)$$

```
[In] Int[(d + e*x)/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/6*(d*ArcTanh[x/2]) + (d*ArcTanh[x])/3 - (e*Log[1 - x^2])/6 + (e*Log[4 - x^2])/6
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] && !MatchQ[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x, x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1107

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(-1), x_Symbol] :> With[{q = Rt[b^
2 - 4*a*c, 2]}, Dist[c/q, Int[1/(b/2 - q/2 + c*x^2), x], x] - Dist[c/q, Int
[1/(b/2 + q/2 + c*x^2), x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c,
0] && PosQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d}{4-5x^2+x^4}\,dx + \int \frac{ex}{4-5x^2+x^4}\,dx \\
&= d\int \frac{1}{4-5x^2+x^4}\,dx + e\int \frac{x}{4-5x^2+x^4}\,dx \\
&= \frac{1}{3}d\int \frac{1}{-4+x^2}\,dx - \frac{1}{3}d\int \frac{1}{-1+x^2}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right) \\
&= -\frac{1}{6}d\tanh^{-1}\left(\frac{x}{2}\right) + \frac{1}{3}d\tanh^{-1}(x) + \frac{1}{6}e\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) \\
&\quad - \frac{1}{6}e\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right)
\end{aligned}$$

$$= -\frac{1}{6}d\tanh^{-1}\left(\frac{x}{2}\right) + \frac{1}{3}d\tanh^{-1}(x) - \frac{1}{6}e\log\left(1-x^2\right) + \frac{1}{6}e\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.11

$$\int \frac{d+ex}{4-5x^2+x^4}\,dx = \frac{1}{12}(-2(d+e)\log(1-x) + (d+2e)\log(2-x) + 2(d-e)\log(1+x) - (d-2e)\log(2+x))$$

```
[In] Integrate[(d + e*x)/(4 - 5*x^2 + x^4),x]
[Out] (-2*(d + e)*Log[1 - x] + (d + 2*e)*Log[2 - x] + 2*(d - e)*Log[1 + x] - (d -
 2*e)*Log[2 + x])/12
```

Maple [A] (verified)

Time = 0.05 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.11

method	result	size
default	$\left(-\frac{d}{12}+\frac{e}{6}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}\right)\ln\left(x-1\right)+\left(\frac{d}{12}+\frac{e}{6}\right)\ln\left(x-2\right)$	50
norman	$\left(-\frac{d}{12}+\frac{e}{6}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}\right)\ln\left(x-1\right)+\left(\frac{d}{12}+\frac{e}{6}\right)\ln\left(x-2\right)$	50
parallelrisch	$\frac{\ln(x-2)d}{12}+\frac{\ln(x-2)e}{6}-\frac{\ln(x-1)d}{6}-\frac{\ln(x-1)e}{6}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}-\frac{\ln(x+2)d}{12}+\frac{\ln(x+2)e}{6}$	58
risch	$\frac{\ln(2-x)d}{12}+\frac{\ln(2-x)e}{6}-\frac{\ln(x+2)d}{12}+\frac{\ln(x+2)e}{6}-\frac{\ln(1-x)d}{6}-\frac{\ln(1-x)e}{6}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}$	66

```
[In] int((e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] (-1/12*d+1/6*e)*ln(x+2)+(1/6*d-1/6*e)*ln(x+1)+(-1/6*d-1/6*e)*ln(x-1)+(1/12*
d+1/6*e)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 43, normalized size of antiderivative = 0.96

$$\int \frac{d+ex}{4-5x^2+x^4}\,dx = -\frac{1}{12}\,(d-2\,e)\log\left(x+2\right) + \frac{1}{6}\,(d-e)\log\left(x+1\right) - \frac{1}{6}\,(d+e)\log\left(x-1\right) + \frac{1}{12}\,(d+2\,e)\log\left(x-2\right)$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] -1/12*(d - 2*e)*log(x + 2) + 1/6*(d - e)*log(x + 1) - 1/6*(d + e)*log(x - 1
) + 1/12*(d + 2*e)*log(x - 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 515 vs. $2(34) = 68$.

Time = 1.81 (sec) , antiderivative size = 515, normalized size of antiderivative = 11.44

$$\int \frac{d + ex}{4 - 5x^2 + x^4}\,dx = $$

$$-\frac{(d-2e)\log\left(x + \frac{-35d^4e + \frac{51d^4(d-2e)}{2} - 180d^2e^3 - 90d^2e^2(d-2e) + 41d^2e(d-2e)^2 - \frac{15d^2(d-2e)^3}{2} + 320e^5 - 96e^4(d-2e) - 80e^3(d-2e)^2 + 24e}{9d^5 - 160d^3e^2 + 256de^4}\right.}{12}$$

$$+\frac{(d-e)\log\left(x + \frac{-35d^4e - 51d^4(d-e) - 180d^2e^3 + 180d^2e^2(d-e) + 164d^2e(d-e)^2 + 60d^2(d-e)^3 + 320e^5 + 192e^4(d-e) - 320e^3(d-e)^2 - 192e^2(}{9d^5 - 160d^3e^2 + 256de^4}\right.}{6}$$

$$-\frac{(d+e)\log\left(x + \frac{-35d^4e + 51d^4(d+e) - 180d^2e^3 - 180d^2e^2(d+e) + 164d^2e(d+e)^2 - 60d^2(d+e)^3 + 320e^5 - 192e^4(d+e) - 320e^3(d+e)^2 + 192e^2(}{9d^5 - 160d^3e^2 + 256de^4}\right.}{6}$$

$$+\frac{(d+2e)\log\left(x + \frac{-35d^4e - \frac{51d^4(d+2e)}{2} - 180d^2e^3 + 90d^2e^2(d+2e) + 41d^2e(d+2e)^2 + \frac{15d^2(d+2e)^3}{2} + 320e^5 + 96e^4(d+2e) - 80e^3(d+2e)^2 - 24e^2}{9d^5 - 160d^3e^2 + 256de^4}\right.}{12}$$

```
[In] integrate((e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] -(d - 2*e)*log(x + (-35*d**4*e + 51*d**4*(d - 2*e)/2 - 180*d**2*e**3 - 90*d
**2*e**2*(d - 2*e) + 41*d**2*e*(d - 2*e)**2 - 15*d**2*(d - 2*e)**3/2 + 320*
e**5 - 96*e**4*(d - 2*e) - 80*e**3*(d - 2*e)**2 + 24*e**2*(d - 2*e)**3)/(9*
d**5 - 160*d**3*e**2 + 256*d*e**4))/12 + (d - e)*log(x + (-35*d**4*e - 51*d
**4*(d - e) - 180*d**2*e**3 + 180*d**2*e**2*(d - e) + 164*d**2*e*(d - e)**2
 + 60*d**2*(d - e)**3 + 320*e**5 + 192*e**4*(d - e) - 320*e**3*(d - e)**2 -
 192*e**2*(d - e)**3)/(9*d**5 - 160*d**3*e**2 + 256*d*e**4))/6 - (d + e)*lo
g(x + (-35*d**4*e + 51*d**4*(d + e) - 180*d**2*e**3 - 180*d**2*e**2*(d + e)
 + 164*d**2*e*(d + e)**2 - 60*d**2*(d + e)**3 + 320*e**5 - 192*e**4*(d + e)
 - 320*e**3*(d + e)**2 + 192*e**2*(d + e)**3)/(9*d**5 - 160*d**3*e**2 + 256
*d*e**4))/6 + (d + 2*e)*log(x + (-35*d**4*e - 51*d**4*(d + 2*e)/2 - 180*d**
2*e**3 + 90*d**2*e**2*(d + 2*e) + 41*d**2*e*(d + 2*e)**2 + 15*d**2*(d + 2*e
)**3/2 + 320*e**5 + 96*e**4*(d + 2*e) - 80*e**3*(d + 2*e)**2 - 24*e**2*(d +
 2*e)**3)/(9*d**5 - 160*d**3*e**2 + 256*d*e**4))/12
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 43, normalized size of antiderivative = 0.96

$$\int \frac{d+ex}{4-5x^2+x^4}\,dx = -\frac{1}{12}\,(d-2\,e)\log\left(x+2\right)+\frac{1}{6}\,(d-e)\log\left(x+1\right) -\frac{1}{6}\,(d+e)\log\left(x-1\right)+\frac{1}{12}\,(d+2\,e)\log\left(x-2\right)$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] -1/12*(d - 2*e)*log(x + 2) + 1/6*(d - e)*log(x + 1) - 1/6*(d + e)*log(x - 1
) + 1/12*(d + 2*e)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.04

$$\int \frac{d+ex}{4-5x^2+x^4}\,dx = -\frac{1}{12}\,(d-2\,e)\log\left(|x+2|\right)+\frac{1}{6}\,(d-e)\log\left(|x+1|\right) -\frac{1}{6}\,(d+e)\log\left(|x-1|\right)+\frac{1}{12}\,(d+2\,e)\log\left(|x-2|\right)$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] -1/12*(d - 2*e)*log(abs(x + 2)) + 1/6*(d - e)*log(abs(x + 1)) - 1/6*(d + e)
*log(abs(x - 1)) + 1/12*(d + 2*e)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.81 (sec) , antiderivative size = 51, normalized size of antiderivative = 1.13

$$\int \frac{d+ex}{4-5x^2+x^4}\,dx = \ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}\right)-\ln\left(x-1\right)\,\left(\frac{d}{6}+\frac{e}{6}\right) +\ln\left(x-2\right)\,\left(\frac{d}{12}+\frac{e}{6}\right)-\ln\left(x+2\right)\,\left(\frac{d}{12}-\frac{e}{6}\right)$$

```
[In] int((d + e*x)/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x + 1)*(d/6 - e/6) - log(x - 1)*(d/6 + e/6) + log(x - 2)*(d/12 + e/6) -
 log(x + 2)*(d/12 - e/6)
```

3.11 $\int \frac{d+ex+fx^2}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 23, antiderivative size = 51

$$\int \frac{d+ex+fx^2}{4-5x^2+x^4} \, dx = -\frac{1}{6}(d+4f)\text{arctanh}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\text{arctanh}(x) -\frac{1}{6}e\log\left(1-x^2\right)+\frac{1}{6}e\log\left(4-x^2\right)$$

```
[Out] -1/6*(d+4*f)*arctanh(1/2*x)+1/3*(d+f)*arctanh(x)-1/6*e*ln(-x^2+1)+1/6*e*ln(
-x^2+4)
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 51, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.304, Rules used = {1687, 1180, 213, 12, 1121, 630, 31}

$$\int \frac{d+ex+fx^2}{4-5x^2+x^4} \, dx = -\frac{1}{6}\text{arctanh}\left(\frac{x}{2}\right)(d+4f)+\frac{1}{3}\text{arctanh}(x)(d+f) -\frac{1}{6}e\log\left(1-x^2\right)+\frac{1}{6}e\log\left(4-x^2\right)$$

```
[In] Int[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/6*((d + 4*f)*ArcTanh[x/2]) + ((d + f)*ArcTanh[x])/3 - (e*Log[1 - x^2])/6
 + (e*Log[4 - x^2])/6
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{ex}{4-5x^2+x^4}\,dx + \int \frac{d+fx^2}{4-5x^2+x^4}\,dx \\
&= e\int \frac{x}{4-5x^2+x^4}\,dx - \frac{1}{3}(d+f)\int \frac{1}{-1+x^2}\,dx + \frac{1}{3}(d+4f)\int \frac{1}{-4+x^2}\,dx
\end{aligned}$$

$$= -\frac{1}{6}(d+4f)\tanh^{-1}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\tanh^{-1}(x)+\frac{1}{2}e\text{Subst}\left(\int\frac{1}{4-5x+x^2}\,dx,x,x^2\right)$$

$$= -\frac{1}{6}(d+4f)\tanh^{-1}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\tanh^{-1}(x) + \frac{1}{6}e\text{Subst}\left(\int\frac{1}{-4+x}\,dx,x,x^2\right)-\frac{1}{6}e\text{Subst}\left(\int\frac{1}{-1+x}\,dx,x,x^2\right)$$

$$= -\frac{1}{6}(d+4f)\tanh^{-1}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\tanh^{-1}(x)-\frac{1}{6}e\log\left(1-x^2\right)+\frac{1}{6}e\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 58, normalized size of antiderivative = 1.14

$$\int\frac{d+ex+fx^2}{4-5x^2+x^4}\,dx=\frac{1}{12}(-2(d+e+f)\log(1-x)+(d+2e+4f)\log(2-x) + 2(d-e+f)\log(1+x)-(d-2e+4f)\log(2+x))$$

```
[In] Integrate[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4),x]
```

```
[Out] (-2*(d + e + f)*Log[1 - x] + (d + 2*e + 4*f)*Log[2 - x] + 2*(d - e + f)*Log
[1 + x] - (d - 2*e + 4*f)*Log[2 + x])/12
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 62, normalized size of antiderivative = 1.22

method	result
default	$\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}\right)\ln\left(x-1\right)+\left(\frac{d}{12}+\frac{e}{6}+\frac{f}{3}\right.$
norman	$\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}\right)\ln\left(x-1\right)+\left(\frac{d}{12}+\frac{e}{6}+\frac{f}{3}\right.$
parallelrisch	$\frac{\ln(x-2)d}{12}+\frac{\ln(x-2)e}{6}+\frac{\ln(x-2)f}{3}-\frac{\ln(x-1)d}{6}-\frac{\ln(x-1)e}{6}-\frac{\ln(x-1)f}{6}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}-$
risch	$-\frac{\ln(1-x)d}{6}-\frac{\ln(1-x)e}{6}-\frac{\ln(1-x)f}{6}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}+\frac{\ln(2-x)d}{12}+\frac{\ln(2-x)e}{6}+\frac{\ln(2-x)f}{3}-$

```
[In] int((f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
```

```
[Out] (-1/12*d+1/6*e-1/3*f)*ln(x+2)+(1/6*d-1/6*e+1/6*f)*ln(x+1)+(-1/6*d-1/6*e-1/6
*f)*ln(x-1)+(1/12*d+1/6*e+1/3*f)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 51, normalized size of antiderivative = 1.00

$$\int \frac{d+ex+fx^2}{4-5x^2+x^4}\,dx = -\frac{1}{12}\,(d-2\,e+4\,f)\log\left(x+2\right)+\frac{1}{6}\,(d-e+f)\log\left(x+1\right) -\frac{1}{6}\,(d+e+f)\log\left(x-1\right)+\frac{1}{12}\,(d+2\,e+4\,f)\log\left(x-2\right)$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] -1/12*(d - 2*e + 4*f)*log(x + 2) + 1/6*(d - e + f)*log(x + 1) - 1/6*(d + e
+ f)*log(x - 1) + 1/12*(d + 2*e + 4*f)*log(x - 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2195 vs. 2(44) = 88.

Time = 95.81 (sec) , antiderivative size = 2195, normalized size of antiderivative = 43.04

$$\int \frac{d+ex+fx^2}{4-5x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] -(d - 2*e + 4*f)*log(x + (-35*d**5*e + 51*d**5*(d - 2*e + 4*f)/2 - 820*d**4
*e*f + 90*d**4*f*(d - 2*e + 4*f) - 180*d**3*e**3 - 90*d**3*e**2*(d - 2*e +
4*f) - 4100*d**3*e*f**2 + 41*d**3*e*(d - 2*e + 4*f)**2 + 42*d**3*f**2*(d -
2*e + 4*f) - 15*d**3*(d - 2*e + 4*f)**3/2 - 432*d**2*e**2*f*(d - 2*e + 4*f)
 - 8000*d**2*e*f**3 + 240*d**2*e*f*(d - 2*e + 4*f)**2 - 240*d**2*f**3*(d -
2*e + 4*f) - 12*d**2*f*(d - 2*e + 4*f)**3 + 320*d*e**5 - 96*d*e**4*(d - 2*e
 + 4*f) + 720*d*e**3*f**2 - 80*d*e**3*(d - 2*e + 4*f)**2 - 1080*d*e**2*f**2
*(d - 2*e + 4*f) + 24*d*e**2*(d - 2*e + 4*f)**3 - 6400*d*e*f**4 + 492*d*e*f
**2*(d - 2*e + 4*f)**2 - 576*d*f**4*(d - 2*e + 4*f) + 30*d*f**2*(d - 2*e +
4*f)**3 + 512*e**5*f - 128*e**3*f*(d - 2*e + 4*f)**2 - 576*e**2*f**3*(d - 2
*e + 4*f) - 1472*e*f**5 + 320*e*f**3*(d - 2*e + 4*f)**2 - 480*f**5*(d - 2*e
 + 4*f) + 48*f**3*(d - 2*e + 4*f)**3)/(9*d**6 + 45*d**5*f - 160*d**4*e**2 -
 36*d**4*f**2 - 1312*d**3*e**2*f - 360*d**3*f**3 + 256*d**2*e**4 - 3840*d**
2*e**2*f**2 - 144*d**2*f**4 + 1280*d*e**4*f - 5248*d*e**2*f**3 + 720*d*f**5
 + 1024*e**4*f**2 - 2560*e**2*f**4 + 576*f**6))/12 + (d - e + f)*log(x + (-
35*d**5*e - 51*d**5*(d - e + f) - 820*d**4*e*f - 180*d**4*f*(d - e + f) - 1
80*d**3*e**3 + 180*d**3*e**2*(d - e + f) - 4100*d**3*e*f**2 + 164*d**3*e*(d
 - e + f)**2 - 84*d**3*f**2*(d - e + f) + 60*d**3*(d - e + f)**3 + 864*d**2
*e**2*f*(d - e + f) - 8000*d**2*e*f**3 + 960*d**2*e*f*(d - e + f)**2 + 480*
d**2*f**3*(d - e + f) + 96*d**2*f*(d - e + f)**3 + 320*d*e**5 + 192*d*e**4*
```

```
(d - e + f) + 720*d*e**3*f**2 - 320*d*e**3*(d - e + f)**2 + 2160*d*e**2*f**
2*(d - e + f) - 192*d*e**2*(d - e + f)**3 - 6400*d*e*f**4 + 1968*d*e*f**2*(
d - e + f)**2 + 1152*d*f**4*(d - e + f) - 240*d*f**2*(d - e + f)**3 + 512*e
**5*f - 512*e**3*f*(d - e + f)**2 + 1152*e**2*f**3*(d - e + f) - 1472*e*f**
5 + 1280*e*f**3*(d - e + f)**2 + 960*f**5*(d - e + f) - 384*f**3*(d - e + f
)**3)/(9*d**6 + 45*d**5*f - 160*d**4*e**2 - 36*d**4*f**2 - 1312*d**3*e**2*f
 - 360*d**3*f**3 + 256*d**2*e**4 - 3840*d**2*e**2*f**2 - 144*d**2*f**4 + 12
80*d*e**4*f - 5248*d*e**2*f**3 + 720*d*f**5 + 1024*e**4*f**2 - 2560*e**2*f*
*4 + 576*f**6))/6 - (d + e + f)*log(x + (-35*d**5*e + 51*d**5*(d + e + f) -
 820*d**4*e*f + 180*d**4*f*(d + e + f) - 180*d**3*e**3 - 180*d**3*e**2*(d +
 e + f) - 4100*d**3*e*f**2 + 164*d**3*e*(d + e + f)**2 + 84*d**3*f**2*(d +
e + f) - 60*d**3*(d + e + f)**3 - 864*d**2*e**2*f*(d + e + f) - 8000*d**2*e
*f**3 + 960*d**2*e*f*(d + e + f)**2 - 480*d**2*f**3*(d + e + f) - 96*d**2*f
*(d + e + f)**3 + 320*d*e**5 - 192*d*e**4*(d + e + f) + 720*d*e**3*f**2 - 3
20*d*e**3*(d + e + f)**2 - 2160*d*e**2*f**2*(d + e + f) + 192*d*e**2*(d + e
 + f)**3 - 6400*d*e*f**4 + 1968*d*e*f**2*(d + e + f)**2 - 1152*d*f**4*(d +
e + f) + 240*d*f**2*(d + e + f)**3 + 512*e**5*f - 512*e**3*f*(d + e + f)**2
 - 1152*e**2*f**3*(d + e + f) - 1472*e*f**5 + 1280*e*f**3*(d + e + f)**2 -
960*f**5*(d + e + f) + 384*f**3*(d + e + f)**3)/(9*d**6 + 45*d**5*f - 160*d
**4*e**2 - 36*d**4*f**2 - 1312*d**3*e**2*f - 360*d**3*f**3 + 256*d**2*e**4
- 3840*d**2*e**2*f**2 - 144*d**2*f**4 + 1280*d*e**4*f - 5248*d*e**2*f**3 +
720*d*f**5 + 1024*e**4*f**2 - 2560*e**2*f**4 + 576*f**6))/6 + (d + 2*e + 4*
f)*log(x + (-35*d**5*e - 51*d**5*(d + 2*e + 4*f)/2 - 820*d**4*e*f - 90*d**4
*f*(d + 2*e + 4*f) - 180*d**3*e**3 + 90*d**3*e**2*(d + 2*e + 4*f) - 4100*d*
*3*e*f**2 + 41*d**3*e*(d + 2*e + 4*f)**2 - 42*d**3*f**2*(d + 2*e + 4*f) + 1
5*d**3*(d + 2*e + 4*f)**3/2 + 432*d**2*e**2*f*(d + 2*e + 4*f) - 8000*d**2*e
*f**3 + 240*d**2*e*f*(d + 2*e + 4*f)**2 + 240*d**2*f**3*(d + 2*e + 4*f) + 1
2*d**2*f*(d + 2*e + 4*f)**3 + 320*d*e**5 + 96*d*e**4*(d + 2*e + 4*f) + 720*
d*e**3*f**2 - 80*d*e**3*(d + 2*e + 4*f)**2 + 1080*d*e**2*f**2*(d + 2*e + 4*
f) - 24*d*e**2*(d + 2*e + 4*f)**3 - 6400*d*e*f**4 + 492*d*e*f**2*(d + 2*e +
 4*f)**2 + 576*d*f**4*(d + 2*e + 4*f) - 30*d*f**2*(d + 2*e + 4*f)**3 + 512*
e**5*f - 128*e**3*f*(d + 2*e + 4*f)**2 + 576*e**2*f**3*(d + 2*e + 4*f) - 14
72*e*f**5 + 320*e*f**3*(d + 2*e + 4*f)**2 + 480*f**5*(d + 2*e + 4*f) - 48*f
**3*(d + 2*e + 4*f)**3)/(9*d**6 + 45*d**5*f - 160*d**4*e**2 - 36*d**4*f**2
- 1312*d**3*e**2*f - 360*d**3*f**3 + 256*d**2*e**4 - 3840*d**2*e**2*f**2 -
144*d**2*f**4 + 1280*d*e**4*f - 5248*d*e**2*f**3 + 720*d*f**5 + 1024*e**4*f
**2 - 2560*e**2*f**4 + 576*f**6))/12
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 51, normalized size of antiderivative = 1.00

$$\int \frac{d + ex + fx^2}{4 - 5x^2 + x^4}\,dx = -\frac{1}{12}\,(d - 2\,e + 4\,f)\log{(x + 2)} + \frac{1}{6}\,(d - e + f)\log{(x + 1)} - \frac{1}{6}\,(d + e + f)\log{(x - 1)} + \frac{1}{12}\,(d + 2\,e + 4\,f)\log{(x - 2)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] -1/12*(d - 2*e + 4*f)*log(x + 2) + 1/6*(d - e + f)*log(x + 1) - 1/6*(d + e
+ f)*log(x - 1) + 1/12*(d + 2*e + 4*f)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.36 (sec) , antiderivative size = 55, normalized size of antiderivative = 1.08

$$\int \frac{d + ex + fx^2}{4 - 5x^2 + x^4}\,dx = -\frac{1}{12}\,(d - 2\,e + 4\,f)\log{(|x + 2|)} + \frac{1}{6}\,(d - e + f)\log{(|x + 1|)} - \frac{1}{6}\,(d + e + f)\log{(|x - 1|)} + \frac{1}{12}\,(d + 2\,e + 4\,f)\log{(|x - 2|)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] -1/12*(d - 2*e + 4*f)*log(abs(x + 2)) + 1/6*(d - e + f)*log(abs(x + 1)) - 1
/6*(d + e + f)*log(abs(x - 1)) + 1/12*(d + 2*e + 4*f)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.76 (sec) , antiderivative size = 63, normalized size of antiderivative = 1.24

$$\int \frac{d + ex + fx^2}{4 - 5x^2 + x^4}\,dx = \ln{(x + 1)}\left(\frac{d}{6} - \frac{e}{6} + \frac{f}{6}\right) - \ln{(x - 1)}\left(\frac{d}{6} + \frac{e}{6} + \frac{f}{6}\right) + \ln{(x - 2)}\left(\frac{d}{12} + \frac{e}{6} + \frac{f}{3}\right) - \ln{(x + 2)}\left(\frac{d}{12} - \frac{e}{6} + \frac{f}{3}\right)$$

```
[In] int((d + e*x + f*x^2)/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x + 1)*(d/6 - e/6 + f/6) - log(x - 1)*(d/6 + e/6 + f/6) + log(x - 2)*(d
/12 + e/6 + f/3) - log(x + 2)*(d/12 - e/6 + f/3)
```

3.12 $\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 28, antiderivative size = 57

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4} \, dx = -\frac{1}{6}(d+4f)\text{arctanh}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\text{arctanh}(x) -\frac{1}{6}(e+g)\log\left(1-x^2\right)+\frac{1}{6}(e+4g)\log\left(4-x^2\right)$$

```
[Out] -1/6*(d+4*f)*arctanh(1/2*x)+1/3*(d+f)*arctanh(x)-1/6*(e+g)*ln(-x^2+1)+1/6*(
e+4*g)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 57, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.214, Rules used = {1687, 1180, 213, 1261, 646, 31}

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4} \, dx = -\frac{1}{6}\text{arctanh}\left(\frac{x}{2}\right)(d+4f)+\frac{1}{3}\text{arctanh}(x)(d+f) -\frac{1}{6}(e+g)\log\left(1-x^2\right)+\frac{1}{6}(e+4g)\log\left(4-x^2\right)$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/6*((d + 4*f)*ArcTanh[x/2]) + ((d + f)*ArcTanh[x])/3 - ((e + g)*Log[1 - x
^2])/6 + ((e + 4*g)*Log[4 - x^2])/6
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{4-5x^2+x^4}\,dx+\int \frac{x(e+gx^2)}{4-5x^2+x^4}\,dx \\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{4-5x+x^2}\,dx,x,x^2\right)-\frac{1}{3}(d+f)\int \frac{1}{-1+x^2}\,dx+\frac{1}{3}(d+4f)\int \frac{1}{-4+x^2}\,dx \\
&= -\frac{1}{6}(d+4f)\tanh^{-1}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\tanh^{-1}(x) \\
&\quad +\frac{1}{6}(-e-g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx,x,x^2\right)+\frac{1}{6}(e+4g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx,x,x^2\right)
\end{aligned}$$

$$= -\frac{1}{6}(d+4f)\tanh^{-1}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f)\tanh^{-1}(x)-\frac{1}{6}(e+g)\log\left(1-x^2\right)+\frac{1}{6}(e+4g)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 68, normalized size of antiderivative = 1.19

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = \frac{1}{12}(-2(d+e+f+g)\log(1-x)+(d+2e+4f+8g)\log(2-x)+2(d-e+f-g)\log(1+x)-(d-2e+4f-8g)\log(2+x))$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4),x]
[Out] (-2*(d + e + f + g)*Log[1 - x] + (d + 2*e + 4*f + 8*g)*Log[2 - x] + 2*(d -
e + f - g)*Log[1 + x] - (d - 2*e + 4*f - 8*g)*Log[2 + x])/12
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 74, normalized size of antiderivative = 1.30

method	result
default	$\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}-\frac{g}{6}\right)\ln\left(x-1\right)+$
norman	$\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}-\frac{g}{6}\right)\ln\left(x-1\right)+$
parallelrisch	$\frac{\ln(x-2)d}{12}+\frac{\ln(x-2)e}{6}+\frac{\ln(x-2)f}{3}+\frac{2\ln(x-2)g}{3}-\frac{\ln(x-1)d}{6}-\frac{\ln(x-1)e}{6}-\frac{\ln(x-1)f}{6}-\frac{\ln(x-1)g}{6}+\frac{\ln(x+1)d}{6}-$
risch	$-\frac{\ln(x+2)d}{12}+\frac{\ln(x+2)e}{6}-\frac{\ln(x+2)f}{3}+\frac{2\ln(x+2)g}{3}+\frac{\ln(2-x)d}{12}+\frac{\ln(2-x)e}{6}+\frac{\ln(2-x)f}{3}+\frac{2\ln(2-x)g}{3}+\frac{\ln(x+1)d}{6}$

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] (-1/12*d+1/6*e-1/3*f+2/3*g)*ln(x+2)+(1/6*d-1/6*e+1/6*f-1/6*g)*ln(x+1)+(-1/6
*d-1/6*e-1/6*f-1/6*g)*ln(x-1)+(1/12*d+1/6*e+1/3*f+2/3*g)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 61, normalized size of antiderivative = 1.07

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = -\frac{1}{12}\,(d-2\,e+4\,f-8\,g)\log\left(x+2\right)+\frac{1}{6}\,(d-e+f-g)\log\left(x+1\right)-\frac{1}{6}\,(d+e+f+g)\log\left(x-1\right)+\frac{1}{12}\,(d+2\,e+4\,f+8\,g)\log\left(x-2\right)$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
```

```
[Out] -1/12*(d - 2*e + 4*f - 8*g)*log(x + 2) + 1/6*(d - e + f - g)*log(x + 1) - 1
/6*(d + e + f + g)*log(x - 1) + 1/12*(d + 2*e + 4*f + 8*g)*log(x - 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 61, normalized size of antiderivative = 1.07

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = & -\frac{1}{12}\,(d-2\,e+4\,f-8\,g)\log\left(x+2\right)\\ & +\frac{1}{6}\,(d-e+f-g)\log\left(x+1\right)-\frac{1}{6}\,(d+e+f+g)\log\left(x-1\right)\\ & +\frac{1}{12}\,(d+2\,e+4\,f+8\,g)\log\left(x-2\right)\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] -1/12*(d - 2*e + 4*f - 8*g)*log(x + 2) + 1/6*(d - e + f - g)*log(x + 1) - 1
/6*(d + e + f + g)*log(x - 1) + 1/12*(d + 2*e + 4*f + 8*g)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 65, normalized size of antiderivative = 1.14

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = & -\frac{1}{12}\,(d-2\,e+4\,f-8\,g)\log\left(|x+2|\right)\\ & +\frac{1}{6}\,(d-e+f-g)\log\left(|x+1|\right)\\ & -\frac{1}{6}\,(d+e+f+g)\log\left(|x-1|\right)\\ & +\frac{1}{12}\,(d+2\,e+4\,f+8\,g)\log\left(|x-2|\right)\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] -1/12*(d - 2*e + 4*f - 8*g)*log(abs(x + 2)) + 1/6*(d - e + f - g)*log(abs(x
 + 1)) - 1/6*(d + e + f + g)*log(abs(x - 1)) + 1/12*(d + 2*e + 4*f + 8*g)*l
og(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.80 (sec) , antiderivative size = 75, normalized size of antiderivative = 1.32

$$\int \frac{d+ex+fx^2+gx^3}{4-5x^2+x^4}\,dx = \ln(x+1)\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)-\ln(x-1)\left(\frac{d}{6}+\frac{e}{6}+\frac{f}{6}+\frac{g}{6}\right)+\ln(x-2)\left(\frac{d}{12}+\frac{e}{6}+\frac{f}{3}+\frac{2g}{3}\right)-\ln(x+2)\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x + 1)*(d/6 - e/6 + f/6 - g/6) - log(x - 1)*(d/6 + e/6 + f/6 + g/6) + l
og(x - 2)*(d/12 + e/6 + f/3 + (2*g)/3) - log(x + 2)*(d/12 - e/6 + f/3 - (2*
g)/3)
```

3.13 $\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 33, antiderivative size = 64

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4} dx = hx-\frac{1}{6}(d+4f+16h)\text{arctanh}\left(\frac{x}{2}\right)+\frac{1}{3}(d+f+h)\text{arctanh}(x) -\frac{1}{6}(e+g)\log\left(1-x^2\right)+\frac{1}{6}(e+4g)\log\left(4-x^2\right)$$

```
[Out] h*x-1/6*(d+4*f+16*h)*arctanh(1/2*x)+1/3*(d+f+h)*arctanh(x)-1/6*(e+g)*ln(-x^
2+1)+1/6*(e+4*g)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 64, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.212, Rules used = {1687, 1690, 1180, 213, 1261, 646, 31}

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4} dx = -\frac{1}{6}\text{arctanh}\left(\frac{x}{2}\right)(d+4f+16h)+\frac{1}{3}\text{arctanh}(x)(d+f+h) -\frac{1}{6}(e+g)\log\left(1-x^2\right)+\frac{1}{6}(e+4g)\log\left(4-x^2\right)+hx$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4),x]
```

```
[Out] h*x - ((d + 4*f + 16*h)*ArcTanh[x/2])/6 + ((d + f + h)*ArcTanh[x])/3 - ((e
+ g)*Log[1 - x^2])/6 + ((e + 4*g)*Log[4 - x^2])/6
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$\text{integral} = \int \frac{x(e+gx^2)}{4-5x^2+x^4}\,dx + \int \frac{d+fx^2+hx^4}{4-5x^2+x^4}\,dx$$

$$= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{4-5x+x^2}\,dx, x, x^2\right) + \int \left(h + \frac{d-4h+(f+5h)x^2}{4-5x^2+x^4}\right) dx$$

$$= hx + \frac{1}{6}(-e-g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) + \frac{1}{6}(e+4g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) + \int \frac{d-4h+(f+5h)x^2}{4-5x^2+x^4}\,dx$$

$$= hx - \frac{1}{6}(e+g)\log\left(1-x^2\right) + \frac{1}{6}(e+4g)\log\left(4-x^2\right) - \frac{1}{3}(d+f+h)\int \frac{1}{-1+x^2}\,dx + \frac{1}{3}(d+4f+16h)\int \frac{1}{-4+x^2}\,dx$$

$$= hx - \frac{1}{6}(d+4f+16h)\tanh^{-1}\left(\frac{x}{2}\right) + \frac{1}{3}(d+f+h)\tanh^{-1}(x) - \frac{1}{6}(e+g)\log\left(1-x^2\right) + \frac{1}{6}(e+4g)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 81, normalized size of antiderivative = 1.27

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = \frac{1}{12}(12hx - 2(d+e+f+g+h)\log(1-x) + (d+2(e+2f+4g+8h))\log(2-x) + 2(d-e+f-g+h)\log(1+x) - (d-2e+4f-8g+16h)\log(2+x))$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4),x]
```

```
[Out] (12*h*x - 2*(d + e + f + g + h)*Log[1 - x] + (d + 2*(e + 2*f + 4*g + 8*h))*
Log[2 - x] + 2*(d - e + f - g + h)*Log[1 + x] - (d - 2*e + 4*f - 8*g + 16*h
)*Log[2 + x])/12
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 89, normalized size of antiderivative = 1.39

method	result
default	$hx+\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}\right.$
norman	$hx+\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{6}-\frac{e}{6}-\frac{f}{6}\right.$
parallelrisch	$hx+\frac{\ln(x-2)d}{12}+\frac{\ln(x-2)e}{6}+\frac{\ln(x-2)f}{3}+\frac{2\ln(x-2)g}{3}+\frac{4\ln(x-2)h}{3}-\frac{\ln(x-1)d}{6}-\frac{\ln(x-1)e}{6}-\frac{\ln(x-1)f}{6}-\ln$
risch	$hx-\frac{\ln(1-x)d}{6}-\frac{\ln(1-x)e}{6}-\frac{\ln(1-x)f}{6}-\frac{\ln(1-x)g}{6}-\frac{\ln(1-x)h}{6}+\frac{\ln(2-x)d}{12}+\frac{\ln(2-x)e}{6}+\frac{\ln(2-x)f}{3}+2\ln(2$

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
```

```
[Out] h*x+(-1/12*d+1/6*e-1/3*f+2/3*g-4/3*h)*ln(x+2)+(1/6*d-1/6*e+1/6*f-1/6*g+1/6*
h)*ln(x+1)+(-1/6*d-1/6*e-1/6*f-1/6*g-1/6*h)*ln(x-1)+(1/12*d+1/6*e+1/3*f+2/3
*g+4/3*h)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 1.14 (sec) , antiderivative size = 72, normalized size of antiderivative = 1.12

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = hx - \frac{1}{12}(d-2e+4f-8g+16h)\log(x+2) + \frac{1}{6}(d-e+f-g+h)\log(x+1) - \frac{1}{6}(d+e+f+g+h)\log(x-1) + \frac{1}{12}(d+2e+4f+8g+16h)\log(x-2)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
```

```
[Out] h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h)*log(x + 2) + 1/6*(d - e + f - g + h
)*log(x + 1) - 1/6*(d + e + f + g + h)*log(x - 1) + 1/12*(d + 2*e + 4*f + 8
*g + 16*h)*log(x - 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 72, normalized size of antiderivative = 1.12

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = hx - \frac{1}{12}(d-2e+4f-8g+16h)\log(x+2) + \frac{1}{6}(d-e+f-g+h)\log(x+1) - \frac{1}{6}(d+e+f+g+h)\log(x-1) + \frac{1}{12}(d+2e+4f+8g+16h)\log(x-2)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h)*log(x + 2) + 1/6*(d - e + f - g + h
)*log(x + 1) - 1/6*(d + e + f + g + h)*log(x - 1) + 1/12*(d + 2*e + 4*f + 8
*g + 16*h)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 76, normalized size of antiderivative = 1.19

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = hx - \frac{1}{12}\,(d-2\,e+4\,f-8\,g+16\,h)\log\left(|x+2|\right) + \frac{1}{6}\,(d-e+f-g+h)\log\left(|x+1|\right) - \frac{1}{6}\,(d+e+f+g+h)\log\left(|x-1|\right) + \frac{1}{12}\,(d+2\,e+4\,f+8\,g+16\,h)\log\left(|x-2|\right)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h)*log(abs(x + 2)) + 1/6*(d - e + f -
g + h)*log(abs(x + 1)) - 1/6*(d + e + f + g + h)*log(abs(x - 1)) + 1/12*(d
+ 2*e + 4*f + 8*g + 16*h)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.90 (sec) , antiderivative size = 90, normalized size of antiderivative = 1.41

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{4-5x^2+x^4}\,dx = h\,x - \ln\left(x-1\right)\left(\frac{d}{6}+\frac{e}{6}+\frac{f}{6}+\frac{g}{6}+\frac{h}{6}\right) + \ln\left(x+1\right)\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right) + \ln\left(x-2\right)\left(\frac{d}{12}+\frac{e}{6}+\frac{f}{3}+\frac{2\,g}{3}+\frac{4\,h}{3}\right) - \ln\left(x+2\right)\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2\,g}{3}+\frac{4\,h}{3}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^4 - 5*x^2 + 4),x)
```

```
[Out] h*x - log(x - 1)*(d/6 + e/6 + f/6 + g/6 + h/6) + log(x + 1)*(d/6 - e/6 + f/
6 - g/6 + h/6) + log(x - 2)*(d/12 + e/6 + f/3 + (2*g)/3 + (4*h)/3) - log(x
+ 2)*(d/12 - e/6 + f/3 - (2*g)/3 + (4*h)/3)
```

3.14 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 38, antiderivative size = 76

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4} dx = hx + \frac{ix^2}{2} - \frac{1}{6}(d+4f+16h)\text{arctanh}\left(\frac{x}{2}\right) + \frac{1}{3}(d+f+h)\text{arctanh}(x) - \frac{1}{6}(e+g+i)\log\left(1-x^2\right) + \frac{1}{6}(e+4g+16i)\log\left(4-x^2\right)$$

```
[Out] h*x+1/2*i*x^2-1/6*(d+4*f+16*h)*arctanh(1/2*x)+1/3*(d+f+h)*arctanh(x)-1/6*(e
+g+i)*ln(-x^2+1)+1/6*(e+4*g+16*i)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 76, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.211, Rules used = {1687, 1690, 1180, 213, 1677, 1671, 646, 31}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4} dx = -\frac{1}{6}\text{arctanh}\left(\frac{x}{2}\right)(d+4f+16h) + \frac{1}{3}\text{arctanh}(x)(d+f+h) - \frac{1}{6}\log\left(1-x^2\right)(e+g+i) + \frac{1}{6}\log\left(4-x^2\right)(e+4g+16i) + hx + \frac{ix^2}{2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4),x]
```

```
[Out] h*x + (i*x^2)/2 - ((d + 4*f + 16*h)*ArcTanh[x/2])/6 + ((d + f + h)*ArcTanh[
x])/3 - ((e + g + i)*Log[1 - x^2])/6 + ((e + 4*g + 16*i)*Log[4 - x^2])/6
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
```

```
&& !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2+hx^4}{4-5x^2+x^4}\,dx + \int \frac{x(e+gx^2+ix^4)}{4-5x^2+x^4}\,dx \\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{4-5x+x^2}\,dx, x, x^2\right) + \int \left(h + \frac{d-4h+(f+5h)x^2}{4-5x^2+x^4}\right)\,dx \\
&= hx + \frac{1}{2}\text{Subst}\left(\int \left(i + \frac{e-4i+(g+5i)x}{4-5x+x^2}\right)\,dx, x, x^2\right) + \int \frac{d-4h+(f+5h)x^2}{4-5x^2+x^4}\,dx \\
&= hx + \frac{ix^2}{2} + \frac{1}{2}\text{Subst}\left(\int \frac{e-4i+(g+5i)x}{4-5x+x^2}\,dx, x, x^2\right) \\
&\quad - \frac{1}{3}(d+f+h)\int \frac{1}{-1+x^2}\,dx + \frac{1}{3}(d+4f+16h)\int \frac{1}{-4+x^2}\,dx \\
&= hx + \frac{ix^2}{2} - \frac{1}{6}(d+4f+16h)\tanh^{-1}\left(\frac{x}{2}\right) + \frac{1}{3}(d+f+h)\tanh^{-1}(x) \\
&\quad + \frac{1}{6}(-e-g-i)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) \\
&\quad + \frac{1}{6}(e+4g+16i)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) \\
&= hx + \frac{ix^2}{2} - \frac{1}{6}(d+4f+16h)\tanh^{-1}\left(\frac{x}{2}\right) + \frac{1}{3}(d+f+h)\tanh^{-1}(x) \\
&\quad - \frac{1}{6}(e+g+i)\log\left(1-x^2\right) + \frac{1}{6}(e+4g+16i)\log\left(4-x^2\right)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 98, normalized size of antiderivative = 1.29

$$\begin{aligned}
\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx &= \frac{1}{12}\big(12hx+6ix^2-2(d+e+f+g+h+i)\log(1-x) \\
&\quad + (d+2e+4(f+2g+4h+8i))\log(2-x) \\
&\quad + 2(d-e+f-g+h-i)\log(1+x) \\
&\quad - (d-2(e-2f+4g-8h+16i))\log(2+x)\big)
\end{aligned}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4),x]
```

```
[Out] (12*h*x + 6*i*x^2 - 2*(d + e + f + g + h + i)*Log[1 - x] + (d + 2*e + 4*(f
+ 2*g + 4*h + 8*i))*Log[2 - x] + 2*(d - e + f - g + h - i)*Log[1 + x] - (d
- 2*(e - 2*f + 4*g - 8*h + 16*i))*Log[2 + x])/12
```

Maple [A] (verified)

Time = 0.09 (sec) , antiderivative size = 107, normalized size of antiderivative = 1.41

method	result
default	$\frac{i x^2}{2}+hx+\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}+\frac{8i}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right)\ln\left(x+1\right)$ -
norman	$\frac{i x^2}{2}+hx+\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}+\frac{8i}{3}\right)\ln\left(x+2\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right)\ln\left(x+1\right)$ -
parallelrisch	$\frac{i x^2}{2}+\frac{\ln(x-2)d}{12}+\frac{\ln(x-2)e}{6}-\frac{\ln(x-1)d}{6}-\frac{\ln(x-1)e}{6}-\frac{\ln(x+1)i}{6}-\frac{\ln(x+2)f}{3}+\frac{\ln(x+1)f}{6}-\frac{\ln(x+2)d}{12}+hx$ -
risch	$\frac{i x^2}{2}-\frac{\ln(x+1)i}{6}-\frac{\ln(x+2)f}{3}-\frac{\ln(1-x)f}{6}+\frac{\ln(x+1)f}{6}+\frac{\ln(2-x)f}{3}+\frac{\ln(2-x)d}{12}+\frac{\ln(2-x)e}{6}-\frac{\ln(x+2)d}{12}+hx$ -

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
```

```
[Out] 1/2*i*x^2+h*x+(-1/12*d+1/6*e-1/3*f+2/3*g-4/3*h+8/3*i)*ln(x+2)+(1/6*d-1/6*e+
1/6*f-1/6*g+1/6*h-1/6*i)*ln(x+1)+(-1/6*d-1/6*e-1/6*f-1/6*g-1/6*h-1/6*i)*ln(
x-1)+(1/12*d+1/6*e+1/3*f+2/3*g+4/3*h+8/3*i)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 5.00 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.16

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx = \frac{1}{2}\,ix^2+hx - \frac{1}{12}\,(d-2\,e+4\,f-8\,g+16\,h-32\,i)\log\left(x+2\right) + \frac{1}{6}\,(d-e+f-g+h-i)\log\left(x+1\right) - \frac{1}{6}\,(d+e+f+g+h+i)\log\left(x-1\right) + \frac{1}{12}\,(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log\left(x-2\right)$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="frica
s")
```

```
[Out] 1/2*i*x^2 + h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(x + 2) + 1/6
*(d - e + f - g + h - i)*log(x + 1) - 1/6*(d + e + f + g + h + i)*log(x - 1
) + 1/12*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(x - 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx=\text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.16

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx=\frac{1}{2}ix^2+hx-\frac{1}{12}(d-2e+4f-8g+16h-32i)\log(x+2)+\frac{1}{6}(d-e+f-g+h-i)\log(x+1)-\frac{1}{6}(d+e+f+g+h+i)\log(x-1)+\frac{1}{12}(d+2e+4f+8g+16h+32i)\log(x-2)$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxim
a")
[Out] 1/2*i*x^2 + h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(x + 2) + 1/6
*(d - e + f - g + h - i)*log(x + 1) - 1/6*(d + e + f + g + h + i)*log(x - 1
) + 1/12*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.21

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx=\frac{1}{2}ix^2+hx-\frac{1}{12}(d-2e+4f-8g+16h-32i)\log(|x+2|)+\frac{1}{6}(d-e+f-g+h-i)\log(|x+1|)-\frac{1}{6}(d+e+f+g+h+i)\log(|x-1|)+\frac{1}{12}(d+2e+4f+8g+16h+32i)\log(|x-2|)$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac"
)
```

```
[Out] 1/2*i*x^2 + h*x - 1/12*(d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(abs(x + 2))
+ 1/6*(d - e + f - g + h - i)*log(abs(x + 1)) - 1/6*(d + e + f + g + h + i)
*log(abs(x - 1)) + 1/12*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 8.15 (sec) , antiderivative size = 108, normalized size of antiderivative = 1.42

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{4-5x^2+x^4}\,dx = h\,x+\frac{i\,x^2}{2}-\ln\left(x-1\right)\,\left(\frac{d}{6}+\frac{e}{6}+\frac{f}{6}+\frac{g}{6}+\frac{h}{6}+\frac{i}{6}\right) + \ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right) + \ln\left(x-2\right)\,\left(\frac{d}{12}+\frac{e}{6}+\frac{f}{3}+\frac{2\,g}{3}+\frac{4\,h}{3}+\frac{8\,i}{3}\right) - \ln\left(x+2\right)\,\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2\,g}{3}+\frac{4\,h}{3}-\frac{8\,i}{3}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^4 - 5*x^2 + 4),x)
```

```
[Out] h*x + (i*x^2)/2 - log(x - 1)*(d/6 + e/6 + f/6 + g/6 + h/6 + i/6) + log(x +
1)*(d/6 - e/6 + f/6 - g/6 + h/6 - i/6) + log(x - 2)*(d/12 + e/6 + f/3 + (2*
g)/3 + (4*h)/3 + (8*i)/3) - log(x + 2)*(d/12 - e/6 + f/3 - (2*g)/3 + (4*h)/
3 - (8*i)/3)
```

3.15 $\int \frac{d+ex}{1+x^2+x^4}\,dx$

Optimal result

Integrand size = 16, antiderivative size = 92

$$\int \frac{d+ex}{1+x^2+x^4}\,dx=-\frac{d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{d\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{\sqrt{3}}-\frac{1}{4}d\log\left(1-x+x^2\right)+\frac{1}{4}d\log\left(1+x+x^2\right)$$

```
[Out] -1/4*d*ln(x^2-x+1)+1/4*d*ln(x^2+x+1)-1/6*d*arctan(1/3*(1-2*x)*3^(1/2))*3^(1
/2)+1/6*d*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/3*e*arctan(1/3*(2*x^2+1)*3^
(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.500, Rules used = {1687, 12, 1108, 648, 632, 210, 642, 1121}

$$\int \frac{d+ex}{1+x^2+x^4}\,dx=-\frac{d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{d\arctan\left(\frac{2x+1}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{\sqrt{3}}-\frac{1}{4}d\log\left(x^2-x+1\right)+\frac{1}{4}d\log\left(x^2+x+1\right)$$

```
[In] Int[(d + e*x)/(1 + x^2 + x^4),x]
```

```
[Out] -1/2*(d*ArcTan[(1 - 2*x)/Sqrt[3]])/Sqrt[3] + (d*ArcTan[(1 + 2*x)/Sqrt[3]])/
(2*Sqrt[3]) + (e*ArcTan[(1 + 2*x^2)/Sqrt[3]])/Sqrt[3] - (d*Log[1 - x + x^2]
)/4 + (d*Log[1 + x + x^2])/4
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1108

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(-1), x_Symbol] :> With[{q = Rt[a/
c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int[(r - x)/(q - r*x
 + x^2), x], x] + Dist[1/(2*c*q*r), Int[(r + x)/(q + r*x + x^2), x], x]]] /
; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
```

```
&& !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d}{1+x^2+x^4}\,dx + \int \frac{ex}{1+x^2+x^4}\,dx \\
&= d\int \frac{1}{1+x^2+x^4}\,dx + e\int \frac{x}{1+x^2+x^4}\,dx \\
&= \frac{1}{2}d\int \frac{1-x}{1-x+x^2}\,dx + \frac{1}{2}d\int \frac{1+x}{1+x+x^2}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&= \frac{1}{4}d\int \frac{1}{1-x+x^2}\,dx - \frac{1}{4}d\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{4}d\int \frac{1}{1+x+x^2}\,dx \\
&\quad + \frac{1}{4}d\int \frac{1+2x}{1+x+x^2}\,dx - e\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) \\
&= \frac{e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{\sqrt{3}} - \frac{1}{4}d\log\left(1-x+x^2\right) + \frac{1}{4}d\log\left(1+x+x^2\right) \\
&\quad - \frac{1}{2}d\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) - \frac{1}{2}d\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) \\
&= -\frac{d\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{d\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{\sqrt{3}} \\
&\quad - \frac{1}{4}d\log\left(1-x+x^2\right) + \frac{1}{4}d\log\left(1+x+x^2\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.17 (sec) , antiderivative size = 98, normalized size of antiderivative = 1.07

$$
\int \frac{d+ex}{1+x^2+x^4}\,dx = \frac{1}{6}i\left(\sqrt{6-6i\sqrt{3}}d\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right) - \sqrt{6+6i\sqrt{3}}d\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right) + 2i\sqrt{3}e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)
$$

```
[In] Integrate[(d + e*x)/(1 + x^2 + x^4),x]
```

```
[Out] (I/6)*(Sqrt[6 - (6*I)*Sqrt[3]]*d*ArcTan[((-I + Sqrt[3])*x)/2] - Sqrt[6 + (6
*I)*Sqrt[3]]*d*ArcTan[((I + Sqrt[3])*x)/2] + (2*I)*Sqrt[3]*e*ArcTan[Sqrt[3]
/(1 + 2*x^2)])
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 68, normalized size of antiderivative = 0.74

method	result
default	$-\frac{d\ln\left(x^{2}-x+1\right)}{4}+\frac{\left(\frac{d}{2}+e\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{3}+\frac{d\ln\left(x^{2}+x+1\right)}{4}+\frac{\left(\frac{d}{2}-e\right)\arctan\left(\frac{(1+2x)\sqrt{3}}{3}\right)\sqrt{3}}{3}$
risch	$\frac{d\ln\left(36d^{2}x^{2}+48e^{2}x^{2}+36d^{2}x+48e^{2}x+36d^{2}+48e^{2}\right)}{4}+\frac{\sqrt{3}\,d\arctan\left(\frac{8e^{2}\sqrt{3}\,x}{3\left(3d^{2}+4e^{2}\right)}-\frac{4e^{2}\sqrt{3}}{3\left(3d^{2}+4e^{2}\right)}+\frac{2d^{2}\sqrt{3}\,x}{3d^{2}+4e^{2}}-\frac{d^{2}\sqrt{3}}{3d^{2}+4e^{2}}\right)}{6}-\frac{\sqrt{3}\,d\,\text{arcta}}{3}$

```
[In] int((e*x+d)/(x^4+x^2+1),x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*d*ln(x^2-x+1)+1/3*(1/2*d+e)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*d*
ln(x^2+x+1)+1/3*(1/2*d-e)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.71

$$\begin{aligned}\int \frac{d+ex}{1+x^2+x^4}\,dx =& \frac{1}{6}\sqrt{3}(d-2\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) \\ &+ \frac{1}{6}\sqrt{3}(d+2\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) \\ &+ \frac{1}{4}\,d\log\left(x^2+x+1\right)-\frac{1}{4}\,d\log\left(x^2-x+1\right)\end{aligned}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1),x, algorithm="fricas")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d + 2*e)
*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*d*log(x^2 + x + 1) - 1/4*d*log(x^2 - x
 + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.66 (sec) , antiderivative size = 923, normalized size of antiderivative = 10.03

$$\int \frac{d+ex}{1+x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(x**4+x**2+1),x)
```

```
[Out] (-d/4 - sqrt(3)*I*(d + 2*e)/12)*log(x + (-7*d**4*e + 6*d**4*(-d/4 - sqrt(3)
*I*(d + 2*e)/12) - 15*d**2*e**3 - 18*d**2*e**2*(-d/4 - sqrt(3)*I*(d + 2*e)/
```

```
12) + 60*d**2*e*(-d/4 - sqrt(3)*I*(d + 2*e)/12)**2 + 72*d**2*(-d/4 - sqrt(3
)*I*(d + 2*e)/12)**3 + 4*e**5 + 24*e**4*(-d/4 - sqrt(3)*I*(d + 2*e)/12) + 4
8*e**3*(-d/4 - sqrt(3)*I*(d + 2*e)/12)**2 + 288*e**2*(-d/4 - sqrt(3)*I*(d +
 2*e)/12)**3)/(3*d**5 - 8*d**3*e**2 - 16*d*e**4)) + (-d/4 + sqrt(3)*I*(d +
2*e)/12)*log(x + (-7*d**4*e + 6*d**4*(-d/4 + sqrt(3)*I*(d + 2*e)/12) - 15*d
**2*e**3 - 18*d**2*e**2*(-d/4 + sqrt(3)*I*(d + 2*e)/12) + 60*d**2*e*(-d/4 +
 sqrt(3)*I*(d + 2*e)/12)**2 + 72*d**2*(-d/4 + sqrt(3)*I*(d + 2*e)/12)**3 +
4*e**5 + 24*e**4*(-d/4 + sqrt(3)*I*(d + 2*e)/12) + 48*e**3*(-d/4 + sqrt(3)*
I*(d + 2*e)/12)**2 + 288*e**2*(-d/4 + sqrt(3)*I*(d + 2*e)/12)**3)/(3*d**5 -
 8*d**3*e**2 - 16*d*e**4)) + (d/4 - sqrt(3)*I*(d - 2*e)/12)*log(x + (-7*d**
4*e + 6*d**4*(d/4 - sqrt(3)*I*(d - 2*e)/12) - 15*d**2*e**3 - 18*d**2*e**2*(
d/4 - sqrt(3)*I*(d - 2*e)/12) + 60*d**2*e*(d/4 - sqrt(3)*I*(d - 2*e)/12)**2
 + 72*d**2*(d/4 - sqrt(3)*I*(d - 2*e)/12)**3 + 4*e**5 + 24*e**4*(d/4 - sqrt
(3)*I*(d - 2*e)/12) + 48*e**3*(d/4 - sqrt(3)*I*(d - 2*e)/12)**2 + 288*e**2*
(d/4 - sqrt(3)*I*(d - 2*e)/12)**3)/(3*d**5 - 8*d**3*e**2 - 16*d*e**4)) + (d
/4 + sqrt(3)*I*(d - 2*e)/12)*log(x + (-7*d**4*e + 6*d**4*(d/4 + sqrt(3)*I*(
d - 2*e)/12) - 15*d**2*e**3 - 18*d**2*e**2*(d/4 + sqrt(3)*I*(d - 2*e)/12) +
 60*d**2*e*(d/4 + sqrt(3)*I*(d - 2*e)/12)**2 + 72*d**2*(d/4 + sqrt(3)*I*(d
- 2*e)/12)**3 + 4*e**5 + 24*e**4*(d/4 + sqrt(3)*I*(d - 2*e)/12) + 48*e**3*(
d/4 + sqrt(3)*I*(d - 2*e)/12)**2 + 288*e**2*(d/4 + sqrt(3)*I*(d - 2*e)/12)*
*3)/(3*d**5 - 8*d**3*e**2 - 16*d*e**4))
```

Maxima [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.71

$$\int \frac{d+ex}{1+x^2+x^4}\,dx = \frac{1}{6}\sqrt{3}(d-2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{6}\sqrt{3}(d+2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{4}d\log\left(x^2+x+1\right) - \frac{1}{4}d\log\left(x^2-x+1\right)$$

```
[In] integrate((e*x+d)/(x^4+x^2+1),x, algorithm="maxima")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d + 2*e)
*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*d*log(x^2 + x + 1) - 1/4*d*log(x^2 - x
 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.71

$$\int \frac{d+ex}{1+x^2+x^4}\,dx = \frac{1}{6}\sqrt{3}(d-2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{6}\sqrt{3}(d+2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{4}d\log\left(x^2+x+1\right) - \frac{1}{4}d\log\left(x^2-x+1\right)$$

```
[In] integrate((e*x+d)/(x^4+x^2+1),x, algorithm="giac")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d + 2*e)
*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*d*log(x^2 + x + 1) - 1/4*d*log(x^2 - x
 + 1)
```

Mupad [B] (verification not implemented)

Time = 0.13 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.28

$$\int \frac{d+ex}{1+x^2+x^4}\,dx = -\ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}\right) + \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}\right)$$

```
[In] int((d + e*x)/(x^2 + x^4 + 1),x)
```

```
[Out] log(x - (3^(1/2)*1i)/2 + 1/2)*(d/4 - (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6)
- log(x - (3^(1/2)*1i)/2 - 1/2)*(d/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6
) + log(x + (3^(1/2)*1i)/2 - 1/2)*((3^(1/2)*d*1i)/12 - d/4 + (3^(1/2)*e*1i)
/6) + log(x + (3^(1/2)*1i)/2 + 1/2)*(d/4 + (3^(1/2)*d*1i)/12 - (3^(1/2)*e*1
i)/6)
```

3.16 $\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx$

Optimal result

Integrand size = 21, antiderivative size = 104

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx=-\frac{(d+f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(d+f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{\sqrt{3}}-\frac{1}{4}(d-f)\log\left(1-x+x^2\right)+\frac{1}{4}(d-f)\log\left(1+x+x^2\right)$$

```
[Out] -1/4*(d-f)*ln(x^2-x+1)+1/4*(d-f)*ln(x^2+x+1)-1/6*(d+f)*arctan(1/3*(1-2*x)*3
^(1/2))*3^(1/2)+1/6*(d+f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/3*e*arctan(
1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 104, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.381, Rules used = {1687, 1183, 648, 632, 210, 642, 12, 1121}

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx=-\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(d+f)}{2\sqrt{3}}+\frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(d+f)}{2\sqrt{3}}+\frac{e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{\sqrt{3}}-\frac{1}{4}(d-f)\log\left(x^2-x+1\right)+\frac{1}{4}(d-f)\log\left(x^2+x+1\right)$$

```
[In] Int[(d + e*x + f*x^2)/(1 + x^2 + x^4),x]
```

```
[Out] -1/2*((d + f)*ArcTan[(1 - 2*x)/Sqrt[3]])/Sqrt[3] + ((d + f)*ArcTan[(1 + 2*x
)/Sqrt[3]])/(2*Sqrt[3]) + (e*ArcTan[(1 + 2*x^2)/Sqrt[3]])/Sqrt[3] - ((d - f
)*Log[1 - x + x^2])/4 + ((d - f)*Log[1 + x + x^2])/4
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
```

```
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{ex}{1+x^2+x^4}\,dx + \int \frac{d+fx^2}{1+x^2+x^4}\,dx \\
&= \frac{1}{2}\int \frac{d-(d-f)x}{1-x+x^2}\,dx + \frac{1}{2}\int \frac{d+(d-f)x}{1+x+x^2}\,dx + e\int \frac{x}{1+x^2+x^4}\,dx \\
&= \frac{1}{2}e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&\quad + \frac{1}{4}(d-f)\int \frac{1+2x}{1+x+x^2}\,dx + \frac{1}{4}(-d+f)\int \frac{-1+2x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{4}(d+f)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{4}(d+f)\int \frac{1}{1+x+x^2}\,dx \\
&= -\frac{1}{4}(d-f)\log\left(1-x+x^2\right) + \frac{1}{4}(d-f)\log\left(1+x+x^2\right) \\
&\quad - e\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) \\
&\quad + \frac{1}{2}(-d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) \\
&\quad + \frac{1}{2}(-d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) \\
&= -\frac{(d+f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(d+f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{\sqrt{3}} \\
&\quad - \frac{1}{4}(d-f)\log\left(1-x+x^2\right) + \frac{1}{4}(d-f)\log\left(1+x+x^2\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.11 (sec) , antiderivative size = 121, normalized size of antiderivative = 1.16

$$
\begin{aligned}
\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx &= \frac{\left(2id+\left(-i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{6+6i\sqrt{3}}} \\
&\quad + \frac{\left(-2id+\left(i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{6-6i\sqrt{3}}} - \frac{e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)}{\sqrt{3}}
\end{aligned}
$$

```
[In] Integrate[(d + e*x + f*x^2)/(1 + x^2 + x^4),x]
```

```
[Out] (((2*I)*d + (-I + Sqrt[3])*f)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqrt[6 + (6*I)*
Sqrt[3]] + (((-2*I)*d + (I + Sqrt[3])*f)*ArcTan[((I + Sqrt[3])*x)/2])/Sqrt[
6 - (6*I)*Sqrt[3]] - (e*ArcTan[Sqrt[3]/(1 + 2*x^2)])/Sqrt[3]
```

Maple [A] (verified)

Time = 0.23 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.79

method	result	size
default	$\frac{(f-d)\ln(x^2-x+1)}{4}+\frac{\left(\frac{d}{2}+e+\frac{f}{2}\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{3}+\frac{(d-f)\ln(x^2+x+1)}{4}+\frac{\left(\frac{d}{2}-e+\frac{f}{2}\right)\arctan\left(\frac{(1+2x)\sqrt{3}}{3}\right)\sqrt{3}}{3}$	82
risch	Expression too large to display	7878

```
[In] int((f*x^2+e*x+d)/(x^4+x^2+1),x,method=_RETURNVERBOSE)
```

```
[Out] 1/4*(f-d)*ln(x^2-x+1)+1/3*(1/2*d+e+1/2*f)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2
))+1/4*(d-f)*ln(x^2+x+1)+1/3*(1/2*d-e+1/2*f)*arctan(1/3*(1+2*x)*3^(1/2))*3^
(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.72

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx = \frac{1}{6}\,\sqrt{3}(d-2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right) + \frac{1}{6}\,\sqrt{3}(d+2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right) + \frac{1}{4}\,(d-f)\log\left(x^2+x+1\right) - \frac{1}{4}\,(d-f)\log\left(x^2-x+1\right)$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="fricas")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d +
2*e + f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f)*log(x^2 + x + 1) - 1/4
*(d - f)*log(x^2 - x + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 65.87 (sec) , antiderivative size = 3589, normalized size of antiderivative = 34.51

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4+x**2+1),x)
```

```
[Out] (-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)*log(x + (-7*d**5*e + 6*d**5*(-d/4
 + f/4 - sqrt(3)*I*(d + 2*e + f)/12) + 25*d**4*e*f + 18*d**4*f*(-d/4 + f/4
- sqrt(3)*I*(d + 2*e + f)/12) - 15*d**3*e**3 - 18*d**3*e**2*(-d/4 + f/4 - s
qrt(3)*I*(d + 2*e + f)/12) - 25*d**3*e*f**2 + 60*d**3*e*(-d/4 + f/4 - sqrt(
3)*I*(d + 2*e + f)/12)**2 - 42*d**3*f**2*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e +
 f)/12) + 72*d**3*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**3 + 108*d**2*e
**2*f*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12) + 20*d**2*e*f**3 - 144*d**2
*e*f*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**2 - 12*d**2*f**3*(-d/4 + f/
4 - sqrt(3)*I*(d + 2*e + f)/12) - 144*d**2*f*(-d/4 + f/4 - sqrt(3)*I*(d + 2
*e + f)/12)**3 + 4*d*e**5 + 24*d*e**4*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)
/12) + 15*d*e**3*f**2 + 48*d*e**3*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)
**2 - 54*d*e**2*f**2*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12) + 288*d*e**2
*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**3 - 20*d*e*f**4 + 180*d*e*f**2*
(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**2 + 36*d*f**4*(-d/4 + f/4 - sqrt
(3)*I*(d + 2*e + f)/12) - 72*d*f**2*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/1
2)**3 - 8*e**5*f - 96*e**3*f*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**2 +
 36*e**2*f**3*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12) + 11*e*f**5 - 48*e*
f**3*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/12)**2 - 6*f**5*(-d/4 + f/4 - sq
rt(3)*I*(d + 2*e + f)/12) + 144*f**3*(-d/4 + f/4 - sqrt(3)*I*(d + 2*e + f)/
12)**3)/(3*d**6 - 3*d**5*f - 8*d**4*e**2 - 3*d**4*f**2 + 40*d**3*e**2*f + 6
*d**3*f**3 - 16*d**2*e**4 - 48*d**2*e**2*f**2 - 3*d**2*f**4 + 16*d*e**4*f +
 40*d*e**2*f**3 - 3*d*f**5 - 16*e**4*f**2 - 8*e**2*f**4 + 3*f**6)) + (-d/4
+ f/4 + sqrt(3)*I*(d + 2*e + f)/12)*log(x + (-7*d**5*e + 6*d**5*(-d/4 + f/4
 + sqrt(3)*I*(d + 2*e + f)/12) + 25*d**4*e*f + 18*d**4*f*(-d/4 + f/4 + sqrt
(3)*I*(d + 2*e + f)/12) - 15*d**3*e**3 - 18*d**3*e**2*(-d/4 + f/4 + sqrt(3)
*I*(d + 2*e + f)/12) - 25*d**3*e*f**2 + 60*d**3*e*(-d/4 + f/4 + sqrt(3)*I*(
d + 2*e + f)/12)**2 - 42*d**3*f**2*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12
) + 72*d**3*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**3 + 108*d**2*e**2*f*
(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12) + 20*d**2*e*f**3 - 144*d**2*e*f*(
-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**2 - 12*d**2*f**3*(-d/4 + f/4 + sq
rt(3)*I*(d + 2*e + f)/12) - 144*d**2*f*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f
)/12)**3 + 4*d*e**5 + 24*d*e**4*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12) +
 15*d*e**3*f**2 + 48*d*e**3*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**2 -
54*d*e**2*f**2*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12) + 288*d*e**2*(-d/4
 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**3 - 20*d*e*f**4 + 180*d*e*f**2*(-d/4
```

```
+ f/4 + sqrt(3)*I*(d + 2*e + f)/12)**2 + 36*d*f**4*(-d/4 + f/4 + sqrt(3)*I*
(d + 2*e + f)/12) - 72*d*f**2*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**3
- 8*e**5*f - 96*e**3*f*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**2 + 36*e*
*2*f**3*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12) + 11*e*f**5 - 48*e*f**3*(
-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**2 - 6*f**5*(-d/4 + f/4 + sqrt(3)*
I*(d + 2*e + f)/12) + 144*f**3*(-d/4 + f/4 + sqrt(3)*I*(d + 2*e + f)/12)**3
)/(3*d**6 - 3*d**5*f - 8*d**4*e**2 - 3*d**4*f**2 + 40*d**3*e**2*f + 6*d**3*
f**3 - 16*d**2*e**4 - 48*d**2*e**2*f**2 - 3*d**2*f**4 + 16*d*e**4*f + 40*d*
e**2*f**3 - 3*d*f**5 - 16*e**4*f**2 - 8*e**2*f**4 + 3*f**6)) + (d/4 - f/4 -
 sqrt(3)*I*(d - 2*e + f)/12)*log(x + (-7*d**5*e + 6*d**5*(d/4 - f/4 - sqrt(
3)*I*(d - 2*e + f)/12) + 25*d**4*e*f + 18*d**4*f*(d/4 - f/4 - sqrt(3)*I*(d
- 2*e + f)/12) - 15*d**3*e**3 - 18*d**3*e**2*(d/4 - f/4 - sqrt(3)*I*(d - 2*
e + f)/12) - 25*d**3*e*f**2 + 60*d**3*e*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f
)/12)**2 - 42*d**3*f**2*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12) + 72*d**3*
(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**3 + 108*d**2*e**2*f*(d/4 - f/4 -
sqrt(3)*I*(d - 2*e + f)/12) + 20*d**2*e*f**3 - 144*d**2*e*f*(d/4 - f/4 - sq
rt(3)*I*(d - 2*e + f)/12)**2 - 12*d**2*f**3*(d/4 - f/4 - sqrt(3)*I*(d - 2*e
 + f)/12) - 144*d**2*f*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**3 + 4*d*e*
*5 + 24*d*e**4*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12) + 15*d*e**3*f**2 +
48*d*e**3*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**2 - 54*d*e**2*f**2*(d/4
 - f/4 - sqrt(3)*I*(d - 2*e + f)/12) + 288*d*e**2*(d/4 - f/4 - sqrt(3)*I*(d
 - 2*e + f)/12)**3 - 20*d*e*f**4 + 180*d*e*f**2*(d/4 - f/4 - sqrt(3)*I*(d -
 2*e + f)/12)**2 + 36*d*f**4*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12) - 72*
d*f**2*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**3 - 8*e**5*f - 96*e**3*f*(
d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**2 + 36*e**2*f**3*(d/4 - f/4 - sqrt
(3)*I*(d - 2*e + f)/12) + 11*e*f**5 - 48*e*f**3*(d/4 - f/4 - sqrt(3)*I*(d -
 2*e + f)/12)**2 - 6*f**5*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12) + 144*f*
*3*(d/4 - f/4 - sqrt(3)*I*(d - 2*e + f)/12)**3)/(3*d**6 - 3*d**5*f - 8*d**4
*e**2 - 3*d**4*f**2 + 40*d**3*e**2*f + 6*d**3*f**3 - 16*d**2*e**4 - 48*d**2
*e**2*f**2 - 3*d**2*f**4 + 16*d*e**4*f + 40*d*e**2*f**3 - 3*d*f**5 - 16*e**
4*f**2 - 8*e**2*f**4 + 3*f**6)) + (d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)*
log(x + (-7*d**5*e + 6*d**5*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) + 25*d
**4*e*f + 18*d**4*f*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) - 15*d**3*e**3
 - 18*d**3*e**2*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) - 25*d**3*e*f**2 +
 60*d**3*e*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)**2 - 42*d**3*f**2*(d/4
- f/4 + sqrt(3)*I*(d - 2*e + f)/12) + 72*d**3*(d/4 - f/4 + sqrt(3)*I*(d - 2
*e + f)/12)**3 + 108*d**2*e**2*f*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) +
 20*d**2*e*f**3 - 144*d**2*e*f*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)**2
- 12*d**2*f**3*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) - 144*d**2*f*(d/4 -
 f/4 + sqrt(3)*I*(d - 2*e + f)/12)**3 + 4*d*e**5 + 24*d*e**4*(d/4 - f/4 + s
qrt(3)*I*(d - 2*e + f)/12) + 15*d*e**3*f**2 + 48*d*e**3*(d/4 - f/4 + sqrt(3
)*I*(d - 2*e + f)/12)**2 - 54*d*e**2*f**2*(d/4 - f/4 + sqrt(3)*I*(d - 2*e +
 f)/12) + 288*d*e**2*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)**3 - 20*d*e*f
**4 + 180*d*e*f**2*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)**2 + 36*d*f**4*
(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) - 72*d*f**2*(d/4 - f/4 + sqrt(3)*I
```

```
*(d - 2*e + f)/12)**3 - 8*e**5*f - 96*e**3*f*(d/4 - f/4 + sqrt(3)*I*(d - 2*
e + f)/12)**2 + 36*e**2*f**3*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) + 11*
e*f**5 - 48*e*f**3*(d/4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12)**2 - 6*f**5*(d/
4 - f/4 + sqrt(3)*I*(d - 2*e + f)/12) + 144*f**3*(d/4 - f/4 + sqrt(3)*I*(d
- 2*e + f)/12)**3)/(3*d**6 - 3*d**5*f - 8*d**4*e**2 - 3*d**4*f**2 + 40*d**3
*e**2*f + 6*d**3*f**3 - 16*d**2*e**4 - 48*d**2*e**2*f**2 - 3*d**2*f**4 + 16
*d*e**4*f + 40*d*e**2*f**3 - 3*d*f**5 - 16*e**4*f**2 - 8*e**2*f**4 + 3*f**6
))
```

Maxima [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.72

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx = \frac{1}{6}\,\sqrt{3}(d-2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right) + \frac{1}{6}\,\sqrt{3}(d+2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right) + \frac{1}{4}\,(d-f)\log\left(x^2+x+1\right) - \frac{1}{4}\,(d-f)\log\left(x^2-x+1\right)$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="maxima")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d +
2*e + f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f)*log(x^2 + x + 1) - 1/4
*(d - f)*log(x^2 - x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.72

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx = \frac{1}{6}\,\sqrt{3}(d-2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right) + \frac{1}{6}\,\sqrt{3}(d+2\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right) + \frac{1}{4}\,(d-f)\log\left(x^2+x+1\right) - \frac{1}{4}\,(d-f)\log\left(x^2-x+1\right)$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="giac")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(d +
2*e + f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f)*log(x^2 + x + 1) - 1/4
*(d - f)*log(x^2 - x + 1)
```

Mupad [B] (verification not implemented)

Time = 7.98 (sec) , antiderivative size = 159, normalized size of antiderivative = 1.53

$$\int \frac{d+ex+fx^2}{1+x^2+x^4}\,dx = -\ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}\right) - \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{4}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{4}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}\right)$$

```
[In] int((d + e*x + f*x^2)/(x^2 + x^4 + 1),x)
```

```
[Out] log(x + (3^(1/2)*1i)/2 - 1/2)*(f/4 - d/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1
i)/6 + (3^(1/2)*f*1i)/12) - log(x - (3^(1/2)*1i)/2 + 1/2)*(f/4 - d/4 + (3^(
1/2)*d*1i)/12 - (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12) - log(x - (3^(1/2)*1i
)/2 - 1/2)*(d/4 - f/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1
i)/12) + log(x + (3^(1/2)*1i)/2 + 1/2)*(d/4 - f/4 + (3^(1/2)*d*1i)/12 - (3^
(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12)
```

3.17 $\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4} \, dx$

Optimal result

Integrand size = 26, antiderivative size = 127

$$\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4} \, dx = -\frac{(d+f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(d+f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(1-x+x^2\right) + \frac{1}{4}(d-f)\log\left(1+x+x^2\right) + \frac{1}{4}g\log\left(1+x^2+x^4\right)$$

```
[Out] -1/4*(d-f)*ln(x^2-x+1)+1/4*(d-f)*ln(x^2+x+1)+1/4*g*ln(x^4+x^2+1)-1/6*(d+f)*
arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/6*(d+f)*arctan(1/3*(1+2*x)*3^(1/2))*3
^(1/2)+1/6*(2*e-g)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.269, Rules used = {1687, 1183, 648, 632, 210, 642, 1261}

$$\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4} \, dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(d+f)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(d+f)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(x^2-x+1\right) + \frac{1}{4}(d-f)\log\left(x^2+x+1\right) + \frac{1}{4}g\log\left(x^4+x^2+1\right)$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4),x]
```

```
[Out] -1/2*((d + f)*ArcTan[(1 - 2*x)/Sqrt[3]])/Sqrt[3] + ((d + f)*ArcTan[(1 + 2*x
)/Sqrt[3]])/(2*Sqrt[3]) + ((2*e - g)*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(2*Sqrt[3
]) - ((d - f)*Log[1 - x + x^2])/4 + ((d - f)*Log[1 + x + x^2])/4 + (g*Log[1
 + x^2 + x^4])/4
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
```

```
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{1+x^2+x^4}\,dx+\int \frac{x(e+gx^2)}{1+x^2+x^4}\,dx\\
&= \frac{1}{2}\int \frac{d-(d-f)x}{1-x+x^2}\,dx+\frac{1}{2}\int \frac{d+(d-f)x}{1+x+x^2}\,dx+\frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{1+x+x^2}\,dx,x,x^2\right)\\
&= \frac{1}{4}(d-f)\int \frac{1+2x}{1+x+x^2}\,dx+\frac{1}{4}(-d+f)\int \frac{-1+2x}{1-x+x^2}\,dx\\
&\quad+\frac{1}{4}(d+f)\int \frac{1}{1-x+x^2}\,dx+\frac{1}{4}(d+f)\int \frac{1}{1+x+x^2}\,dx\\
&\quad+\frac{1}{4}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx,x,x^2\right)+\frac{1}{4}g\text{Subst}\left(\int \frac{1+2x}{1+x+x^2}\,dx,x,x^2\right)\\
&= -\frac{1}{4}(d-f)\log\left(1-x+x^2\right)+\frac{1}{4}(d-f)\log\left(1+x+x^2\right)\\
&\quad+\frac{1}{4}g\log\left(1+x^2+x^4\right)+\frac{1}{2}(-d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx,x,-1+2x\right)\\
&\quad+\frac{1}{2}(-d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx,x,1+2x\right)\\
&\quad+\frac{1}{2}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx,x,1+2x^2\right)\\
&= -\frac{(d+f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(d+f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}}\\
&\quad-\frac{1}{4}(d-f)\log\left(1-x+x^2\right)+\frac{1}{4}(d-f)\log\left(1+x+x^2\right)+\frac{1}{4}g\log\left(1+x^2+x^4\right)
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.36 (sec) , antiderivative size = 150, normalized size of antiderivative = 1.18

$$\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx$$
$$=\frac{2\sqrt{2-2i\sqrt{3}}\left(2id+\left(-i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)+2\left(\sqrt{2+2i\sqrt{3}}\left(-2id+\left(i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\right.\right.}{8\sqrt{3}}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4),x]
```

```
[Out] (2*Sqrt[2 - (2*I)*Sqrt[3]]*((2*I)*d + (-I + Sqrt[3])*f)*ArcTan[((-I + Sqrt[
3])*x)/2] + 2*(Sqrt[2 + (2*I)*Sqrt[3]]*((-2*I)*d + (I + Sqrt[3])*f)*ArcTan[
((I + Sqrt[3])*x)/2] + (-4*e + 2*g)*ArcTan[Sqrt[3]/(1 + 2*x^2)] + Sqrt[3]*g
*Log[1 + x^2 + x^4]))/(8*Sqrt[3])
```

Maple [A] (verified)

Time = 0.27 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.71

method	result
default	$\frac{(f-d+g)\ln(x^2-x+1)}{4}+\frac{\left(\frac{d}{2}+e+\frac{f}{2}-\frac{g}{2}\right)\sqrt{3}\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{3}+\frac{(d-f+g)\ln(x^2+x+1)}{4}+\frac{\left(\frac{d}{2}-e+\frac{f}{2}+\frac{g}{2}\right)\arctan\left(\frac{(1+2x)\sqrt{3}}{3}\right)\sqrt{3}}{3}$
risch	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x,method=_RETURNVERBOSE)
```

```
[Out] 1/4*(f-d+g)*ln(x^2-x+1)+1/3*(1/2*d+e+1/2*f-1/2*g)*3^(1/2)*arctan(1/3*(2*x-1
)*3^(1/2))+1/4*(d-f+g)*ln(x^2+x+1)+1/3*(1/2*d-e+1/2*f+1/2*g)*arctan(1/3*(1+
2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.41 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.65

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx &= \frac{1}{6}\sqrt{3}(d-2\,e+f+g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)\\ &\quad+\frac{1}{6}\sqrt{3}(d+2\,e+f-g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)\\ &\quad+\frac{1}{4}\,(d-f+g)\log\left(x^2+x+1\right)-\frac{1}{4}\,(d-f-g)\log\left(x^2-x+1\right)\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="fricas")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f + g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(
d + 2*e + f - g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f + g)*log(x^2 +
x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4+x**2+1),x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.65

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx &= \frac{1}{6}\sqrt{3}(d-2e+f+g)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right)\\ &+\frac{1}{6}\sqrt{3}(d+2e+f-g)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right)\\ &+\frac{1}{4}(d-f+g)\log\left(x^2+x+1\right)-\frac{1}{4}(d-f-g)\log\left(x^2-x+1\right)\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="maxima")
[Out] 1/6*sqrt(3)*(d - 2*e + f + g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(
d + 2*e + f - g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f + g)*log(x^2 +
x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.65

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx &= \frac{1}{6}\sqrt{3}(d-2e+f+g)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right)\\ &+\frac{1}{6}\sqrt{3}(d+2e+f-g)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right)\\ &+\frac{1}{4}(d-f+g)\log\left(x^2+x+1\right)-\frac{1}{4}(d-f-g)\log\left(x^2-x+1\right)\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="giac")
[Out] 1/6*sqrt(3)*(d - 2*e + f + g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqrt(3)*(
d + 2*e + f - g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*(d - f + g)*log(x^2 +
x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Mupad [B] (verification not implemented)

Time = 8.11 (sec) , antiderivative size = 199, normalized size of antiderivative = 1.57

$$\int \frac{d+ex+fx^2+gx^3}{1+x^2+x^4}\,dx = -\ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{4}-\frac{g}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{12}\right)-\ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{4}-\frac{d}{4}-\frac{g}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{12}\right)+\ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{4}-\frac{d}{4}+\frac{g}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{12}\right)+\ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{4}+\frac{g}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{12}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{6}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{12}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{12}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^2 + x^4 + 1),x)
```

```
[Out] log(x + (3^(1/2)*1i)/2 - 1/2)*(f/4 - d/4 + g/4 + (3^(1/2)*d*1i)/12 + (3^(1/
2)*e*1i)/6 + (3^(1/2)*f*1i)/12 - (3^(1/2)*g*1i)/12) - log(x - (3^(1/2)*1i)/
2 + 1/2)*(f/4 - d/4 - g/4 + (3^(1/2)*d*1i)/12 - (3^(1/2)*e*1i)/6 + (3^(1/2)
*f*1i)/12 + (3^(1/2)*g*1i)/12) - log(x - (3^(1/2)*1i)/2 - 1/2)*(d/4 - f/4 -
 g/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 - (3^(1/2)*
g*1i)/12) + log(x + (3^(1/2)*1i)/2 + 1/2)*(d/4 - f/4 + g/4 + (3^(1/2)*d*1i)
/12 - (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 + (3^(1/2)*g*1i)/12)
```

3.18 $\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 136

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = hx-\frac{(d+f-2h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(d+f-2h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}}-\frac{1}{4}(d-f)\log\left(1-x+x^2\right)+\frac{1}{4}(d-f)\log\left(1+x+x^2\right)+\frac{1}{4}g\log\left(1+x^2+x^4\right)$$

```
[Out] h*x-1/4*(d-f)*ln(x^2-x+1)+1/4*(d-f)*ln(x^2+x+1)+1/4*g*ln(x^4+x^2+1)-1/6*(d+
f-2*h)*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/6*(d+f-2*h)*arctan(1/3*(1+2*x)
*3^(1/2))*3^(1/2)+1/6*(2*e-g)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 136, normalized size of antiderivative = 1.00, number of steps used = 17, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.258, Rules used

$= \{1687, 1690, 1183, 648, 632, 210, 642, 1261\}$

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(d+f-2h)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(d+f-2h)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(x^2-x+1\right) + \frac{1}{4}(d-f)\log\left(x^2+x+1\right) + \frac{1}{4}g\log\left(x^4+x^2+1\right) + hx$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4),x]
```

```
[Out] h*x - ((d + f - 2*h)*ArcTan[(1 - 2*x)/Sqrt[3]])/(2*Sqrt[3]) + ((d + f - 2*h
)*ArcTan[(1 + 2*x)/Sqrt[3]])/(2*Sqrt[3]) + ((2*e - g)*ArcTan[(1 + 2*x^2)/Sq
rt[3]])/(2*Sqrt[3]) - ((d - f)*Log[1 - x + x^2])/4 + ((d - f)*Log[1 + x + x
^2])/4 + (g*Log[1 + x^2 + x^4])/4
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{1+x^2+x^4}\,dx+\int \frac{d+fx^2+hx^4}{1+x^2+x^4}\,dx\\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{1+x+x^2}\,dx,x,x^2\right)+\int \left(h+\frac{d-h+(f-h)x^2}{1+x^2+x^4}\right)\,dx\\
&= hx+\frac{1}{4}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx,x,x^2\right)\\
&\quad+\frac{1}{4}g\text{Subst}\left(\int \frac{1+2x}{1+x+x^2}\,dx,x,x^2\right)+\int \frac{d-h+(f-h)x^2}{1+x^2+x^4}\,dx\\
&= hx+\frac{1}{4}g\log\left(1+x^2+x^4\right)+\frac{1}{2}\int \frac{d-h-(d-f)x}{1-x+x^2}\,dx\\
&\quad+\frac{1}{2}\int \frac{d-h+(d-f)x}{1+x+x^2}\,dx+\frac{1}{2}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx,x,1+2x^2\right)
\end{aligned}
$$

$$= hx + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{1}{4}g\log\left(1+x^2+x^4\right) + \frac{1}{4}(d-f)\int\frac{1+2x}{1+x+x^2}\,dx + \frac{1}{4}(-d+f)\int\frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{4}(d+f-2h)\int\frac{1}{1-x+x^2}\,dx + \frac{1}{4}(d+f-2h)\int\frac{1}{1+x+x^2}\,dx$$

$$= hx + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(1-x+x^2\right) + \frac{1}{4}(d-f)\log\left(1+x+x^2\right) + \frac{1}{4}g\log\left(1+x^2+x^4\right) + \frac{1}{2}(-d-f+2h)\mathrm{Subst}\left(\int\frac{1}{-3-x^2}\,dx, x, -1+2x\right) + \frac{1}{2}(-d-f+2h)\mathrm{Subst}\left(\int\frac{1}{-3-x^2}\,dx, x, 1+2x\right)$$

$$= hx - \frac{(d+f-2h)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(d+f-2h)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(1-x+x^2\right) + \frac{1}{4}(d-f)\log\left(1+x+x^2\right) + \frac{1}{4}g\log\left(1+x^2+x^4\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.47 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.21

$$\int\frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = \frac{1}{24}\Bigg(24hx + 4\left(\left(3i+\sqrt{3}\right)d + \left(-3i+\sqrt{3}\right)f - 2\sqrt{3}h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right) + 4\left(\left(-3i+\sqrt{3}\right)d + \left(3i+\sqrt{3}\right)f - 2\sqrt{3}h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right) - 8\sqrt{3}e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right) + 4\sqrt{3}g\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right) + 6g\log\left(1+x^2+x^4\right)\Bigg)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4),x]
```

```
[Out] (24*h*x + 4*((3*I + Sqrt[3])*d + (-3*I + Sqrt[3])*f - 2*Sqrt[3]*h)*ArcTan[(
(-I + Sqrt[3])*x)/2] + 4*((-3*I + Sqrt[3])*d + (3*I + Sqrt[3])*f - 2*Sqrt[3
]*h)*ArcTan[((I + Sqrt[3])*x)/2] - 8*Sqrt[3]*e*ArcTan[Sqrt[3]/(1 + 2*x^2)]
+ 4*Sqrt[3]*g*ArcTan[Sqrt[3]/(1 + 2*x^2)] + 6*g*Log[1 + x^2 + x^4])/24
```

Maple [A] (verified)

Time = 0.39 (sec) , antiderivative size = 99, normalized size of antiderivative = 0.73

method	result
default	$hx+\frac{(f-d+g)\ln(x^2-x+1)}{4}+\frac{\left(\frac{d}{2}+e+\frac{f}{2}-\frac{g}{2}-h\right)\sqrt{3}\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{3}+\frac{(d-f+g)\ln(x^2+x+1)}{4}+\frac{\left(\frac{d}{2}-e+\frac{f}{2}+\frac{g}{2}-h\right)\arctan\left(\right.}{3}$
risch	Expression too large to display

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x,method=_RETURNVERBOSE)
```

```
[Out] h*x+1/4*(f-d+g)*ln(x^2-x+1)+1/3*(1/2*d+e+1/2*f-1/2*g-h)*3^(1/2)*arctan(1/3*
(2*x-1)*3^(1/2))+1/4*(d-f+g)*ln(x^2+x+1)+1/3*(1/2*d-e+1/2*f+1/2*g-h)*arctan
(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 1.12 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.68

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx=\frac{1}{6}\sqrt{3}(d-2e+f+g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right)+\frac{1}{6}\sqrt{3}(d+2e+f-g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right)+hx+\frac{1}{4}(d-f+g)\log\left(x^2+x+1\right)-\frac{1}{4}(d-f-g)\log\left(x^2-x+1\right)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="fricas")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f + g - 2*h)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqr
t(3)*(d + 2*e + f - g - 2*h)*arctan(1/3*sqrt(3)*(2*x - 1)) + h*x + 1/4*(d -
 f + g)*log(x^2 + x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx=\text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1),x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.68

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = \frac{1}{6}\sqrt{3}(d-2e+f+g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{6}\sqrt{3}(d+2e+f-g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + hx + \frac{1}{4}(d-f+g)\log\left(x^2+x+1\right) - \frac{1}{4}(d-f-g)\log\left(x^2-x+1\right)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="maxima")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f + g - 2*h)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqr
t(3)*(d + 2*e + f - g - 2*h)*arctan(1/3*sqrt(3)*(2*x - 1)) + h*x + 1/4*(d -
 f + g)*log(x^2 + x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.68

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = \frac{1}{6}\sqrt{3}(d-2e+f+g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{6}\sqrt{3}(d+2e+f-g-2h)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + hx + \frac{1}{4}(d-f+g)\log\left(x^2+x+1\right) - \frac{1}{4}(d-f-g)\log\left(x^2-x+1\right)$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="giac")
```

```
[Out] 1/6*sqrt(3)*(d - 2*e + f + g - 2*h)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/6*sqr
t(3)*(d + 2*e + f - g - 2*h)*arctan(1/3*sqrt(3)*(2*x - 1)) + h*x + 1/4*(d -
 f + g)*log(x^2 + x + 1) - 1/4*(d - f - g)*log(x^2 - x + 1)
```

Mupad [B] (verification not implemented)

Time = 11.14 (sec) , antiderivative size = 1209, normalized size of antiderivative = 8.89

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{1+x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^2 + x^4 + 1),x)
```

```
[Out] log(d*f*9i - d*e*6i + d*g*3i - d*h*3i + e*h*6i + f*h*3i - g*h*3i - 3*3^(1/2
)*d^2 - d^2*x*6i - f^2*x*3i - d^2*3i - f^2*6i + 2*3^(1/2)*d*e + 3*3^(1/2)*d
*f - 3^(1/2)*d*g - 4*3^(1/2)*e*f + 3*3^(1/2)*d*h + 2*3^(1/2)*e*h + 2*3^(1/2
)*f*g - 3*3^(1/2)*f*h - 3^(1/2)*g*h + d*f*x*9i - e*f*x*6i + d*h*x*3i + e*h*
x*6i + f*g*x*3i - f*h*x*3i - g*h*x*3i + 3*3^(1/2)*f^2*x - 3*3^(1/2)*d*f*x -
 2*3^(1/2)*d*g*x - 2*3^(1/2)*e*f*x + 3*3^(1/2)*d*h*x - 2*3^(1/2)*e*h*x + 3^
(1/2)*f*g*x - 3*3^(1/2)*f*h*x + 3^(1/2)*g*h*x + 4*3^(1/2)*d*e*x)*(d/4 - f/4
 + g/4 - (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6 - (3^(1/2)*f*1i)/12 - (3^(1/2
)*g*1i)/12 + (3^(1/2)*h*1i)/6) - log(d*g*3i - d*f*9i - d*e*6i + d*h*3i + e*
h*6i - f*h*3i - g*h*3i - 3*3^(1/2)*d^2 - d^2*x*6i - f^2*x*3i + d^2*3i + f^2
*6i - 2*3^(1/2)*d*e + 3*3^(1/2)*d*f + 3^(1/2)*d*g + 4*3^(1/2)*e*f + 3*3^(1/
2)*d*h - 2*3^(1/2)*e*h - 2*3^(1/2)*f*g - 3*3^(1/2)*f*h + 3^(1/2)*g*h + d*f*
x*9i + e*f*x*6i + d*h*x*3i - e*h*x*6i - f*g*x*3i - f*h*x*3i + g*h*x*3i - 3*
3^(1/2)*f^2*x + 3*3^(1/2)*d*f*x - 2*3^(1/2)*d*g*x - 2*3^(1/2)*e*f*x - 3*3^(
1/2)*d*h*x - 2*3^(1/2)*e*h*x + 3^(1/2)*f*g*x + 3*3^(1/2)*f*h*x + 3^(1/2)*g*
h*x + 4*3^(1/2)*d*e*x)*(d/4 - f/4 - g/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i
)/6 + (3^(1/2)*f*1i)/12 - (3^(1/2)*g*1i)/12 - (3^(1/2)*h*1i)/6) + log(d*f*9
i - d*e*6i + d*g*3i - d*h*3i + e*h*6i + f*h*3i - g*h*3i + 3*3^(1/2)*d^2 - d
^2*x*6i - f^2*x*3i - d^2*3i - f^2*6i - 2*3^(1/2)*d*e - 3*3^(1/2)*d*f + 3^(1
/2)*d*g + 4*3^(1/2)*e*f - 3*3^(1/2)*d*h - 2*3^(1/2)*e*h - 2*3^(1/2)*f*g + 3
*3^(1/2)*f*h + 3^(1/2)*g*h + d*f*x*9i - e*f*x*6i + d*h*x*3i + e*h*x*6i + f*
g*x*3i - f*h*x*3i - g*h*x*3i - 3*3^(1/2)*f^2*x + 3*3^(1/2)*d*f*x + 2*3^(1/2
)*d*g*x + 2*3^(1/2)*e*f*x - 3*3^(1/2)*d*h*x + 2*3^(1/2)*e*h*x - 3^(1/2)*f*g
*x + 3*3^(1/2)*f*h*x - 3^(1/2)*g*h*x - 4*3^(1/2)*d*e*x)*(d/4 - f/4 + g/4 +
(3^(1/2)*d*1i)/12 - (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 + (3^(1/2)*g*1i)/1
2 - (3^(1/2)*h*1i)/6) + log(d*g*3i - d*f*9i - d*e*6i + d*h*3i + e*h*6i - f*
h*3i - g*h*3i + 3*3^(1/2)*d^2 - d^2*x*6i - f^2*x*3i + d^2*3i + f^2*6i + 2*3
^(1/2)*d*e - 3*3^(1/2)*d*f - 3^(1/2)*d*g - 4*3^(1/2)*e*f - 3*3^(1/2)*d*h +
2*3^(1/2)*e*h + 2*3^(1/2)*f*g + 3*3^(1/2)*f*h - 3^(1/2)*g*h + d*f*x*9i + e*
f*x*6i + d*h*x*3i - e*h*x*6i - f*g*x*3i - f*h*x*3i + g*h*x*3i + 3*3^(1/2)*f
^2*x - 3*3^(1/2)*d*f*x + 2*3^(1/2)*d*g*x + 2*3^(1/2)*e*f*x + 3*3^(1/2)*d*h*
x + 2*3^(1/2)*e*h*x - 3^(1/2)*f*g*x - 3*3^(1/2)*f*h*x - 3^(1/2)*g*h*x - 4*3
^(1/2)*d*e*x)*(f/4 - d/4 + g/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6 + (3^
(1/2)*f*1i)/12 - (3^(1/2)*g*1i)/12 - (3^(1/2)*h*1i)/6) + h*x
```

3.19 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 151

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = hx+\frac{ix^2}{2}-\frac{(d+f-2h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(d+f-2h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{2\sqrt{3}}+\frac{(2e-g-i)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{2\sqrt{3}}-\frac{1}{4}(d-f)\log\left(1-x+x^2\right)+\frac{1}{4}(d-f)\log\left(1+x+x^2\right)+\frac{1}{4}(g-i)\log\left(1+x^2+x^4\right)$$

```
[Out] h*x+1/2*i*x^2-1/4*(d-f)*ln(x^2-x+1)+1/4*(d-f)*ln(x^2+x+1)+1/4*(g-i)*ln(x^4+
x^2+1)-1/6*(d+f-2*h)*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/6*(d+f-2*h)*arct
an(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/6*(2*e-g-i)*arctan(1/3*(2*x^2+1)*3^(1/2))
*3^(1/2)
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 151, normalized size of antiderivative = 1.00, number of steps used = 19, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.250$, Rules used = {1687, 1690, 1183, 648, 632, 210, 642, 1677, 1671}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(d+f-2h)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(d+f-2h)}{2\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g-i)}{2\sqrt{3}} - \frac{1}{4}(d-f)\log\left(x^2-x+1\right)+\frac{1}{4}(d-f)\log\left(x^2+x+1\right) + \frac{1}{4}(g-i)\log\left(x^4+x^2+1\right)+hx+\frac{ix^2}{2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4),x]
```

```
[Out] h*x + (i*x^2)/2 - ((d + f - 2*h)*ArcTan[(1 - 2*x)/Sqrt[3]])/(2*Sqrt[3]) + (
(d + f - 2*h)*ArcTan[(1 + 2*x)/Sqrt[3]])/(2*Sqrt[3]) + ((2*e - g - i)*ArcTa
n[(1 + 2*x^2)/Sqrt[3]])/(2*Sqrt[3]) - ((d - f)*Log[1 - x + x^2])/4 + ((d -
f)*Log[1 + x + x^2])/4 + ((g - i)*Log[1 + x^2 + x^4])/4
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
```

```
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+fx^2+hx^4}{1+x^2+x^4}\,dx+\int \frac{x(e+gx^2+ix^4)}{1+x^2+x^4}\,dx\\ &= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{1+x+x^2}\,dx,x,x^2\right)+\int \left(h+\frac{d-h+(f-h)x^2}{1+x^2+x^4}\right)\,dx\end{aligned}$$

$$
\begin{aligned}
&= hx + \frac{1}{2}\text{Subst}\left(\int \left(i + \frac{e - i + (g - i)x}{1 + x + x^2}\right) dx, x, x^2\right) + \int \frac{d - h + (f - h)x^2}{1 + x^2 + x^4} dx \\
&= hx + \frac{ix^2}{2} + \frac{1}{2}\int \frac{d - h - (d - f)x}{1 - x + x^2} dx + \frac{1}{2}\int \frac{d - h + (d - f)x}{1 + x + x^2} dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e - i + (g - i)x}{1 + x + x^2} dx, x, x^2\right) \\
&= hx + \frac{ix^2}{2} + \frac{1}{4}(d - f)\int \frac{1 + 2x}{1 + x + x^2} dx + \frac{1}{4}(-d + f)\int \frac{-1 + 2x}{1 - x + x^2} dx \\
&\quad + \frac{1}{4}(d + f - 2h)\int \frac{1}{1 - x + x^2} dx + \frac{1}{4}(d + f - 2h)\int \frac{1}{1 + x + x^2} dx \\
&\quad + \frac{1}{4}(2e - g - i)\text{Subst}\left(\int \frac{1}{1 + x + x^2} dx, x, x^2\right) \\
&\quad + \frac{1}{4}(g - i)\text{Subst}\left(\int \frac{1 + 2x}{1 + x + x^2} dx, x, x^2\right) \\
&= hx + \frac{ix^2}{2} - \frac{1}{4}(d - f)\log\left(1 - x + x^2\right) + \frac{1}{4}(d - f)\log\left(1 + x + x^2\right) \\
&\quad + \frac{1}{4}(g - i)\log\left(1 + x^2 + x^4\right) + \frac{1}{2}(-d - f + 2h)\text{Subst}\left(\int \frac{1}{-3 - x^2} dx, x, -1 + 2x\right) \\
&\quad + \frac{1}{2}(-d - f + 2h)\text{Subst}\left(\int \frac{1}{-3 - x^2} dx, x, 1 + 2x\right) \\
&\quad + \frac{1}{2}(-2e + g + i)\text{Subst}\left(\int \frac{1}{-3 - x^2} dx, x, 1 + 2x^2\right) \\
&= hx + \frac{ix^2}{2} - \frac{(d + f - 2h)\tan^{-1}\left(\frac{1 - 2x}{\sqrt{3}}\right)}{2\sqrt{3}} + \frac{(d + f - 2h)\tan^{-1}\left(\frac{1 + 2x}{\sqrt{3}}\right)}{2\sqrt{3}} \\
&\quad + \frac{(2e - g - i)\tan^{-1}\left(\frac{1 + 2x^2}{\sqrt{3}}\right)}{2\sqrt{3}} - \frac{1}{4}(d - f)\log\left(1 - x + x^2\right) \\
&\quad + \frac{1}{4}(d - f)\log\left(1 + x + x^2\right) + \frac{1}{4}(g - i)\log\left(1 + x^2 + x^4\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.47 (sec) , antiderivative size = 187, normalized size of antiderivative = 1.24

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = \frac{1}{12}\Bigg(6x(2h+ix)+\left(1+i\sqrt{3}\right)\left(2\sqrt{3}d-\left(3i+\sqrt{3}\right)f-\left(-3i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)+\left(i+\sqrt{3}\right)\left(-2i\sqrt{3}d+\left(3+i\sqrt{3}\right)f+i\left(3i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)-2\sqrt{3}(2e-g-i)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)+3(g-i)\log\left(1+x^2+x^4\right)\Bigg)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4),x]
```

```
[Out] (6*x*(2*h + i*x) + (1 + I*Sqrt[3])*(2*Sqrt[3]*d - (3*I + Sqrt[3])*f - (-3*I
 + Sqrt[3])*h)*ArcTan[((-I + Sqrt[3])*x)/2] + (I + Sqrt[3])*((-2*I)*Sqrt[3]
*d + (3 + I*Sqrt[3])*f + I*(3*I + Sqrt[3])*h)*ArcTan[((I + Sqrt[3])*x)/2] -
 2*Sqrt[3]*(2*e - g - i)*ArcTan[Sqrt[3]/(1 + 2*x^2)] + 3*(g - i)*Log[1 + x^
2 + x^4])/12
```

Maple [A] (verified)

Time = 0.46 (sec) , antiderivative size = 117, normalized size of antiderivative = 0.77

method	result
default	$\frac{i\,x^2}{2}+hx+\frac{(g-i+f-d)\ln\left(x^2-x+1\right)}{4}+\frac{\left(\frac{d}{2}+e+\frac{f}{2}-\frac{g}{2}-h-\frac{i}{2}\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{3}+\frac{(d-f+g-i)\ln\left(x^2+x+1\right)}{4}+\frac{\left(\frac{d}{2}-e+\right.}{}$
risch	Expression too large to display

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x,method=_RETURNVERBOSE)
```

```
[Out] 1/2*i*x^2+h*x+1/4*(g-i+f-d)*ln(x^2-x+1)+1/3*(1/2*d+e+1/2*f-1/2*g-h-1/2*i)*3
^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*(d-f+g-i)*ln(x^2+x+1)+1/3*(1/2*d-e+1
/2*f+1/2*g-h+1/2*i)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 4.39 (sec) , antiderivative size = 106, normalized size of antiderivative = 0.70

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = \frac{1}{2}\,ix^2+\frac{1}{6}\,\sqrt{3}(d-2\,e+f+g-2\,h+i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right)+\frac{1}{6}\,\sqrt{3}(d+2\,e+f-g-2\,h-i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right)+hx+\frac{1}{4}\,(d-f+g-i)\log\left(x^2+x+1\right)-\frac{1}{4}\,(d-f-g+i)\log\left(x^2-x+1\right)$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="fricas"
)
```

```
[Out] 1/2*i*x^2 + 1/6*sqrt(3)*(d - 2*e + f + g - 2*h + i)*arctan(1/3*sqrt(3)*(2*x
 + 1)) + 1/6*sqrt(3)*(d + 2*e + f - g - 2*h - i)*arctan(1/3*sqrt(3)*(2*x -
1)) + h*x + 1/4*(d - f + g - i)*log(x^2 + x + 1) - 1/4*(d - f - g + i)*log(
x^2 - x + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1),x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 106, normalized size of antiderivative = 0.70

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = \frac{1}{2}\,ix^2+\frac{1}{6}\,\sqrt{3}(d-2\,e+f+g-2\,h+i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right)+\frac{1}{6}\,\sqrt{3}(d+2\,e+f-g-2\,h-i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right)+hx+\frac{1}{4}\,(d-f+g-i)\log\left(x^2+x+1\right)-\frac{1}{4}\,(d-f-g+i)\log\left(x^2-x+1\right)$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="maxima"
)
```

```
[Out] 1/2*i*x^2 + 1/6*sqrt(3)*(d - 2*e + f + g - 2*h + i)*arctan(1/3*sqrt(3)*(2*x
 + 1)) + 1/6*sqrt(3)*(d + 2*e + f - g - 2*h - i)*arctan(1/3*sqrt(3)*(2*x -
1)) + h*x + 1/4*(d - f + g - i)*log(x^2 + x + 1) - 1/4*(d - f - g + i)*log(
x^2 - x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 106, normalized size of antiderivative = 0.70

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx &= \frac{1}{2}\,ix^2+\frac{1}{6}\,\sqrt{3}(d-2\,e+f+g-2\,h+i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right)\\&\quad+\frac{1}{6}\,\sqrt{3}(d+2\,e+f-g-2\,h-i)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right)+hx\\&\quad+\frac{1}{4}\,(d-f+g-i)\log\left(x^2+x+1\right)-\frac{1}{4}\,(d-f-g+i)\log\left(x^2-x+1\right)\end{aligned}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1),x, algorithm="giac")
```

```
[Out] 1/2*i*x^2 + 1/6*sqrt(3)*(d - 2*e + f + g - 2*h + i)*arctan(1/3*sqrt(3)*(2*x
 + 1)) + 1/6*sqrt(3)*(d + 2*e + f - g - 2*h - i)*arctan(1/3*sqrt(3)*(2*x -
1)) + h*x + 1/4*(d - f + g - i)*log(x^2 + x + 1) - 1/4*(d - f - g + i)*log(
x^2 - x + 1)
```

Mupad [B] (verification not implemented)

Time = 12.66 (sec) , antiderivative size = 1509, normalized size of antiderivative = 9.99

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{1+x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^2 + x^4 + 1),x)
```

```
[Out] h*x - log(d*g*3i - d*f*9i - d*e*6i + d*h*3i + d*i*3i + e*h*6i - f*h*3i - g*
h*3i - h*i*3i - 3*3^(1/2)*d^2 - d^2*x*6i - f^2*x*3i + d^2*3i + f^2*6i - 2*3
^(1/2)*d*e + 3*3^(1/2)*d*f + 3^(1/2)*d*g + 4*3^(1/2)*e*f + 3*3^(1/2)*d*h +
3^(1/2)*d*i - 2*3^(1/2)*e*h - 2*3^(1/2)*f*g - 3*3^(1/2)*f*h - 2*3^(1/2)*f*i
 + 3^(1/2)*g*h + 3^(1/2)*h*i + d*f*x*9i + e*f*x*6i + d*h*x*3i - e*h*x*6i -
f*g*x*3i - f*h*x*3i - f*i*x*3i + g*h*x*3i + h*i*x*3i - 3*3^(1/2)*f^2*x + 3*
3^(1/2)*d*f*x - 2*3^(1/2)*d*g*x - 2*3^(1/2)*e*f*x - 3*3^(1/2)*d*h*x - 2*3^(
```

```
1/2)*d*i*x - 2*3^(1/2)*e*h*x + 3^(1/2)*f*g*x + 3*3^(1/2)*f*h*x + 3^(1/2)*f*
i*x + 3^(1/2)*g*h*x + 3^(1/2)*h*i*x + 4*3^(1/2)*d*e*x)*(d/4 - f/4 - g/4 + i
/4 + (3^(1/2)*d*1i)/12 + (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 - (3^(1/2)*g*
1i)/12 - (3^(1/2)*h*1i)/6 - (3^(1/2)*i*1i)/12) - log(d*e*6i + d*f*9i - d*g*
3i - d*h*3i - d*i*3i - e*h*6i + f*h*3i + g*h*3i + h*i*3i - 3*3^(1/2)*d^2 +
d^2*x*6i + f^2*x*3i - d^2*3i - f^2*6i - 2*3^(1/2)*d*e + 3*3^(1/2)*d*f + 3^(
1/2)*d*g + 4*3^(1/2)*e*f + 3*3^(1/2)*d*h + 3^(1/2)*d*i - 2*3^(1/2)*e*h - 2*
3^(1/2)*f*g - 3*3^(1/2)*f*h - 2*3^(1/2)*f*i + 3^(1/2)*g*h + 3^(1/2)*h*i - d
*f*x*9i - e*f*x*6i - d*h*x*3i + e*h*x*6i + f*g*x*3i + f*h*x*3i + f*i*x*3i -
 g*h*x*3i - h*i*x*3i - 3*3^(1/2)*f^2*x + 3*3^(1/2)*d*f*x - 2*3^(1/2)*d*g*x
- 2*3^(1/2)*e*f*x - 3*3^(1/2)*d*h*x - 2*3^(1/2)*d*i*x - 2*3^(1/2)*e*h*x + 3
^(1/2)*f*g*x + 3*3^(1/2)*f*h*x + 3^(1/2)*f*i*x + 3^(1/2)*g*h*x + 3^(1/2)*h*
i*x + 4*3^(1/2)*d*e*x)*(d/4 - f/4 - g/4 + i/4 - (3^(1/2)*d*1i)/12 - (3^(1/2
)*e*1i)/6 - (3^(1/2)*f*1i)/12 + (3^(1/2)*g*1i)/12 + (3^(1/2)*h*1i)/6 + (3^(
1/2)*i*1i)/12) - log(d*f*9i - d*e*6i + d*g*3i - d*h*3i + d*i*3i + e*h*6i +
f*h*3i - g*h*3i - h*i*3i - 3*3^(1/2)*d^2 - d^2*x*6i - f^2*x*3i - d^2*3i - f
^2*6i + 2*3^(1/2)*d*e + 3*3^(1/2)*d*f - 3^(1/2)*d*g - 4*3^(1/2)*e*f + 3*3^(
1/2)*d*h - 3^(1/2)*d*i + 2*3^(1/2)*e*h + 2*3^(1/2)*f*g - 3*3^(1/2)*f*h + 2*
3^(1/2)*f*i - 3^(1/2)*g*h - 3^(1/2)*h*i + d*f*x*9i - e*f*x*6i + d*h*x*3i +
e*h*x*6i + f*g*x*3i - f*h*x*3i + f*i*x*3i - g*h*x*3i - h*i*x*3i + 3*3^(1/2)
*f^2*x - 3*3^(1/2)*d*f*x - 2*3^(1/2)*d*g*x - 2*3^(1/2)*e*f*x + 3*3^(1/2)*d*
h*x - 2*3^(1/2)*d*i*x - 2*3^(1/2)*e*h*x + 3^(1/2)*f*g*x - 3*3^(1/2)*f*h*x +
 3^(1/2)*f*i*x + 3^(1/2)*g*h*x + 3^(1/2)*h*i*x + 4*3^(1/2)*d*e*x)*(f/4 - d/
4 - g/4 + i/4 + (3^(1/2)*d*1i)/12 - (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 +
(3^(1/2)*g*1i)/12 - (3^(1/2)*h*1i)/6 + (3^(1/2)*i*1i)/12) + log(d*f*9i - d*
e*6i + d*g*3i - d*h*3i + d*i*3i + e*h*6i + f*h*3i - g*h*3i - h*i*3i + 3*3^(
1/2)*d^2 - d^2*x*6i - f^2*x*3i - d^2*3i - f^2*6i - 2*3^(1/2)*d*e - 3*3^(1/2
)*d*f + 3^(1/2)*d*g + 4*3^(1/2)*e*f - 3*3^(1/2)*d*h + 3^(1/2)*d*i - 2*3^(1/
2)*e*h - 2*3^(1/2)*f*g + 3*3^(1/2)*f*h - 2*3^(1/2)*f*i + 3^(1/2)*g*h + 3^(1
/2)*h*i + d*f*x*9i - e*f*x*6i + d*h*x*3i + e*h*x*6i + f*g*x*3i - f*h*x*3i +
 f*i*x*3i - g*h*x*3i - h*i*x*3i - 3*3^(1/2)*f^2*x + 3*3^(1/2)*d*f*x + 2*3^(
1/2)*d*g*x + 2*3^(1/2)*e*f*x - 3*3^(1/2)*d*h*x + 2*3^(1/2)*d*i*x + 2*3^(1/2
)*e*h*x - 3^(1/2)*f*g*x + 3*3^(1/2)*f*h*x - 3^(1/2)*f*i*x - 3^(1/2)*g*h*x -
 3^(1/2)*h*i*x - 4*3^(1/2)*d*e*x)*(d/4 - f/4 + g/4 - i/4 + (3^(1/2)*d*1i)/1
2 - (3^(1/2)*e*1i)/6 + (3^(1/2)*f*1i)/12 + (3^(1/2)*g*1i)/12 - (3^(1/2)*h*1
i)/6 + (3^(1/2)*i*1i)/12) + (i*x^2)/2
```

3.20 $\int \frac{d+ex}{a+bx^2+cx^4} \, dx$

Optimal result

Integrand size = 20, antiderivative size = 189

$$\int \frac{d+ex}{a+bx^2+cx^4} \, dx = \frac{\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[Out] -e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(1/2)+d*arctan(x*2^
(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*2^(1/2)*c^(1/2)/(-4*a*c+b^2)^(1
/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)-d*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)
^(1/2))^(1/2))*2^(1/2)*c^(1/2)/(-4*a*c+b^2)^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1
/2)
```

Rubi [A] (verified)

Time = 0.13 (sec) , antiderivative size = 189, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.350, Rules used = {1687, 12, 1107, 211, 1121, 632, 212}

$$\int \frac{d+ex}{a+bx^2+cx^4} \, dx = \frac{\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)}{\sqrt{b^2-4ac}\sqrt{\sqrt{b^2-4ac}+b}} - \frac{e\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[In] Int[(d + e*x)/(a + b*x^2 + c*x^4),x]
```

```
[Out] (Sqrt[2]*Sqrt[c]*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])
/(Sqrt[b^2 - 4*a*c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) - (Sqrt[2]*Sqrt[c]*d*ArcTa
n[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(Sqrt[b^2 - 4*a*c]*Sqrt
[b + Sqrt[b^2 - 4*a*c]]) - (e*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/Sqr
t[b^2 - 4*a*c]
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1107

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(-1), x_Symbol] :> With[{q = Rt[b^
2 - 4*a*c, 2]}, Dist[c/q, Int[1/(b/2 - q/2 + c*x^2), x], x] - Dist[c/q, Int
[1/(b/2 + q/2 + c*x^2), x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c,
0] && PosQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d}{a+bx^2+cx^4}\,dx + \int \frac{ex}{a+bx^2+cx^4}\,dx \\
&= d\int \frac{1}{a+bx^2+cx^4}\,dx + e\int \frac{x}{a+bx^2+cx^4}\,dx \\
&= \frac{(cd)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{\sqrt{b^2-4ac}} - \frac{(cd)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{\sqrt{b^2-4ac}} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right) \\
&= \frac{\sqrt{2}\sqrt{c}d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{2}\sqrt{c}d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad - e\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right) \\
&= \frac{\sqrt{2}\sqrt{c}d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{2}\sqrt{c}d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.17 (sec) , antiderivative size = 194, normalized size of antiderivative = 1.03

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx = \frac{\frac{2\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b-\sqrt{b^2-4ac}}} - \frac{2\sqrt{2}\sqrt{c}d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{b+\sqrt{b^2-4ac}}} + e\left(\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right) - \log\left(b+\sqrt{b^2-4\right.\right.}{2\sqrt{b^2-4ac}}$$

```
[In] Integrate[(d + e*x)/(a + b*x^2 + c*x^4),x]
```

```
[Out] ((2*Sqrt[2]*Sqrt[c]*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]
]])/Sqrt[b - Sqrt[b^2 - 4*a*c]] - (2*Sqrt[2]*Sqrt[c]*d*ArcTan[(Sqrt[2]*Sqrt
[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/Sqrt[b + Sqrt[b^2 - 4*a*c]] + e*(Log[-
b + Sqrt[b^2 - 4*a*c] - 2*c*x^2] - Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2]))/(
2*Sqrt[b^2 - 4*a*c])
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.08 (sec) , antiderivative size = 43, normalized size of antiderivative = 0.23

method	result
risch	$\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2 b+a\right)} \frac{(\_Re+d)\ln(x-\_R)}{2c\_R^3+\_Rb}\right)}{2}$
default	$4c\left(\frac{\sqrt{-4ac+b^2}\left(\frac{e\ln\left(2c\,x^2+\sqrt{-4ac+b^2}+b\right)}{4c}+\frac{d\sqrt{2}\,\arctan\left(\frac{cx\sqrt{2}}{\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{2\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{8ac-2b^2}-\frac{\sqrt{-4ac+b^2}\left(\frac{e\ln\left(-2c\,x^2+\sqrt{-4ac+b^2}-b\right)}{4c}-\frac{d\sqrt{2}\ \mathrm{ar}}{}\right.}{8ac-2b^2}\right.$

```
[In] int((e*x+d)/(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)
[Out] 1/2*sum((_R*e+d)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 3.16 (sec) , antiderivative size = 398481, normalized size of antiderivative = 2108.37

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a),x, algorithm="fricas")
[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx = \text{Timed out}$$

```
[In] integrate((e*x+d)/(c*x**4+b*x**2+a),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx=\int \frac{ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] integrate((e*x + d)/(c*x^4 + b*x^2 + a), x)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1342 vs. $2(149) = 298$.

Time = 1.40 (sec) , antiderivative size = 1342, normalized size of antiderivative = 7.10

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a),x, algorithm="giac")
[Out] 1/4*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4 - 8*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c
 - 2*b^4*c + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^2 + 8*sqrt(2)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*b^2*c^2 + 16*a*b^2*c^2 + 2*b^3*c^2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a*c^3 - 32*a^2*c^3 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*b*c^2 + 2*(b^2 - 4*a*c)*b^2*c - 8*(b^2 - 4*a*c)*a*c^2 - 2*(b^2 - 4*a*c)*
b*c^2)*d*arctan(2*sqrt(1/2)*x/sqrt((b + sqrt(b^2 - 4*a*c))/c))/((a*b^4 - 8*
a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*a
bs(c)) + 1/4*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4 - 8*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*b^3*c + 2*b^4*c + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^2 +
8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + sqrt(2)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*b^2*c^2 - 16*a*b^2*c^2 + 2*b^3*c^2 - 4*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a*c^3 + 32*a^2*c^3 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b^2*c + 8*(b^2 - 4*a*c)*a*c^2 - 2*(b^2
- 4*a*c)*b*c^2)*d*arctan(2*sqrt(1/2)*x/sqrt((b - sqrt(b^2 - 4*a*c))/c))/((a
*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a
^2*c^3)*abs(c)) - 1/2*(b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c)
```

```
*e*log(x^2 + 1/2*(b - sqrt(b^2 - 4*a*c))/c)/((b^4 - 8*a*b^2*c - 2*b^3*c + 1
6*a^2*c^2 + 8*a*b*c^2 + b^2*c^2 - 4*a*c^3)*c^2) + 1/4*(b^5*c - 8*a*b^3*c^2
- 2*b^4*c^2 + 16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4 + (b^4*c - 6
*a*b^2*c^2 - 2*b^3*c^2 + 8*a^2*c^3 + 4*a*b*c^3 + b^2*c^3 - 2*a*c^4)*sqrt(b^
2 - 4*a*c))*e*log(x^2 + 1/2*(b + sqrt(b^2 - 4*a*c))/c)/((a*b^4 - 8*a^2*b^2*
c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2)
```

Mupad [B] (verification not implemented)

Time = 8.36 (sec) , antiderivative size = 1308, normalized size of antiderivative = 6.92

$$\int \frac{d+ex}{a+bx^2+cx^4}\,dx = \sum_{k=1}^{4} \ln\Big(c^2\Big(d\,e^2+e^3\,x$$
$$+\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$-\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$-\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$+\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a\,c\,d^2\,e$$
$$-\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a\,c\,d^2\,e$$
$$-\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$+\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a\,c\,d^2\,e$$
$$+\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$+\mathrm{root}\big(128\,a^2\,b^2\,c\,z^4-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2+16\,a$$
$$-256\,a^3\,c^2\,z^4-16\,a\,b^4\,z^4+16\,a\,b\,c\,d^2\,z^2-32\,a^2\,c\,e^2\,z^2+8\,a\,b^2\,e^2\,z^2-4\,b^3\,d^2\,z^2$$
$$+16\,a\,c\,d^2\,e\,z-4\,b^2\,d^2\,e\,z-b\,d^2\,e^2-c\,d^4-a\,e^4,z,k\big)$$

```
[In] int((d + e*x)/(a + b*x^2 + c*x^4),x)
```

```
[Out] symsum(log(c^2*(d*e^2 + e^3*x + 4*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4
- 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*
b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4, z
, k)^2*b^2*d - 8*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 +
16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*
a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4, z, k)^3*b^3*x - 16
*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2
 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*
b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4, z, k)^2*a*c*d + 2*root(128*a^2*b^2
*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z
^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d
^2*e^2 - c*d^4 - a*e^4, z, k)*b*e^2*x - 4*root(128*a^2*b^2*c*z^4 - 256*a^3*
c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*
z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 -
```

```
a*e^4, z, k)*c*d^2*x - 4*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^
4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z
^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4, z, k)^2*b^
2*e*x + 4*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*
c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2
*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4, z, k)*b*d*e + 32*root(128
*a^2*b^2*c*z^4 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2
*c*e^2*z^2 + 8*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e
*z - b*d^2*e^2 - c*d^4 - a*e^4, z, k)^3*a*b*c*x + 16*root(128*a^2*b^2*c*z^4
 - 256*a^3*c^2*z^4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8
*a*b^2*e^2*z^2 - 4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2
 - c*d^4 - a*e^4, z, k)^2*a*c*e*x))*root(128*a^2*b^2*c*z^4 - 256*a^3*c^2*z^
4 - 16*a*b^4*z^4 + 16*a*b*c*d^2*z^2 - 32*a^2*c*e^2*z^2 + 8*a*b^2*e^2*z^2 -
4*b^3*d^2*z^2 + 16*a*c*d^2*e*z - 4*b^2*d^2*e*z - b*d^2*e^2 - c*d^4 - a*e^4,
 z, k), k, 1, 4)
```

3.21 $\int \frac{d+ex+fx^2}{a+bx^2+cx^4} \, dx$

Optimal result

Integrand size = 25, antiderivative size = 211

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4} \, dx = \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[Out] -e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(1/2)+1/2*arctan(x*
2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*(f+(-b*f+2*c*d)/(-4*a*c+b^2)^
(1/2))*2^(1/2)/c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/2*arctan(x*2^(1/2)*c^
(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*(f+(b*f-2*c*d)/(-4*a*c+b^2)^(1/2))*2^(1
/2)/c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 211, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.280, Rules used = {1687, 1180, 211, 12, 1121, 632, 212}

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4} \, dx = \frac{\arctan\left(\frac{\sqrt{2}\sqrt{cx}}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{2cd-bf}{\sqrt{b^2-4ac}}+f\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\arctan\left(\frac{\sqrt{2}\sqrt{cx}}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)}{\sqrt{2}\sqrt{c}\sqrt{\sqrt{b^2-4ac}+b}} - \frac{e\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[In] Int[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4),x]
```

```
[Out] ((f + (2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b -
Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((f -
(2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b
^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (e*ArcTanh[(
b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/Sqrt[b^2 - 4*a*c]
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
```

```
&&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{ex}{a+bx^2+cx^4}\,dx + \int \frac{d+fx^2}{a+bx^2+cx^4}\,dx \\
&= e\int \frac{x}{a+bx^2+cx^4}\,dx + \frac{1}{2}\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&\quad + \frac{1}{2}\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right) \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad - e\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right) \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.14 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.11

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx$$
$$= \frac{\frac{\sqrt{2}\left(2cd+\left(-b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{2}\left(-2cd+\left(b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} + e\log\left(-b+\sqrt{b^2-4ac}\right.}{2\sqrt{b^2-4ac}}$$

```
[In] Integrate[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4),x]
[Out] ((Sqrt[2]*(2*c*d + (-b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/S
qrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[
```

```
2]*(-2*c*d + (b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
 Sqrt[b^2 - 4*a*c]]])/(Sqrt[c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + e*Log[-b + Sq
rt[b^2 - 4*a*c] - 2*c*x^2] - e*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(2*Sqr
t[b^2 - 4*a*c])
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.07 (sec) , antiderivative size = 48, normalized size of antiderivative = 0.23

method	result
risch	$\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2 b+a\right)} \frac{\left(\_R^2 f+\_R e+d\right)\ln\left(x-\_R\right)}{2c\_R^3+\_R b}\right)}{2}$
default	$4c\left(-\frac{\sqrt{-4ac+b^2}\left(-\frac{e\ln\left(2cx^2+\sqrt{-4ac+b^2}+b\right)}{2}+\frac{\left(f\sqrt{-4ac+b^2}+bf-2cd\right)\sqrt{2}\arctan\left(\frac{cx\sqrt{2}}{\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{2\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{4c(4ac-b^2)}-\frac{\sqrt{-4ac+b^2}\left(\frac{e\ln\left(-2\right.}{}\right.}{}\right.$

```
[In] int((f*x^2+e*x+d)/(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)

[Out] 1/2*sum((_R^2*f+_R*e+d)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a)
)
```

Fricas [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 18.02 (sec) , antiderivative size = 723401, normalized size of antiderivative = 3428.44

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="fricas")

[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(c*x**4+b*x**2+a),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx = \int \frac{fx^2+ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] integrate((f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a), x)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1714 vs. $2(173) = 346$.

Time = 1.29 (sec) , antiderivative size = 1714, normalized size of antiderivative = 8.12

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="giac")
[Out] -1/2*(b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*
(b - sqrt(b^2 - 4*a*c))/c)/((b^4 - 8*a*b^2*c - 2*b^3*c + 16*a^2*c^2 + 8*a*b
*c^2 + b^2*c^2 - 4*a*c^3)*c^2) + 1/4*((sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*b^4 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c - 2*b^4*c + 16*sqrt(2)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^2*c^2 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2
+ sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^2 + 16*a*b^2*c^2 + 2*b^3*c^
2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^3 - 32*a^2*c^3 - 8*a*b*c^
3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^2 + 2*(b^2 - 4*a*c)*b^2*c - 8*(b^2
 - 4*a*c)*a*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^
```

```
2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*f)*arctan(
2*sqrt(1/2)*x/sqrt((b + sqrt(b^2 - 4*a*c))/c))/((a*b^4 - 8*a^2*b^2*c - 2*a*
b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*abs(c)) + 1/4*((s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c + 2*b
^4*c + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^2 + 8*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*b^2*c^2 - 16*a*b^2*c^2 + 2*b^3*c^2 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*c^3 + 32*a^2*c^3 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c
^2 - 2*(b^2 - 4*a*c)*b^2*c + 8*(b^2 - 4*a*c)*a*c^2 - 2*(b^2 - 4*a*c)*b*c^2)
*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^2
- 2*(b^2 - 4*a*c)*a*c^2)*f)*arctan(2*sqrt(1/2)*x/sqrt((b - sqrt(b^2 - 4*a*c
))/c))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2
*c^2 - 4*a^2*c^3)*abs(c)) + 1/4*(b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 + 16*a^2*b
*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4 + (b^4*c - 6*a*b^2*c^2 - 2*b^3*c^2
 + 8*a^2*c^3 + 4*a*b*c^3 + b^2*c^3 - 2*a*c^4)*sqrt(b^2 - 4*a*c))*e*log(x^2
+ 1/2*(b + sqrt(b^2 - 4*a*c))/c)/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3
*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2)
```

Mupad [B] (verification not implemented)

Time = 8.64 (sec) , antiderivative size = 3942, normalized size of antiderivative = 18.68

$$\int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2)/(a + b*x^2 + c*x^4),x)
```

```
[Out] symsum(log(c^2*d*e^2 - c^2*d^2*f + c^2*e^3*x - a*c*f^3 - 8*root(16*a*b^4*c*
z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c
^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 +
32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z +
 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a
*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^
4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^3*b^3*c^2*x + b*c*d*f^2 - 16*roo
t(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z
^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c
^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^
```

```
2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*
d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2
*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*a*c^3*d - 4*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*c^3*d^2*x + 4*root(16
*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2 +
 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*d
^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c*
e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e^
2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^2
 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*b^2*c^2*d + 32*root(1
6*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
+ 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*
d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c
*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e
^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^
2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^3*a*b*c^3*x + 16*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*a*c^3*e*x + 4*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*a*c^2*f^2*x + 2*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b*c^2*e^2*x - 2*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b^2*c*f^2*x - 4*root(
```

```
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*b^2*c^2*e*x + 4*roo
t(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z
^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c
^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^
2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*
d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2
*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b*c^2*d*e - 8*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*a*c^2*e*f + b*c*e*f^2
*x - 2*c^2*d*e*f*x + 4*root(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*
c^3*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*
a*b^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2
 + 4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z -
 16*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f
^3 + b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4,
 z, k)*b*c^2*d*f*x)*root(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3
*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b
^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 +
4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16
*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3
+ b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z,
 k), k, 1, 4)
```

3.22 $\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4} dx$

Optimal result

Integrand size = 30, antiderivative size = 245

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4} dx = \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}} + \frac{g\log(a+bx^2+cx^4)}{4c}$$

```
[Out] 1/4*g*ln(c*x^4+b*x^2+a)/c-1/2*(-b*g+2*c*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)
^(1/2))/c/(-4*a*c+b^2)^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(
1/2))^(1/2))*(f+(-b*f+2*c*d)/(-4*a*c+b^2)^(1/2))*2^(1/2)/c^(1/2)/(b-(-4*a*c
+b^2)^(1/2))^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2
))*(f+(b*f-2*c*d)/(-4*a*c+b^2)^(1/2))*2^(1/2)/c^(1/2)/(b+(-4*a*c+b^2)^(1/2)
)^(1/2)
```

Rubi [A] (verified)

Time = 0.11 (sec) , antiderivative size = 245, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.267, Rules used

$= \{1687, 1180, 211, 1261, 648, 632, 212, 642\}$

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx = \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{2cd-bf}{\sqrt{b^2-4ac}}+f\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)}{\sqrt{2}\sqrt{c}\sqrt{\sqrt{b^2-4ac}+b}} - \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}} + \frac{g\log\left(a+bx^2+cx^4\right)}{4c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4),x]
```

```
[Out] ((f + (2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b -
Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((f -
(2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b
^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - ((2*c*e - b*
g)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(2*c*Sqrt[b^2 - 4*a*c]) + (g*L
og[a + b*x^2 + c*x^4])/(4*c)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
```

```
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{a+bx^2+cx^4}\,dx + \int \frac{x(e+gx^2)}{a+bx^2+cx^4}\,dx \\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{a+bx+cx^2}\,dx, x, x^2\right) \\
&\quad + \frac{1}{2}\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&\quad + \frac{1}{2}\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad + \frac{g\text{Subst}\left(\int \frac{b+2cx}{a+bx+cx^2}\,dx, x, x^2\right)}{4c} + \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{4c}
\end{aligned}$$

$$= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}}+\frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{g\log\left(a+bx^2+cx^4\right)}{4c}-\frac{(2ce-bg)\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{2c}$$

$$= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}}+\frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}}+\frac{g\log\left(a+bx^2+cx^4\right)}{4c}$$

Mathematica [A] (verified)

Time = 0.18 (sec) , antiderivative size = 280, normalized size of antiderivative = 1.14

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx$$

$$= \frac{\frac{2\sqrt{2}\sqrt{c}\left(2cd+\left(-b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b-\sqrt{b^2-4ac}}}+\frac{2\sqrt{2}\sqrt{c}\left(-2cd+\left(b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{b+\sqrt{b^2-4ac}}}+(2ce+(-b+}{4c\sqrt{b^2-}}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4),x]
```

```
[Out] ((2*Sqrt[2]*Sqrt[c]*(2*c*d + (-b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sq
rt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/Sqrt[b - Sqrt[b^2 - 4*a*c]] + (2*Sqr
t[2]*Sqrt[c]*(-2*c*d + (b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x
)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/Sqrt[b + Sqrt[b^2 - 4*a*c]] + (2*c*e + (-b
+ Sqrt[b^2 - 4*a*c])*g)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2] + (-2*c*e + (
b + Sqrt[b^2 - 4*a*c])*g)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(4*c*Sqrt[b
^2 - 4*a*c])
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.07 (sec) , antiderivative size = 53, normalized size of antiderivative = 0.22

method	result
risch	$\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2 b+a\right)}\frac{\left(\_R^3 g+\_R^2 f+\_Re+d\right)\ln\left(x-\_R\right)}{2c\_R^3+\_Rb}\right)}{2}$
default	$4c\left(-\frac{\sqrt{-4ac+b^2}\left(\frac{\left(\sqrt{-4ac+b^2}\,g+bg-2ec\right)\ln\left(2c\,x^2+\sqrt{-4ac+b^2}+b\right)}{4c}+\frac{\left(f\sqrt{-4ac+b^2}+bf-2cd\right)\sqrt{2}\arctan\left(\frac{cx\sqrt{2}}{\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{2\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{4c(4ac-b^2)}-\ldots\right.$

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)
[Out] 1/2*sum((_R^3*g+_R^2*f+_R*e+d)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z
^2*b+a))
```

Fricas [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 249.08 (sec) , antiderivative size = 2136355, normalized size of antiderivative = 8719.82

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="fricas")
[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx=\int \frac{gx^3+fx^2+ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a), x)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3270 vs. $2(203) = 406$.

Time = 1.33 (sec) , antiderivative size = 3270, normalized size of antiderivative = 13.35

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx=\text{Too large to display}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="giac")
[Out] 1/4*g*log(abs(c*x^4 + b*x^2 + a))/c + 1/8*((2*b^4*c^2 - 16*a*b^2*c^3 + 32*a
^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4 + 8*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c + 2*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c - 16*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^2 - 8*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^3 - 2*(b^2 - 4*a*c)*b^2*c^2 + 8*(b^2 - 4*
a*c)*a*c^3)*c^2*f + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^2 - 8*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 - 2*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*b^3*c^3 - 2*b^4*c^3 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^2*c^4 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^4 + 16*a*b^2*c^4 - 4*sqrt(2)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^5 - 32*a^2*c^5 + 2*(b^2 - 4*a*c)*b^2*c^3 -
8*(b^2 - 4*a*c)*a*c^4)*d*abs(c) + 2*(2*b^3*c^5 - 8*a*b*c^6 - sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^3 + 4*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*b*c^5 - 2*(b^2 - 4*a*c)*b*c^5)*d - (2*b^4*c^4 -
8*a*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4
*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^
3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^3 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - 2*(b^2 -
 4*a*c)*b^2*c^4)*f)*arctan(2*sqrt(1/2)*x/sqrt((b*c + sqrt(b^2*c^2 - 4*a*c^3
))/c^2))/((a*b^4*c^2 - 8*a^2*b^2*c^3 - 2*a*b^3*c^3 + 16*a^3*c^4 + 8*a^2*b*c
```

```
^4 + a*b^2*c^4 - 4*a^2*c^5)*c^2) + 1/8*((2*b^4*c^2 - 16*a*b^2*c^3 + 32*a^2*
c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4 + 8*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c + 2*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c - 16*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^2 - 8*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a*c^3 - 2*(b^2 - 4*a*c)*b^2*c^2 + 8*(b^2 - 4*a*c
)*a*c^3)*c^2*f + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^2 - 8*sqr
t(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 - 2*sqrt(2)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*b^3*c^3 + 2*b^4*c^3 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^2*c^4 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + sqrt(2)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - 16*a*b^2*c^4 - 4*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a*c^5 + 32*a^2*c^5 - 2*(b^2 - 4*a*c)*b^2*c^3 + 8*(
b^2 - 4*a*c)*a*c^4)*d*abs(c) + 2*(2*b^3*c^5 - 8*a*b*c^6 - sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*b*c^5 - 2*(b^2 - 4*a*c)*b*c^5)*d - (2*b^4*c^4 - 8*a
*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^
2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 +
 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^3 - sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - 2*(b^2 - 4*
a*c)*b^2*c^4)*f)*arctan(2*sqrt(1/2)*x/sqrt((b*c - sqrt(b^2*c^2 - 4*a*c^3))/
c^2))/((a*b^4*c^2 - 8*a^2*b^2*c^3 - 2*a*b^3*c^3 + 16*a^3*c^4 + 8*a^2*b*c^4
+ a*b^2*c^4 - 4*a^2*c^5)*c^2) + 1/16*(2*(b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 +
16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4 - (b^4*c - 8*a*b^2*c^2 - 2
*b^3*c^2 + 16*a^2*c^3 + 8*a*b*c^3 + b^2*c^3 - 4*a*c^4)*sqrt(b^2 - 4*a*c))*e
*abs(c) - (b^6 - 8*a*b^4*c - 2*b^5*c + 16*a^2*b^2*c^2 + 8*a*b^3*c^2 + b^4*c
^2 - 4*a*b^2*c^3 + (b^5 - 8*a*b^3*c - 2*b^4*c + 16*a^2*b*c^2 + 8*a*b^2*c^2
+ b^3*c^2 - 4*a*b*c^3)*sqrt(b^2 - 4*a*c))*g*abs(c) - 2*(b^5*c^2 - 8*a*b^3*c
^3 - 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5 + (b^4*c^
2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (b^6*c - 8*a*
b^4*c^2 - 2*b^5*c^2 + 16*a^2*b^2*c^3 + 8*a*b^3*c^3 + b^4*c^3 - 4*a*b^2*c^4
+ (b^5*c - 4*a*b^3*c^2 - 2*b^4*c^2 + b^3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2
 + 1/2*(b*c + sqrt(b^2*c^2 - 4*a*c^3))/c^2)/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3
*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(c)) + 1/16*(
2*(b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 + 16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 -
 4*a*b*c^4 + (b^4*c - 8*a*b^2*c^2 - 2*b^3*c^2 + 16*a^2*c^3 + 8*a*b*c^3 + b^
2*c^3 - 4*a*c^4)*sqrt(b^2 - 4*a*c))*e*abs(c) - (b^6 - 8*a*b^4*c - 2*b^5*c +
 16*a^2*b^2*c^2 + 8*a*b^3*c^2 + b^4*c^2 - 4*a*b^2*c^3 - (b^5 - 8*a*b^3*c -
2*b^4*c + 16*a^2*b*c^2 + 8*a*b^2*c^2 + b^3*c^2 - 4*a*b*c^3)*sqrt(b^2 - 4*a*
c))*g*abs(c) - 2*(b^5*c^2 - 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^
2*c^4 + b^3*c^4 - 4*a*b*c^5 + (b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4)
*sqrt(b^2 - 4*a*c))*e + (b^6*c - 8*a*b^4*c^2 - 2*b^5*c^2 + 16*a^2*b^2*c^3 +
```

```
 8*a*b^3*c^3 + b^4*c^3 - 4*a*b^2*c^4 - (b^5*c - 4*a*b^3*c^2 - 2*b^4*c^2 + b
^3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2 + 1/2*(b*c - sqrt(b^2*c^2 - 4*a*c^3))
/c^2)/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*
c^2 - 4*a^2*c^3)*c^2*abs(c))
```

Mupad [B] (verification not implemented)

Time = 8.94 (sec) , antiderivative size = 15179, normalized size of antiderivative = 61.96

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4),x)
```

```
[Out] symsum(log(c^2*d*e^2 + b^2*d*g^2 - c^2*d^2*f + c^2*e^3*x - a*c*f^3 - 8*root
(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2
*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a
*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*
f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16
*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*
z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c
*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*
a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z
+ 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4
*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*
d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e
^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*
a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2
*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*
d^4, z, k)^3*b^3*c^2*x - a*c*d*g^2 + b*c*d*f^2 - a*b*f*g^2 - a*b*g^3*x - 16
*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^
2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 +
 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b
*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2
 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2
*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^
2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z
+ 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2
*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*
z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a
^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^
2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2
 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^
2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 -
 c^3*d^4, z, k)^2*a*c^3*d - 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 -
```

256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g
*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40
*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^
3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 3
2*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z
 + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2
*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*
b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16
*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*
d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e
*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 +
2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f
^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*
c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*c^3*d^2*x - 2*root(128*a^2*b^2
*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*
a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f
*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*
a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2
*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^
4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4
*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*
g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*
g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f
*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*
b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b
*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^
2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2
*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*b
^3*g^2*x + b^2*e*g^2*x + c^2*d^2*g*x + 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^
4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 25
6*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e
*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2
- 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*
g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^
2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z -
 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e
*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d
^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g
 + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*
a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2
*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^
2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2
*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)^2*b^2*c^2*d + 32*ro
ot(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c
^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16

*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^
2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 +
16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^
2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b
*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 1
6*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*
z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z -
 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*
c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d
*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 -
2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e
^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^
3*d^4, z, k)^3*a*b*c^3*x + 16*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 -
 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g
*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40
*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^
3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 3
2*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z
 + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2
*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*
b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16
*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*
d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e
*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 +
2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f
^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*
c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)^2*a*c^3*e*x + 4*root(128*a^2*b
^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 1
6*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d
*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 +
8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d
^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*
b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z -
 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^
2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*
e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e
*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 +
2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a
*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*
g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c
^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)
*a*c^2*f^2*x + 2*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*
z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2
*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^
2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4

*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2
*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2
*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*
a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e
*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z
- 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2
*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*
d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f
- a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*
c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2
*e^4 - a^3*g^4 - c^3*d^4, z, k)*b*c^2*e^2*x - 2*root(128*a^2*b^2*c^3*z^4 -
16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z
^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*
b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e
^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a
^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 -
8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^
2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^
2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*
a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*
c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e
*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2
+ 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2
*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 -
b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*b^2*c*f^2*x
+ 8*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2
*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^
2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^
2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*
z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*
c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16
*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2
*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*
d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g
^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g -
4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a
*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*
f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a
*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^
4 - c^3*d^4, z, k)^2*b^3*c*g*x - 2*b*c*d*e*g + 2*a*c*e*f*g - 4*root(128*a^2
*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 +
16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2
*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2
+ 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3
*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*

```
a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z
 - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*
e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^
3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d
*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2
+ 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3
*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^
2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b
*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z,
k)^2*b^2*c^2*e*x + 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*
c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32
*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*
c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2
 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3
*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2
*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z
+ 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d
^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^
3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4
*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*
b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d
^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 -
b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a
*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*b*c^2*d*e + 8*root(128*a^2*b^2*c^3*z^4
- 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g
*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*
a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2
*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96
*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2
 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*
e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*
a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 1
6*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*
b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2
*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^
2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c
^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2
- b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*a*c^2*d*g
- 8*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2
*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^
2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^
2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*
z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*
c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16
*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2
```

```
*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*
d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g
^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g -
4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a
*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*
f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a
*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^
4 - c^3*d^4, z, k)*a*c^2*e*f - 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^
4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^
3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 +
 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2
*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2
- 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*
g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*
c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z -
 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z +
 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b
*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^
2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3
 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*
b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a
^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*b^2*c*d*g + a*c*e*g^2*x + b
*c*e*f^2*x - a*c*f^2*g*x - 2*b*c*e^2*g*x - 2*c^2*d*e*f*x + 10*root(128*a^2*
b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 +
16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*
d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 +
 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*
d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a
*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z
- 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e
^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3
*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*
e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 +
 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*
a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2
*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*
c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k
)*a*b*c*g^2*x + 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4
*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^
2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g
^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 -
4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^
2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^
2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4
*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*
```

e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z
 - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^
2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2
*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*
f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2
*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^
2*e^4 - a^3*g^4 - c^3*d^4, z, k)*b*c^2*d*f*x - 8*root(128*a^2*b^2*c^3*z^4 -
 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*
z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a
*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*
e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*
a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2
 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e
^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a
^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16
*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b
*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*
e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2
 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^
2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 -
 b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*a*c^2*e*g*x
 - 32*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a
^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*
z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*
a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^
2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^
3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z -
16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f
^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*
c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2
*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g
 - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4
*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^
2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 -
 a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*
g^4 - c^3*d^4, z, k)^2*a*b*c^2*g*x + 4*root(128*a^2*b^2*c^3*z^4 - 16*a*b^4*
c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3 + 16*a*b^4*c*g*z^3 + 256*
a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c^2*d*f*z^2 - 8*a*b^3*c*e*g
*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^2 + 8*a*b^2*c^2*e^2*z^2 -
64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c^3*d^2*z^2 - 96*a^3*c^2*g^
2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 - 4*a*b^4*g^2*z^2 - 8*a*b^2*
c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2*z - 4*a*b^2*c*e^2*g*z - 1
6*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^2*e^2*g*z - 16*a^2*c^2*e*f
^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*b^3*e*g^2*z + 16*a*c^3*d^2
*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c*d*e*f*g + 2*a*b*c*e^3*g +

```
 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^2 + 2*b^2*c*d^2*e*g - 4*a*
c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f + 3*a*b*c*d^2*g^2 + 2*a^2*b
*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*e^2*g^2 - 2*a*c^2*d^2*f^2
- a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 - b*c^2*d^2*e^2 - b^3*d^2*g
^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z, k)*a*b*c*f*g)*root(128*a
^2*b^2*c^3*z^4 - 16*a*b^4*c^2*z^4 - 256*a^3*c^4*z^4 - 128*a^2*b^2*c^2*g*z^3
 + 16*a*b^4*c*g*z^3 + 256*a^3*c^3*g*z^3 + 32*a^2*b*c^2*e*g*z^2 + 16*a*b^2*c
^2*d*f*z^2 - 8*a*b^3*c*e*g*z^2 + 40*a^2*b^2*c*g^2*z^2 + 16*a^2*b*c^2*f^2*z^
2 + 8*a*b^2*c^2*e^2*z^2 - 64*a^2*c^3*d*f*z^2 - 4*a*b^3*c*f^2*z^2 + 16*a*b*c
^3*d^2*z^2 - 96*a^3*c^2*g^2*z^2 - 32*a^2*c^3*e^2*z^2 - 4*b^3*c^2*d^2*z^2 -
4*a*b^4*g^2*z^2 - 8*a*b^2*c*d*f*g*z + 32*a^2*c^2*d*f*g*z - 16*a^2*b*c*e*g^2
*z - 4*a*b^2*c*e^2*g*z - 16*a*b*c^2*d^2*g*z + 4*a*b^2*c*e*f^2*z + 16*a^2*c^
2*e^2*g*z - 16*a^2*c^2*e*f^2*z - 4*b^2*c^2*d^2*e*z + 4*b^3*c*d^2*g*z + 4*a*
b^3*e*g^2*z + 16*a*c^3*d^2*e*z + 16*a^3*c*g^3*z - 4*a^2*b^2*g^3*z - 4*a*b*c
*d*e*f*g + 2*a*b*c*e^3*g + 2*a*b*c*d*f^3 + 4*a^2*c*e*f^2*g - 4*a^2*c*d*f*g^
2 + 2*b^2*c*d^2*e*g - 4*a*c^2*d^2*e*g + 2*a*b^2*d*f*g^2 + 4*a*c^2*d*e^2*f +
 3*a*b*c*d^2*g^2 + 2*a^2*b*e*g^3 + 2*b*c^2*d^3*f - a*b*c*e^2*f^2 - 2*a^2*c*
e^2*g^2 - 2*a*c^2*d^2*f^2 - a^2*b*f^2*g^2 - b^2*c*d^2*f^2 - a*b^2*e^2*g^2 -
 b*c^2*d^2*e^2 - b^3*d^2*g^2 - a^2*c*f^4 - a*c^2*e^4 - a^3*g^4 - c^3*d^4, z
, k), k, 1, 4)
```

3.23 $\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4} dx$

Optimal result

Integrand size = 35, antiderivative size = 290

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx = \frac{hx}{c}+\frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}}+\frac{g\log\left(a+bx^2+cx^4\right)}{4c}$$

```
[Out] h*x/c+1/4*g*ln(c*x^4+b*x^2+a)/c-1/2*(-b*g+2*c*e)*arctanh((2*c*x^2+b)/(-4*a*
c+b^2)^(1/2))/c/(-4*a*c+b^2)^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+
b^2)^(1/2))^(1/2))*(c*f-b*h+(2*c^2*d+b^2*h-c*(2*a*h+b*f))/(-4*a*c+b^2)^(1/2
))/c^(3/2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2
)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*(c*f-b*h+(2*a*c*h-b^2*h+b*c*f-2*c^2*d)/(-4*
a*c+b^2)^(1/2))/c^(3/2)*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.45 (sec) , antiderivative size = 290, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.257, Rules used

$= \{1687, 1690, 1180, 211, 1261, 648, 632, 212, 642\}$

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx = \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{-c(2ah+bf)+b^2h+2c^2d}{\sqrt{b^2-4ac}}-bh+cf\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-2ach+b^2h-bcf+2c^2d}{\sqrt{b^2-4ac}}-bh+cf\right)}{\sqrt{2}c^{3/2}\sqrt{\sqrt{b^2-4ac}+b}} - \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}} + \frac{g\log\left(a+bx^2+cx^4\right)}{4c} + \frac{hx}{c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4),x]
```

```
[Out] (h*x)/c + ((c*f - b*h + (2*c^2*d + b^2*h - c*(b*f + 2*a*h))/Sqrt[b^2 - 4*a*
c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(3/
2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((c*f - b*h - (2*c^2*d - b*c*f + b^2*h -
2*a*c*h)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 -
4*a*c]]])/(Sqrt[2]*c^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - ((2*c*e - b*g)*Ar
cTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(2*c*Sqrt[b^2 - 4*a*c]) + (g*Log[a
+ b*x^2 + c*x^4])/(4*c)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{a+bx^2+cx^4}\,dx + \int \frac{d+fx^2+hx^4}{a+bx^2+cx^4}\,dx \\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{a+bx+cx^2}\,dx, x, x^2\right) + \int \left(\frac{h}{c} + \frac{cd-ah+(cf-bh)x^2}{c\,(a+bx^2+cx^4)}\right)dx \\
&= \frac{hx}{c} + \frac{\int \frac{cd-ah+(cf-bh)x^2}{a+bx^2+cx^4}\,dx}{c} + \frac{g\text{Subst}\left(\int \frac{b+2cx}{a+bx+cx^2}\,dx, x, x^2\right)}{4c} \\
&\quad + \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{4c}
\end{aligned}
$$

$$= \frac{hx}{c} + \frac{g\log\left(a+bx^2+cx^4\right)}{4c} - \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{2c}$$
$$+ \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{2c}$$
$$+ \frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{2c}$$
$$= \frac{hx}{c} + \frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c\sqrt{b^2-4ac}} + \frac{g\log\left(a+bx^2+cx^4\right)}{4c}$$

Mathematica [A] (verified)

Time = 0.31 (sec) , antiderivative size = 383, normalized size of antiderivative = 1.32

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx$$
$$= \frac{4\sqrt{c}hx + \frac{2\sqrt{2}\left(2c^2d+b\left(b-\sqrt{b^2-4ac}\right)h+c\left(-bf+\sqrt{b^2-4ac}f-2ah\right)\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{2\sqrt{2}\left(2c^2d+b\left(b+\sqrt{b^2-4ac}\right)h-c\left(bf+\sqrt{b^2-4\right.\right.}{\sqrt{b^2-4ac}\sqrt{b+\sqrt{}}}}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4),x]
```

```
[Out] (4*Sqrt[c]*h*x + (2*Sqrt[2]*(2*c^2*d + b*(b - Sqrt[b^2 - 4*a*c])*h + c*(-(b
*f) + Sqrt[b^2 - 4*a*c]*f - 2*a*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqr
t[b^2 - 4*a*c]]])/(Sqrt[b^2 - 4*a*c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) - (2*Sqrt
[2]*(2*c^2*d + b*(b + Sqrt[b^2 - 4*a*c])*h - c*(b*f + Sqrt[b^2 - 4*a*c]*f +
 2*a*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(Sqrt[b^2
 - 4*a*c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(2*c*e + (-b + Sqrt[b^2 -
 4*a*c])*g)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/Sqrt[b^2 - 4*a*c] + (Sqr
t[c]*(-2*c*e + (b + Sqrt[b^2 - 4*a*c])*g)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x
^2])/Sqrt[b^2 - 4*a*c])/(4*c^(3/2))
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.10 (sec) , antiderivative size = 78, normalized size of antiderivative = 0.27

method	result
risch	$\frac{hx}{c}+\frac{\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2b+a\right)}\frac{\left(cg\_R^3+(-bh+cf)\_R^2+\_Rce-ah+cd\right)\ln\left(x-\_R\right)}{2c\_R^3+\_Rb}}{2c}$
default	$\frac{hx}{c}+\frac{\sqrt{-4ac+b^2}\left(\frac{\left(-\sqrt{-4ac+b^2}\,cg-gbc+2e\,c^2\right)\ln\left(2c\,x^2+\sqrt{-4ac+b^2}+b\right)}{4c}+\frac{\left(\sqrt{-4ac+b^2}\,bh-\sqrt{-4ac+b^2}\,fc-2ach+b^2h-fbc+2c^2d\right)\sqrt{2}\,\text{arcta}}{2\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right.}{c(4ac-b^2)}$

[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE)

[Out] h*x/c+1/2/c*sum((c*g*_R^3+(-b*h+c*f)*_R^2+_R*c*e-a*h+c*d)/(2*_R^3*c+_R*b)*l
n(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="fricas")

[Out] Timed out

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a),x)

[Out] Timed out

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx=\int \frac{hx^4+gx^3+fx^2+ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="maxima")
[Out] h*x/c + integrate((c*g*x^3 + c*e*x + (c*f - b*h)*x^2 + c*d - a*h)/(c*x^4 +
b*x^2 + a), x)/c
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5199 vs. $2(248) = 496$.

Time = 1.47 (sec) , antiderivative size = 5199, normalized size of antiderivative = 17.93

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx=\text{Too large to display}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="giac")
[Out] h*x/c + 1/4*g*log(abs(c*x^4 + b*x^2 + a))/c - 1/8*((2*b^4*c^3 - 16*a*b^2*c^
4 + 32*a^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
b^4*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c
^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^2 -
16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^3 - 8*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^3 - sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^3 + 4*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^4 - 2*(b^2 - 4*a*c)*b^2*c^3
 + 8*(b^2 - 4*a*c)*a*c^4)*c^2*f - (2*b^5*c^2 - 16*a*b^3*c^3 + 32*a^2*b*c^4
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5 + 8*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c + 2*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c - 16*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^3 - 2*(b^2 - 4*a*c)*b^3*c^2 + 8*(b^2 - 4*a
*c)*a*b*c^3)*c^2*h - 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 8
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 2*sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*b^3*c^4 - 2*b^4*c^4 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*c^5 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^5 + sqr
t(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^5 + 16*a*b^2*c^5 - 4*sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^6 - 32*a^2*c^6 + 2*(b^2 - 4*a*c)*b^2*c^4 -
 8*(b^2 - 4*a*c)*a*c^5)*d*abs(c) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a*b^4*c^2 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 2*sq
```

```
rt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 2*a*b^4*c^3 + 16*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^2*b*c^4 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 16
*a^2*b^2*c^4 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^5 - 32*a^3*c
^5 + 2*(b^2 - 4*a*c)*a*b^2*c^3 - 8*(b^2 - 4*a*c)*a^2*c^4)*h*abs(c) + 2*(2*b
^3*c^6 - 8*a*b*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*b^3*c^4 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^
5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^6 - 2*(b^
2 - 4*a*c)*b*c^6)*d - (2*b^4*c^5 - 8*a*b^2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*b^2*c^5 - 2*(b^2 - 4*a*c)*b^2*c^5)*f + (2*b^5*c^4 - 12*a*b
^3*c^5 + 16*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b^5*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b
^4*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*
c^4 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4
 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 2*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 -
4*a*c)*b^3*c^4 + 4*(b^2 - 4*a*c)*a*b*c^5)*h)*arctan(2*sqrt(1/2)*x/sqrt((b*c
^3 + sqrt(b^2*c^6 - 4*a*c^7))/c^4))/((a*b^4*c^3 - 8*a^2*b^2*c^4 - 2*a*b^3*c
^4 + 16*a^3*c^5 + 8*a^2*b*c^5 + a*b^2*c^5 - 4*a^2*c^6)*c^2) + 1/8*((2*b^4*c
^3 - 16*a*b^2*c^4 + 32*a^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*b^4*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a*b^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*b^3*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b
*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^3 +
4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^4 - 2*(b^2
- 4*a*c)*b^2*c^3 + 8*(b^2 - 4*a*c)*a*c^4)*c^2*f - (2*b^5*c^2 - 16*a*b^3*c^3
 + 32*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c
+ 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c - 16*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^2 - 8*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 - sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^2 + 4*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^3 - 2*(b^2 - 4*a*c)*b^3*c^
2 + 8*(b^2 - 4*a*c)*a*b*c^3)*c^2*h + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 2*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 2*b^4*c^4 + 16*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^5 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a*b*c^5 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - 16*a*b^2*c^5
 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^6 + 32*a^2*c^6 - 2*(b^2 -
```

```
4*a*c)*b^2*c^4 + 8*(b^2 - 4*a*c)*a*c^5)*d*abs(c) - 2*(sqrt(2)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
2*b^2*c^3 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*a*b^4*c
^3 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 8*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*b^2*c^4 - 16*a^2*b^2*c^4 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
2*c^5 + 32*a^3*c^5 - 2*(b^2 - 4*a*c)*a*b^2*c^3 + 8*(b^2 - 4*a*c)*a^2*c^4)*h
*abs(c) + 2*(2*b^3*c^6 - 8*a*b*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*b^3*c^4 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a*b*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*b*c^6 - 2*(b^2 - 4*a*c)*b*c^6)*d - (2*b^4*c^5 - 8*a*b^2*c^6 - sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 + 4*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 2*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - 2*(b^2 - 4*a*c)*b^2*c^5)*f + (2*
b^5*c^4 - 12*a*b^3*c^5 + 16*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^2*b*c^4 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*b^3*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b
*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 4*(b^2 - 4*a*c)*a*b*c^5)*h)*arctan(2*sqrt(
1/2)*x/sqrt((b*c^3 - sqrt(b^2*c^6 - 4*a*c^7))/c^4))/((a*b^4*c^3 - 8*a^2*b^2
*c^4 - 2*a*b^3*c^4 + 16*a^3*c^5 + 8*a^2*b*c^5 + a*b^2*c^5 - 4*a^2*c^6)*c^2)
 + 1/16*(2*(b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 + 16*a^2*b*c^3 + 8*a*b^2*c^3 +
b^3*c^3 - 4*a*b*c^4 - (b^4*c - 8*a*b^2*c^2 - 2*b^3*c^2 + 16*a^2*c^3 + 8*a*b
*c^3 + b^2*c^3 - 4*a*c^4)*sqrt(b^2 - 4*a*c))*e*abs(c) - (b^6 - 8*a*b^4*c -
2*b^5*c + 16*a^2*b^2*c^2 + 8*a*b^3*c^2 + b^4*c^2 - 4*a*b^2*c^3 + (b^5 - 8*a
*b^3*c - 2*b^4*c + 16*a^2*b*c^2 + 8*a*b^2*c^2 + b^3*c^2 - 4*a*b*c^3)*sqrt(b
^2 - 4*a*c))*g*abs(c) - 2*(b^5*c^2 - 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4
 + 8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5 + (b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 +
 b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (b^6*c - 8*a*b^4*c^2 - 2*b^5*c^2 + 16*a^2*
b^2*c^3 + 8*a*b^3*c^3 + b^4*c^3 - 4*a*b^2*c^4 + (b^5*c - 4*a*b^3*c^2 - 2*b^
4*c^2 + b^3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2 + 1/2*(b*c^3 + sqrt(b^2*c^6
- 4*a*c^7))/c^4)/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c
^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(c)) + 1/16*(2*(b^5*c - 8*a*b^3*c^2 - 2*
b^4*c^2 + 16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4 + (b^4*c - 8*a*b
^2*c^2 - 2*b^3*c^2 + 16*a^2*c^3 + 8*a*b*c^3 + b^2*c^3 - 4*a*c^4)*sqrt(b^2 -
 4*a*c))*e*abs(c) - (b^6 - 8*a*b^4*c - 2*b^5*c + 16*a^2*b^2*c^2 + 8*a*b^3*c
^2 + b^4*c^2 - 4*a*b^2*c^3 - (b^5 - 8*a*b^3*c - 2*b^4*c + 16*a^2*b*c^2 + 8*
a*b^2*c^2 + b^3*c^2 - 4*a*b*c^3)*sqrt(b^2 - 4*a*c))*g*abs(c) - 2*(b^5*c^2 -
 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5
 + (b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (b^
```

```
6*c - 8*a*b^4*c^2 - 2*b^5*c^2 + 16*a^2*b^2*c^3 + 8*a*b^3*c^3 + b^4*c^3 - 4*
a*b^2*c^4 - (b^5*c - 4*a*b^3*c^2 - 2*b^4*c^2 + b^3*c^3)*sqrt(b^2 - 4*a*c))*
g)*log(x^2 + 1/2*(b*c^3 - sqrt(b^2*c^6 - 4*a*c^7))/c^4)/((a*b^4 - 8*a^2*b^2
*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(
c))
```

Mupad [B] (verification not implemented)

Time = 8.47 (sec) , antiderivative size = 5981, normalized size of antiderivative = 20.62

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4),x)
```

```
[Out] symsum(log((x*(c^3*e^3 + c^3*d^2*g + b^3*e*h^2 - a*b*c*g^3 - 2*c^3*d*e*f +
a*c^2*e*g^2 + b*c^2*e*f^2 - a*c^2*f^2*g - 2*b*c^2*e^2*g + b^2*c*e*g^2 - a*b
^2*g*h^2 + a^2*c*g*h^2 - 2*a*b*c*e*h^2 + 2*b*c^2*d*e*h - 2*a*c^2*d*g*h + 2*
a*c^2*e*f*h - 2*b^2*c*e*f*h + 2*a*b*c*f*g*h))/c - root(128*a^2*b^2*c^4*z^4
- 16*a*b^4*c^3*z^4 - 256*a^3*c^5*z^4 - 128*a^2*b^2*c^3*g*z^3 + 16*a*b^4*c^2
*g*z^3 + 256*a^3*c^4*g*z^3 + 32*a^2*b*c^3*e*g*z^2 + 32*a^2*b*c^3*d*h*z^2 -
8*a*b^3*c^2*e*g*z^2 - 8*a*b^3*c^2*d*h*z^2 + 16*a*b^2*c^3*d*f*z^2 + 8*a*b^4*
c*f*h*z^2 - 48*a^2*b^2*c^2*f*h*z^2 - 48*a^3*b*c^2*h^2*z^2 + 28*a^2*b^3*c*h^
2*z^2 + 16*a^2*b*c^3*f^2*z^2 - 4*a*b^3*c^2*f^2*z^2 + 8*a*b^2*c^3*e^2*z^2 +
64*a^3*c^3*f*h*z^2 - 64*a^2*c^4*d*f*z^2 - 4*a*b^4*c*g^2*z^2 + 16*a*b*c^4*d^
2*z^2 + 40*a^2*b^2*c^2*g^2*z^2 - 96*a^3*c^3*g^2*z^2 - 32*a^2*c^4*e^2*z^2 -
4*b^3*c^3*d^2*z^2 - 4*a*b^5*h^2*z^2 + 8*a^2*b^2*c*f*g*h*z + 32*a^2*b*c^2*e*
f*h*z - 8*a*b^2*c^2*d*f*g*z + 8*a*b^2*c^2*d*e*h*z - 8*a*b^3*c*e*f*h*z - 20*
a^2*b^2*c*e*h^2*z - 16*a^2*b*c^2*e*g^2*z - 4*a*b^2*c^2*e^2*g*z + 4*a*b^2*c^
2*e*f^2*z - 32*a^3*c^2*f*g*h*z + 32*a^2*c^3*d*f*g*z - 32*a^2*c^3*d*e*h*z +
16*a^3*b*c*g*h^2*z + 4*a*b^3*c*e*g^2*z - 16*a*b*c^3*d^2*g*z - 4*a^2*b^3*g*h
^2*z + 16*a^3*c^2*e*h^2*z + 16*a^2*c^3*e^2*g*z + 4*b^3*c^2*d^2*g*z - 16*a^2
*c^3*e*f^2*z - 4*b^2*c^3*d^2*e*z - 4*a^2*b^2*c*g^3*z + 4*a*b^4*e*h^2*z + 16
*a*c^4*d^2*e*z + 16*a^3*c^2*g^3*z - 4*a^2*b*c*e*f*g*h - 4*a*b*c^2*d*e*f*g +
 8*a^2*c^2*d*e*g*h - 2*a^2*b*c*d*g^2*h + 2*a*b^2*c*e^2*f*h - 4*a*b^2*c*d*f^
2*h - 2*a^2*b*c*d*f*h^2 - 2*a*b*c^2*d^2*f*h + 2*a*b^2*c*d*f*g^2 - 2*a*b*c^2
*d*e^2*h - 4*a^2*c^2*e^2*f*h + 2*a^2*b^2*e*g*h^2 + 4*a^2*c^2*e*f^2*g + 4*a^
2*c^2*d*f^2*h - 4*a^2*c^2*d*f*g^2 + 2*b^2*c^2*d^2*e*g + 3*a^2*b*c*e^2*h^2 +
 4*a*b^2*c*d^2*h^2 + 3*a*b*c^2*d^2*g^2 + 4*a^3*c*f*g^2*h - 4*a^3*c*e*g*h^2
+ 2*b^3*c*d^2*f*h + 2*a*b^3*d*f*h^2 - 4*a*c^3*d^2*e*g + 2*a^2*b*c*f^3*h + 4
*a*c^3*d*e^2*f + 2*a^2*b*c*e*g^3 + 2*a*b*c^2*e^3*g + 2*a*b*c^2*d*f^3 + 2*a^
3*b*f*h^3 + 4*a^3*c*d*h^3 + 4*a*c^3*d^3*h + 2*b*c^3*d^3*f - a^2*b*c*f^2*g^2
 - a*b^2*c*e^2*g^2 - a*b*c^2*e^2*f^2 - 6*a^2*c^2*d^2*h^2 - 2*a^2*c^2*e^2*g^
2 - 2*a^3*c*f^2*h^2 - 2*b^2*c^2*d^3*h - 2*a^2*b^2*d*h^3 - 2*a*c^3*d^2*f^2 -
 a^2*b^2*f^2*h^2 - b^2*c^2*d^2*f^2 - a^3*b*g^2*h^2 - b^3*c*d^2*g^2 - a*b^3*
```

```
e^2*h^2 - b*c^3*d^2*e^2 - b^4*d^2*h^2 - a^2*c^2*f^4 - a^3*c*g^4 - a*c^3*e^4
 - a^4*h^4 - c^4*d^4, z, k)*(root(128*a^2*b^2*c^4*z^4 - 16*a*b^4*c^3*z^4 -
256*a^3*c^5*z^4 - 128*a^2*b^2*c^3*g*z^3 + 16*a*b^4*c^2*g*z^3 + 256*a^3*c^4*
g*z^3 + 32*a^2*b*c^3*e*g*z^2 + 32*a^2*b*c^3*d*h*z^2 - 8*a*b^3*c^2*e*g*z^2 -
 8*a*b^3*c^2*d*h*z^2 + 16*a*b^2*c^3*d*f*z^2 + 8*a*b^4*c*f*h*z^2 - 48*a^2*b^
2*c^2*f*h*z^2 - 48*a^3*b*c^2*h^2*z^2 + 28*a^2*b^3*c*h^2*z^2 + 16*a^2*b*c^3*
f^2*z^2 - 4*a*b^3*c^2*f^2*z^2 + 8*a*b^2*c^3*e^2*z^2 + 64*a^3*c^3*f*h*z^2 -
64*a^2*c^4*d*f*z^2 - 4*a*b^4*c*g^2*z^2 + 16*a*b*c^4*d^2*z^2 + 40*a^2*b^2*c^
2*g^2*z^2 - 96*a^3*c^3*g^2*z^2 - 32*a^2*c^4*e^2*z^2 - 4*b^3*c^3*d^2*z^2 - 4
*a*b^5*h^2*z^2 + 8*a^2*b^2*c*f*g*h*z + 32*a^2*b*c^2*e*f*h*z - 8*a*b^2*c^2*d
*f*g*z + 8*a*b^2*c^2*d*e*h*z - 8*a*b^3*c*e*f*h*z - 20*a^2*b^2*c*e*h^2*z - 1
6*a^2*b*c^2*e*g^2*z - 4*a*b^2*c^2*e^2*g*z + 4*a*b^2*c^2*e*f^2*z - 32*a^3*c^
2*f*g*h*z + 32*a^2*c^3*d*f*g*z - 32*a^2*c^3*d*e*h*z + 16*a^3*b*c*g*h^2*z +
4*a*b^3*c*e*g^2*z - 16*a*b*c^3*d^2*g*z - 4*a^2*b^3*g*h^2*z + 16*a^3*c^2*e*h
^2*z + 16*a^2*c^3*e^2*g*z + 4*b^3*c^2*d^2*g*z - 16*a^2*c^3*e*f^2*z - 4*b^2*
c^3*d^2*e*z - 4*a^2*b^2*c*g^3*z + 4*a*b^4*e*h^2*z + 16*a*c^4*d^2*e*z + 16*a
^3*c^2*g^3*z - 4*a^2*b*c*e*f*g*h - 4*a*b*c^2*d*e*f*g + 8*a^2*c^2*d*e*g*h -
2*a^2*b*c*d*g^2*h + 2*a*b^2*c*e^2*f*h - 4*a*b^2*c*d*f^2*h - 2*a^2*b*c*d*f*h
^2 - 2*a*b*c^2*d^2*f*h + 2*a*b^2*c*d*f*g^2 - 2*a*b*c^2*d*e^2*h - 4*a^2*c^2*
e^2*f*h + 2*a^2*b^2*e*g*h^2 + 4*a^2*c^2*e*f^2*g + 4*a^2*c^2*d*f^2*h - 4*a^2
*c^2*d*f*g^2 + 2*b^2*c^2*d^2*e*g + 3*a^2*b*c*e^2*h^2 + 4*a*b^2*c*d^2*h^2 +
3*a*b*c^2*d^2*g^2 + 4*a^3*c*f*g^2*h - 4*a^3*c*e*g*h^2 + 2*b^3*c*d^2*f*h + 2
*a*b^3*d*f*h^2 - 4*a*c^3*d^2*e*g + 2*a^2*b*c*f^3*h + 4*a*c^3*d*e^2*f + 2*a^
2*b*c*e*g^3 + 2*a*b*c^2*e^3*g + 2*a*b*c^2*d*f^3 + 2*a^3*b*f*h^3 + 4*a^3*c*d
*h^3 + 4*a*c^3*d^3*h + 2*b*c^3*d^3*f - a^2*b*c*f^2*g^2 - a*b^2*c*e^2*g^2 -
a*b*c^2*e^2*f^2 - 6*a^2*c^2*d^2*h^2 - 2*a^2*c^2*e^2*g^2 - 2*a^3*c*f^2*h^2 -
 2*b^2*c^2*d^3*h - 2*a^2*b^2*d*h^3 - 2*a*c^3*d^2*f^2 - a^2*b^2*f^2*h^2 - b^
2*c^2*d^2*f^2 - a^3*b*g^2*h^2 - b^3*c*d^2*g^2 - a*b^3*e^2*h^2 - b*c^3*d^2*e
^2 - b^4*d^2*h^2 - a^2*c^2*f^4 - a^3*c*g^4 - a*c^3*e^4 - a^4*h^4 - c^4*d^4,
 z, k)*((x*(4*b^2*c^3*e - 8*b^3*c^2*g - 16*a*c^4*e + 32*a*b*c^3*g))/c - (4*
b^2*c^3*d + 16*a^2*c^3*h - 16*a*c^4*d - 4*a*b^2*c^2*h)/c + (root(128*a^2*b^
2*c^4*z^4 - 16*a*b^4*c^3*z^4 - 256*a^3*c^5*z^4 - 128*a^2*b^2*c^3*g*z^3 + 16
*a*b^4*c^2*g*z^3 + 256*a^3*c^4*g*z^3 + 32*a^2*b*c^3*e*g*z^2 + 32*a^2*b*c^3*
d*h*z^2 - 8*a*b^3*c^2*e*g*z^2 - 8*a*b^3*c^2*d*h*z^2 + 16*a*b^2*c^3*d*f*z^2
+ 8*a*b^4*c*f*h*z^2 - 48*a^2*b^2*c^2*f*h*z^2 - 48*a^3*b*c^2*h^2*z^2 + 28*a^
2*b^3*c*h^2*z^2 + 16*a^2*b*c^3*f^2*z^2 - 4*a*b^3*c^2*f^2*z^2 + 8*a*b^2*c^3*
e^2*z^2 + 64*a^3*c^3*f*h*z^2 - 64*a^2*c^4*d*f*z^2 - 4*a*b^4*c*g^2*z^2 + 16*
a*b*c^4*d^2*z^2 + 40*a^2*b^2*c^2*g^2*z^2 - 96*a^3*c^3*g^2*z^2 - 32*a^2*c^4*
e^2*z^2 - 4*b^3*c^3*d^2*z^2 - 4*a*b^5*h^2*z^2 + 8*a^2*b^2*c*f*g*h*z + 32*a^
2*b*c^2*e*f*h*z - 8*a*b^2*c^2*d*f*g*z + 8*a*b^2*c^2*d*e*h*z - 8*a*b^3*c*e*f
*h*z - 20*a^2*b^2*c*e*h^2*z - 16*a^2*b*c^2*e*g^2*z - 4*a*b^2*c^2*e^2*g*z +
4*a*b^2*c^2*e*f^2*z - 32*a^3*c^2*f*g*h*z + 32*a^2*c^3*d*f*g*z - 32*a^2*c^3*
d*e*h*z + 16*a^3*b*c*g*h^2*z + 4*a*b^3*c*e*g^2*z - 16*a*b*c^3*d^2*g*z - 4*a
^2*b^3*g*h^2*z + 16*a^3*c^2*e*h^2*z + 16*a^2*c^3*e^2*g*z + 4*b^3*c^2*d^2*g*
z - 16*a^2*c^3*e*f^2*z - 4*b^2*c^3*d^2*e*z - 4*a^2*b^2*c*g^3*z + 4*a*b^4*e*
```

h^2*z + 16*a*c^4*d^2*e*z + 16*a^3*c^2*g^3*z - 4*a^2*b*c*e*f*g*h - 4*a*b*c^2
*d*e*f*g + 8*a^2*c^2*d*e*g*h - 2*a^2*b*c*d*g^2*h + 2*a*b^2*c*e^2*f*h - 4*a*
b^2*c*d*f^2*h - 2*a^2*b*c*d*f*h^2 - 2*a*b*c^2*d^2*f*h + 2*a*b^2*c*d*f*g^2 -
2*a*b*c^2*d*e^2*h - 4*a^2*c^2*e^2*f*h + 2*a^2*b^2*e*g*h^2 + 4*a^2*c^2*e*f^
2*g + 4*a^2*c^2*d*f^2*h - 4*a^2*c^2*d*f*g^2 + 2*b^2*c^2*d^2*e*g + 3*a^2*b*c
*e^2*h^2 + 4*a*b^2*c*d^2*h^2 + 3*a*b*c^2*d^2*g^2 + 4*a^3*c*f*g^2*h - 4*a^3*
c*e*g*h^2 + 2*b^3*c*d^2*f*h + 2*a*b^3*d*f*h^2 - 4*a*c^3*d^2*e*g + 2*a^2*b*c
*f^3*h + 4*a*c^3*d*e^2*f + 2*a^2*b*c*e*g^3 + 2*a*b*c^2*e^3*g + 2*a*b*c^2*d*
f^3 + 2*a^3*b*f*h^3 + 4*a^3*c*d*h^3 + 4*a*c^3*d^3*h + 2*b*c^3*d^3*f - a^2*b
*c*f^2*g^2 - a*b^2*c*e^2*g^2 - a*b*c^2*e^2*f^2 - 6*a^2*c^2*d^2*h^2 - 2*a^2*
c^2*e^2*g^2 - 2*a^3*c*f^2*h^2 - 2*b^2*c^2*d^3*h - 2*a^2*b^2*d*h^3 - 2*a*c^3
*d^2*f^2 - a^2*b^2*f^2*h^2 - b^2*c^2*d^2*f^2 - a^3*b*g^2*h^2 - b^3*c*d^2*g^
2 - a*b^3*e^2*h^2 - b*c^3*d^2*e^2 - b^4*d^2*h^2 - a^2*c^2*f^4 - a^3*c*g^4 -
a*c^3*e^4 - a^4*h^4 - c^4*d^4, z, k)*x*(8*b^3*c^3 - 32*a*b*c^4))/c) - (4*b
*c^3*d*e + 8*a*c^3*d*g - 8*a*c^3*e*f - 4*b^2*c^2*d*g - 8*a^2*c^2*g*h + 4*a*
b*c^2*e*h + 4*a*b*c^2*f*g)/c + (x*(4*c^4*d^2 + 2*b^4*h^2 - 4*a*c^3*f^2 - 2*
b*c^3*e^2 + 2*b^3*c*g^2 + 2*b^2*c^2*f^2 + 4*a^2*c^2*h^2 - 4*b*c^3*d*f - 8*a
*c^3*d*h + 8*a*c^3*e*g - 4*b^3*c*f*h - 10*a*b*c^2*g^2 - 8*a*b^2*c*h^2 + 4*b
^2*c^2*d*h + 12*a*b*c^2*f*h))/c) - (a*c^2*f^3 - a^2*b*h^3 - c^3*d*e^2 + c^3
*d^2*f - b^3*d*h^2 + a*c^2*d*g^2 - b*c^2*d*f^2 - b^2*c*d*g^2 + a*b^2*f*h^2
+ a*c^2*e^2*h - b*c^2*d^2*h + a^2*c*f*h^2 - a^2*c*g^2*h + 2*a*b*c*d*h^2 + a
*b*c*f*g^2 - 2*a*b*c*f^2*h + 2*b*c^2*d*e*g - 2*a*c^2*d*f*h - 2*a*c^2*e*f*g
+ 2*b^2*c*d*f*h)/c)*root(128*a^2*b^2*c^4*z^4 - 16*a*b^4*c^3*z^4 - 256*a^3*c
^5*z^4 - 128*a^2*b^2*c^3*g*z^3 + 16*a*b^4*c^2*g*z^3 + 256*a^3*c^4*g*z^3 + 3
2*a^2*b*c^3*e*g*z^2 + 32*a^2*b*c^3*d*h*z^2 - 8*a*b^3*c^2*e*g*z^2 - 8*a*b^3*
c^2*d*h*z^2 + 16*a*b^2*c^3*d*f*z^2 + 8*a*b^4*c*f*h*z^2 - 48*a^2*b^2*c^2*f*h
*z^2 - 48*a^3*b*c^2*h^2*z^2 + 28*a^2*b^3*c*h^2*z^2 + 16*a^2*b*c^3*f^2*z^2 -
4*a*b^3*c^2*f^2*z^2 + 8*a*b^2*c^3*e^2*z^2 + 64*a^3*c^3*f*h*z^2 - 64*a^2*c^
4*d*f*z^2 - 4*a*b^4*c*g^2*z^2 + 16*a*b*c^4*d^2*z^2 + 40*a^2*b^2*c^2*g^2*z^2
- 96*a^3*c^3*g^2*z^2 - 32*a^2*c^4*e^2*z^2 - 4*b^3*c^3*d^2*z^2 - 4*a*b^5*h^
2*z^2 + 8*a^2*b^2*c*f*g*h*z + 32*a^2*b*c^2*e*f*h*z - 8*a*b^2*c^2*d*f*g*z +
8*a*b^2*c^2*d*e*h*z - 8*a*b^3*c*e*f*h*z - 20*a^2*b^2*c*e*h^2*z - 16*a^2*b*c
^2*e*g^2*z - 4*a*b^2*c^2*e^2*g*z + 4*a*b^2*c^2*e*f^2*z - 32*a^3*c^2*f*g*h*z
+ 32*a^2*c^3*d*f*g*z - 32*a^2*c^3*d*e*h*z + 16*a^3*b*c*g*h^2*z + 4*a*b^3*c
*e*g^2*z - 16*a*b*c^3*d^2*g*z - 4*a^2*b^3*g*h^2*z + 16*a^3*c^2*e*h^2*z + 16
*a^2*c^3*e^2*g*z + 4*b^3*c^2*d^2*g*z - 16*a^2*c^3*e*f^2*z - 4*b^2*c^3*d^2*e
*z - 4*a^2*b^2*c*g^3*z + 4*a*b^4*e*h^2*z + 16*a*c^4*d^2*e*z + 16*a^3*c^2*g^
3*z - 4*a^2*b*c*e*f*g*h - 4*a*b*c^2*d*e*f*g + 8*a^2*c^2*d*e*g*h - 2*a^2*b*c
*d*g^2*h + 2*a*b^2*c*e^2*f*h - 4*a*b^2*c*d*f^2*h - 2*a^2*b*c*d*f*h^2 - 2*a*
b*c^2*d^2*f*h + 2*a*b^2*c*d*f*g^2 - 2*a*b*c^2*d*e^2*h - 4*a^2*c^2*e^2*f*h +
2*a^2*b^2*e*g*h^2 + 4*a^2*c^2*e*f^2*g + 4*a^2*c^2*d*f^2*h - 4*a^2*c^2*d*f*
g^2 + 2*b^2*c^2*d^2*e*g + 3*a^2*b*c*e^2*h^2 + 4*a*b^2*c*d^2*h^2 + 3*a*b*c^2
*d^2*g^2 + 4*a^3*c*f*g^2*h - 4*a^3*c*e*g*h^2 + 2*b^3*c*d^2*f*h + 2*a*b^3*d*
f*h^2 - 4*a*c^3*d^2*e*g + 2*a^2*b*c*f^3*h + 4*a*c^3*d*e^2*f + 2*a^2*b*c*e*g
^3 + 2*a*b*c^2*e^3*g + 2*a*b*c^2*d*f^3 + 2*a^3*b*f*h^3 + 4*a^3*c*d*h^3 + 4*

```
a*c^3*d^3*h + 2*b*c^3*d^3*f - a^2*b*c*f^2*g^2 - a*b^2*c*e^2*g^2 - a*b*c^2*e
^2*f^2 - 6*a^2*c^2*d^2*h^2 - 2*a^2*c^2*e^2*g^2 - 2*a^3*c*f^2*h^2 - 2*b^2*c^
2*d^3*h - 2*a^2*b^2*d*h^3 - 2*a*c^3*d^2*f^2 - a^2*b^2*f^2*h^2 - b^2*c^2*d^2
*f^2 - a^3*b*g^2*h^2 - b^3*c*d^2*g^2 - a*b^3*e^2*h^2 - b*c^3*d^2*e^2 - b^4*
d^2*h^2 - a^2*c^2*f^4 - a^3*c*g^4 - a*c^3*e^4 - a^4*h^4 - c^4*d^4, z, k), k
, 1, 4) + (h*x)/c
```

3.24 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx$

Optimal result

Integrand size = 40, antiderivative size = 321

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx \\
&= \frac{hx}{c}+\frac{ix^2}{2c}+\frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} \\
&+\frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} \\
&-\frac{(2c^2e-bcg+b^2i-2aci)\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2\sqrt{b^2-4ac}}+\frac{(cg-bi)\log\left(a+bx^2+cx^4\right)}{4c^2}
\end{aligned}
$$

```
[Out] h*x/c+1/2*i*x^2/c+1/4*(-b*i+c*g)*ln(c*x^4+b*x^2+a)/c^2-1/2*(-2*a*c*i+b^2*i-
b*c*g+2*c^2*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/c^2/(-4*a*c+b^2)^(1/
2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*(c*f-b*h+(2*c
^2*d+b^2*h-c*(2*a*h+b*f))/(-4*a*c+b^2)^(1/2))/c^(3/2)*2^(1/2)/(b-(-4*a*c+b^
2)^(1/2))^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*
(c*f-b*h+(2*a*c*h-b^2*h+b*c*f-2*c^2*d)/(-4*a*c+b^2)^(1/2))/c^(3/2)*2^(1/2)/
(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.30 (sec) , antiderivative size = 321, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.250, Rules used = {1687, 1690, 1180, 211, 1677, 1671, 648, 632, 212, 642}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx = \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{-c(2ah+bf)+b^2h+2c^2d}{\sqrt{b^2-4ac}}-bh+cf\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-2ach+b^2h-bcf+2c^2d}{\sqrt{b^2-4ac}}-bh+cf\right)}{\sqrt{2}c^{3/2}\sqrt{\sqrt{b^2-4ac}+b}} - \frac{\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)(-2aci+b^2i-bcg+2c^2e)}{2c^2\sqrt{b^2-4ac}} + \frac{(cg-bi)\log\left(a+bx^2+cx^4\right)}{4c^2}+\frac{hx}{c}+\frac{ix^2}{2c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4),x]
```

```
[Out] (h*x)/c + (i*x^2)/(2*c) + ((c*f - b*h + (2*c^2*d + b^2*h - c*(b*f + 2*a*h))
/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]]
)/(Sqrt[2]*c^(3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((c*f - b*h - (2*c^2*d -
b*c*f + b^2*h - 2*a*c*h)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt
[b + Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (
(2*c^2*e - b*c*g + b^2*i - 2*a*c*i)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]
])/(2*c^2*Sqrt[b^2 - 4*a*c]) + ((c*g - b*i)*Log[a + b*x^2 + c*x^4])/(4*c^2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
```

```
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2+hx^4}{a+bx^2+cx^4}\,dx + \int \frac{x(e+gx^2+ix^4)}{a+bx^2+cx^4}\,dx$$

$$= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{a+bx+cx^2}\,dx, x, x^2\right) + \int \left(\frac{h}{c} + \frac{cd-ah+(cf-bh)x^2}{c\,(a+bx^2+cx^4)}\right) dx$$

$$= \frac{hx}{c} + \frac{1}{2}\text{Subst}\left(\int \left(\frac{i}{c} + \frac{ce-ai+(cg-bi)x}{c\,(a+bx+cx^2)}\right) dx, x, x^2\right) + \frac{\int \frac{cd-ah+(cf-bh)x^2}{a+bx^2+cx^4}\,dx}{c}$$

$$= \frac{hx}{c} + \frac{ix^2}{2c} + \frac{\text{Subst}\left(\int \frac{ce-ai+(cg-bi)x}{a+bx+cx^2}\,dx, x, x^2\right)}{2c} + \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{2c} + \frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{2c}$$

$$= \frac{hx}{c} + \frac{ix^2}{2c} + \frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(cg-bi)\text{Subst}\left(\int \frac{b+2cx}{a+bx+cx^2}\,dx, x, x^2\right)}{4c^2} + \frac{(2c^2e-bcg+b^2i-2aci)\,\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{4c^2}$$

$$= \frac{hx}{c} + \frac{ix^2}{2c} + \frac{\left(cf-bh+\frac{2c^2d+b^2h-c(bf+2ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(cf-bh-\frac{2c^2d-bcf+b^2h-2ach}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(cg-bi)\log\left(a+bx^2+cx^4\right)}{4c^2} - \frac{(2c^2e-bcg+b^2i-2aci)\,\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{2c^2}$$

$$= \frac{hx}{c} + \frac{ix^2}{2c} + \frac{\left(cf - bh + \frac{2c^2d + b^2h - c(bf + 2ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(cf - bh - \frac{2c^2d - bcf + b^2h - 2ach}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}c^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(2c^2e - bcg + b^2i - 2aci)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2\sqrt{b^2-4ac}} + \frac{(cg - bi)\log(a + bx^2 + cx^4)}{4c^2}$$

Mathematica [A] (verified)

Time = 0.39 (sec) , antiderivative size = 441, normalized size of antiderivative = 1.37

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + ix^5}{a + bx^2 + cx^4}\,dx$$
$$= \frac{4chx + 2cix^2 + \frac{2\sqrt{2}\sqrt{c}\left(2c^2d + b\left(b - \sqrt{b^2-4ac}\right)h + c\left(-bf + \sqrt{b^2-4ac}f - 2ah\right)\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{b^2-4ac}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{2\sqrt{2}\sqrt{c}\left(2c^2d + b\left(b + \sqrt{b^2-4ac}\right)\right.}{\sqrt{}}}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4),x]
```

```
[Out] (4*c*h*x + 2*c*i*x^2 + (2*Sqrt[2]*Sqrt[c]*(2*c^2*d + b*(b - Sqrt[b^2 - 4*a*
c])*h + c*(-(b*f) + Sqrt[b^2 - 4*a*c]*f - 2*a*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x
)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[b^2 - 4*a*c]*Sqrt[b - Sqrt[b^2 - 4*a*
c]]) - (2*Sqrt[2]*Sqrt[c]*(2*c^2*d + b*(b + Sqrt[b^2 - 4*a*c])*h - c*(b*f +
 Sqrt[b^2 - 4*a*c]*f + 2*a*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2
 - 4*a*c]]])/(Sqrt[b^2 - 4*a*c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((2*c^2*e +
b*(b - Sqrt[b^2 - 4*a*c])*i + c*(-(b*g) + Sqrt[b^2 - 4*a*c]*g - 2*a*i))*Log
[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/Sqrt[b^2 - 4*a*c] - ((2*c^2*e + b*(b +
Sqrt[b^2 - 4*a*c])*i - c*(b*g + Sqrt[b^2 - 4*a*c]*g + 2*a*i))*Log[b + Sqrt[
b^2 - 4*a*c] + 2*c*x^2])/Sqrt[b^2 - 4*a*c])/(4*c^2)
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.10 (sec) , antiderivative size = 99, normalized size of antiderivative = 0.31

method	result
risch	$\frac{hx}{c}+\frac{i\,x^{2}}{2c}+\frac{\sum_{\_R=\mathrm{RootOf}\left(c\_Z^{4}+\_Z^{2}b+a\right)}\frac{\left((-bi+gc)\_R^{3}+(-bh+cf)\_R^{2}+(-ai+ec)\_R-ah+cd\right)\ln\left(x-\_R\right)}{2c\_R^{3}+\_Rb}}{2c}$
default	$\frac{hx+\frac{1}{2}i\,x^{2}}{c}+\frac{\sqrt{-4ac+b^{2}}\left(\frac{\left(\sqrt{-4ac+b^{2}}\,bi-\sqrt{-4ac+b^{2}}\,cg-2aci+b^{2}i-gbc+2e\,c^{2}\right)\ln\left(2c\,x^{2}+\sqrt{-4ac+b^{2}}+b\right)}{4c}+\frac{\left(\sqrt{-4ac+b^{2}}\,bh-\sqrt{-4ac+b^{2}}\,fc-2\cdots\right.}{\cdots}\right.}{c\left(4ac-b^{2}\right)}$

[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x,method=_RETURNVERBOSE
)

[Out] h*x/c+1/2*i*x^2/c+1/2/c*sum(((-b*i+c*g)*_R^3+(-b*h+c*f)*_R^2+(-a*i+c*e)*_R-
a*h+c*d)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="fri
cas")

[Out] Timed out

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a),x)

[Out] Timed out

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx = \int \frac{ix^5+hx^4+gx^3+fx^2+ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="max
ima")
[Out] 1/2*(i*x^2 + 2*h*x)/c - integrate(-((c*g - b*i)*x^3 + (c*f - b*h)*x^2 + c*d
 - a*h + (c*e - a*i)*x)/(c*x^4 + b*x^2 + a), x)/c
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5941 vs. 2(277) = 554.

Time = 1.55 (sec) , antiderivative size = 5941, normalized size of antiderivative = 18.51

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x, algorithm="gia
c")
[Out] 1/4*(c*g - b*i)*log(abs(c*x^4 + b*x^2 + a))/c^2 + 1/2*(c*i*x^2 + 2*c*h*x)/c
^2 + 1/8*((2*b^4*c^3 - 16*a*b^2*c^4 + 32*a^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*b^3*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*b^2*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a*c^4 - 2*(b^2 - 4*a*c)*b^2*c^3 + 8*(b^2 - 4*a*c)*a*c^4)*c^2*f - (2*b^5*
c^2 - 16*a*b^3*c^3 + 32*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*b^4*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^2*b*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^
2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^2 +
 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^3 - 2*(b
^2 - 4*a*c)*b^3*c^2 + 8*(b^2 - 4*a*c)*a*b*c^3)*c^2*h + 2*(sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a*b^2*c^4 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - 2*b^4*c^4 +
 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^5 + 8*sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a*b*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*
c^5 + 16*a*b^2*c^5 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^6 - 32*a
```

```
^2*c^6 + 2*(b^2 - 4*a*c)*b^2*c^4 - 8*(b^2 - 4*a*c)*a*c^5)*d*abs(c) - 2*(sqr
t(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 8*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b
^3*c^3 - 2*a*b^4*c^3 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 +
 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 16*a^2*b^2*c^4 - 4*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^2*c^5 - 32*a^3*c^5 + 2*(b^2 - 4*a*c)*a*b^2*c^3 - 8*(b^2 -
 4*a*c)*a^2*c^4)*h*abs(c) + 2*(2*b^3*c^6 - 8*a*b*c^7 - sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b*c^6 - 2*(b^2 - 4*a*c)*b*c^6)*d - (2*b^4*c^5 - 8*a*b^
2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 +
 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 2*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - 2*(b^2 - 4*a*c
)*b^2*c^5)*f + (2*b^5*c^4 - 12*a*b^3*c^5 + 16*a^2*b*c^6 - sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*b^3*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 4*(b^2 - 4*a*c)*a*b*c^5)
*h)*arctan(2*sqrt(1/2)*x/sqrt((b*c^5 + sqrt(b^2*c^10 - 4*a*c^11))/c^6))/((a
*b^4*c^3 - 8*a^2*b^2*c^4 - 2*a*b^3*c^4 + 16*a^3*c^5 + 8*a^2*b*c^5 + a*b^2*c
^5 - 4*a^2*c^6)*c^2) - 1/8*((2*b^4*c^3 - 16*a*b^2*c^4 + 32*a^2*c^5 - sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c + 8*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 + 2*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^2 - 16*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a*c^4 - 2*(b^2 - 4*a*c)*b^2*c^3 + 8*(b^2 - 4*a*c)*a*c^
4)*c^2*f - (2*b^5*c^2 - 16*a*b^3*c^3 + 32*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*b^4*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^2*b*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a*b*c^3 - 2*(b^2 - 4*a*c)*b^3*c^2 + 8*(b^2 - 4*a*c)*a*b*c^3)*c^2*h - 2*
(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a*b^2*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3
*c^4 + 2*b^4*c^4 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^5 + 8*s
```

qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - 16*a*b^2*c^5 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^6 + 32*a^2*c^6 - 2*(b^2 - 4*a*c)*b^2*c^4 + 8*(b^2 - 4*a*c)*a*c^5)*d*abs(c) + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*a*b^4*c^3 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 16*a^2*b^2*c^4 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^5 + 32*a^3*c^5 - 2*(b^2 - 4*a*c)*a*b^2*c^3 + 8*(b^2 - 4*a*c)*a^2*c^4)*h*abs(c) + 2*(2*b^3*c^6 - 8*a*b*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^6 - 2*(b^2 - 4*a*c)*b*c^6)*d - (2*b^4*c^5 - 8*a*b^2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 - 2*(b^2 - 4*a*c)*b^2*c^5)*f + (2*b^5*c^4 - 12*a*b^3*c^5 + 16*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 4*(b^2 - 4*a*c)*a*b*c^5)*h)*arctan(2*sqrt(1/2)*x/sqrt((b*c^5 - sqrt(b^2*c^10 - 4*a*c^11))/c^6))/((a*b^4*c^3 - 8*a^2*b^2*c^4 - 2*a*b^3*c^4 + 16*a^3*c^5 + 8*a^2*b*c^5 + a*b^2*c^5 - 4*a^2*c^6)*c^2) + 1/16*(2*(b^5*c^2 - 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5 + (b^4*c^2 - 8*a*b^2*c^3 - 2*b^3*c^3 + 16*a^2*c^4 + 8*a*b*c^4 + b^2*c^4 - 4*a*c^5)*sqrt(b^2 - 4*a*c))*e*abs(c) - (b^6*c - 8*a*b^4*c^2 - 2*b^5*c^2 + 16*a^2*b^2*c^3 + 8*a*b^3*c^3 + b^4*c^3 - 4*a*b^2*c^4 + (b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 + 16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(c) + (b^7 - 10*a*b^5*c - 2*b^6*c + 32*a^2*b^3*c^2 + 12*a*b^4*c^2 + b^5*c^2 - 32*a^3*b*c^3 - 16*a^2*b^2*c^3 - 6*a*b^3*c^3 + 8*a^2*b*c^4 + (b^6 - 10*a*b^4*c - 2*b^5*c + 32*a^2*b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 32*a^3*c^3 - 16*a^2*b*c^3 - 6*a*b^2*c^3 + 8*a^2*c^4)*sqrt(b^2 - 4*a*c))*i*abs(c) + 2*(b^5*c^3 - 8*a*b^3*c^4 - 2*b^4*c^4 + 16*a^2*b*c^5 + 8*a*b^2*c^5 + b^3*c^5 - 4*a*b*c^6 + (b^4*c^3 - 4*a*b^2*c^4 - 2*b^3*c^4 + b^2*c^5)*sqrt(b^2 - 4*a*c))*e - (b^6*c^2 - 8*a*b^4*c^3 - 2*b^5*c^3 + 16*a^2*b^2*c^4 + 8*a*b^3*c^4 + b^4*c^4 - 4*a*b^2*c^5 - (b^5*c^2 - 4*a*b^3*c^3 - 2*b^4*c^3 + b^3*c^4)*sqrt(b^2 - 4*a*c))*g + (b^7*c - 10*a*b^5*c^2 - 2*b^6*c^2 + 32*a^2*b^3*c^3 + 12*a*b^4*c^3 + b^5*c^3 - 32*a^3*b*c^4 - 16*a^2*b^2*c^4 - 6*a*b^3*c^4 + 8*a^2*b*c^5 + (b^6*c - 6*a*b^4*c^2 - 2*b^5*c^2 + 8*a^2*b^2*c^3 + 4*a*b^3*c^3 + b^4*c^3 - 2*

a*b^2*c^4)*sqrt(b^2 - 4*a*c))*i)*log(x^2 + 1/2*(b*c^5 + sqrt(b^2*c^10 - 4*a
*c^11))/c^6)/((a*b^4*c - 8*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^2*b
*c^3 + a*b^2*c^3 - 4*a^2*c^4)*c^2*abs(c)) + 1/16*(2*(b^5*c^2 - 8*a*b^3*c^3
- 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5 + (b^4*c^2 -
8*a*b^2*c^3 - 2*b^3*c^3 + 16*a^2*c^4 + 8*a*b*c^4 + b^2*c^4 - 4*a*c^5)*sqrt
(b^2 - 4*a*c))*e*abs(c) - (b^6*c - 8*a*b^4*c^2 - 2*b^5*c^2 + 16*a^2*b^2*c^3
+ 8*a*b^3*c^3 + b^4*c^3 - 4*a*b^2*c^4 - (b^5*c - 8*a*b^3*c^2 - 2*b^4*c^2 +
16*a^2*b*c^3 + 8*a*b^2*c^3 + b^3*c^3 - 4*a*b*c^4)*sqrt(b^2 - 4*a*c))*g*abs
(c) + (b^7 - 10*a*b^5*c - 2*b^6*c + 32*a^2*b^3*c^2 + 12*a*b^4*c^2 + b^5*c^2
- 32*a^3*b*c^3 - 16*a^2*b^2*c^3 - 6*a*b^3*c^3 + 8*a^2*b*c^4 - (b^6 - 10*a*
b^4*c - 2*b^5*c + 32*a^2*b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 32*a^3*c^3 - 16
*a^2*b*c^3 - 6*a*b^2*c^3 + 8*a^2*c^4)*sqrt(b^2 - 4*a*c))*i*abs(c) - 2*(b^5*
c^3 - 8*a*b^3*c^4 - 2*b^4*c^4 + 16*a^2*b*c^5 + 8*a*b^2*c^5 + b^3*c^5 - 4*a*
b*c^6 + (b^4*c^3 - 4*a*b^2*c^4 - 2*b^3*c^4 + b^2*c^5)*sqrt(b^2 - 4*a*c))*e
+ (b^6*c^2 - 8*a*b^4*c^3 - 2*b^5*c^3 + 16*a^2*b^2*c^4 + 8*a*b^3*c^4 + b^4*c
^4 - 4*a*b^2*c^5 - (b^5*c^2 - 4*a*b^3*c^3 - 2*b^4*c^3 + b^3*c^4)*sqrt(b^2 -
4*a*c))*g - (b^7*c - 10*a*b^5*c^2 - 2*b^6*c^2 + 32*a^2*b^3*c^3 + 12*a*b^4*
c^3 + b^5*c^3 - 32*a^3*b*c^4 - 16*a^2*b^2*c^4 - 6*a*b^3*c^4 + 8*a^2*b*c^5 -
(b^6*c - 6*a*b^4*c^2 - 2*b^5*c^2 + 8*a^2*b^2*c^3 + 4*a*b^3*c^3 + b^4*c^3 -
2*a*b^2*c^4)*sqrt(b^2 - 4*a*c))*i)*log(x^2 + 1/2*(b*c^5 - sqrt(b^2*c^10 -
4*a*c^11))/c^6)/((a*b^4*c - 8*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^
2*b*c^3 + a*b^2*c^3 - 4*a^2*c^4)*c^2*abs(c))

Mupad [B] (verification not implemented)

Time = 8.74 (sec) , antiderivative size = 11383, normalized size of antiderivative = 35.46

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4),x)

[Out] symsum(log((x*(c^4*e^3 - a^3*c*i^3 + c^4*d^2*g + b^4*e*i^2 + a^2*b^2*i^3 +
b^2*c^2*e*g^2 + 3*a^2*c^2*e*i^2 + a^2*c^2*g*h^2 + 2*b^2*c^2*e^2*i - a^2*c^2
*g^2*i - 2*c^4*d*e*f - a*b*c^2*g^3 + a*c^3*e*g^2 + b*c^3*e*f^2 - a*c^3*f^2*
g - 2*b*c^3*e^2*g - 3*a*c^3*e^2*i - b*c^3*d^2*i + b^3*c*e*h^2 - a*b^3*g*i^2
- 2*a*b*c^2*e*h^2 - 3*a*b^2*c*e*i^2 - a*b^2*c*g*h^2 + 2*a*b^2*c*g^2*i + a^
2*b*c*h^2*i - 2*b^2*c^2*e*f*h - 2*a^2*c^2*f*h*i + 2*b*c^3*d*e*h + 2*a*c^3*d
*f*i - 2*a*c^3*d*g*h + 2*a*c^3*e*f*h - 2*b^3*c*e*g*i + 2*a*b*c^2*e*g*i + 2*
a*b*c^2*f*g*h))/c^2 - (a*c^3*f^3 - c^4*d*e^2 + c^4*d^2*f - b^4*d*i^2 - b^2*
c^2*d*g^2 - a^2*c^2*d*i^2 + a^2*c^2*f*h^2 - a^2*c^2*g^2*h - a^2*b^2*h*i^2 -
a^2*b*c*h^3 + a*c^3*d*g^2 - b*c^3*d*f^2 + a*c^3*e^2*h - b*c^3*d^2*h - b^3*
c*d*h^2 + a*b^3*f*i^2 + a^3*c*h*i^2 + 2*a*b*c^2*d*h^2 + a*b*c^2*f*g^2 + 3*a
*b^2*c*d*i^2 - 2*a*b*c^2*f^2*h + a*b^2*c*f*h^2 - 2*a^2*b*c*f*i^2 - 2*b^2*c^
2*d*e*i + 2*b^2*c^2*d*f*h - 2*a^2*c^2*e*h*i + 2*a^2*c^2*f*g*i + 2*b*c^3*d*e

*g + 2*a*c^3*d*e*i - 2*a*c^3*d*f*h - 2*a*c^3*e*f*g + 2*b^3*c*d*g*i - 4*a*b*
c^2*d*g*i + 2*a*b*c^2*e*f*i - 2*a*b^2*c*f*g*i + 2*a^2*b*c*g*h*i)/c^2 - root
(128*a^2*b^2*c^5*z^4 - 16*a*b^4*c^4*z^4 - 256*a^3*c^6*z^4 + 128*a^2*b^3*c^3
*i*z^3 - 128*a^2*b^2*c^4*g*z^3 - 256*a^3*b*c^4*i*z^3 - 16*a*b^5*c^2*i*z^3 +
16*a*b^4*c^3*g*z^3 + 256*a^3*c^5*g*z^3 + 160*a^3*b*c^3*g*i*z^2 + 8*a*b^4*c
^2*f*h*z^2 + 8*a*b^4*c^2*e*i*z^2 + 32*a^2*b*c^4*e*g*z^2 + 32*a^2*b*c^4*d*h*
z^2 - 8*a*b^3*c^3*e*g*z^2 - 8*a*b^3*c^3*d*h*z^2 + 16*a*b^2*c^4*d*f*z^2 + 8*
a*b^5*c*g*i*z^2 - 72*a^2*b^3*c^2*g*i*z^2 - 48*a^2*b^2*c^3*f*h*z^2 - 48*a^2*
b^2*c^3*e*i*z^2 + 32*a^2*b^4*c*i^2*z^2 - 48*a^3*b*c^3*h^2*z^2 - 4*a*b^4*c^2
*g^2*z^2 + 16*a^2*b*c^4*f^2*z^2 - 4*a*b^3*c^3*f^2*z^2 + 8*a*b^2*c^4*e^2*z^2
+ 64*a^3*c^4*f*h*z^2 + 64*a^3*c^4*e*i*z^2 - 64*a^2*c^5*d*f*z^2 - 4*a*b^5*c
*h^2*z^2 + 16*a*b*c^5*d^2*z^2 - 56*a^3*b^2*c^2*i^2*z^2 + 28*a^2*b^3*c^2*h^2
*z^2 + 40*a^2*b^2*c^3*g^2*z^2 - 32*a^4*c^3*i^2*z^2 - 96*a^3*c^4*g^2*z^2 - 3
2*a^2*c^5*e^2*z^2 - 4*b^3*c^4*d^2*z^2 - 4*a*b^6*i^2*z^2 + 32*a^2*b*c^3*e*f*
h*z - 32*a^2*b*c^3*d*f*i*z - 8*a*b^3*c^2*e*f*h*z + 8*a*b^3*c^2*d*f*i*z - 8*
a*b^2*c^3*d*f*g*z + 8*a*b^2*c^3*d*e*h*z - 8*a*b^4*c*e*g*i*z + 40*a^2*b^2*c^
2*e*g*i*z + 8*a^2*b^2*c^2*f*g*h*z - 8*a^2*b^2*c^2*d*h*i*z + 4*a^3*b^2*c*h^2
*i*z - 32*a^3*b*c^2*g^2*i*z + 12*a^3*b^2*c*g*i^2*z + 8*a^2*b^3*c*g^2*i*z +
16*a^3*b*c^2*g*h^2*z - 4*a^2*b^3*c*g*h^2*z + 32*a^3*b*c^2*e*i^2*z - 24*a^2*
b^3*c*e*i^2*z - 16*a^2*b*c^3*e^2*i*z + 4*a*b^3*c^2*e^2*i*z + 20*a*b^2*c^3*d
^2*i*z - 16*a^2*b*c^3*e*g^2*z + 4*a*b^3*c^2*e*g^2*z - 4*a*b^2*c^3*e^2*g*z +
4*a*b^2*c^3*e*f^2*z - 32*a^3*c^3*f*g*h*z - 32*a^3*c^3*e*g*i*z + 32*a^3*c^3
*d*h*i*z + 32*a^2*c^4*d*f*g*z - 32*a^2*c^4*d*e*h*z + 4*a*b^4*c*e*h^2*z - 16
*a*b*c^4*d^2*g*z - 4*a^2*b^2*c^2*f^2*i*z - 20*a^2*b^2*c^2*e*h^2*z - 4*a^2*b
^2*c^2*g^3*z - 16*a^4*c^2*h^2*i*z + 16*a^4*c^2*g*i^2*z + 16*a^3*c^3*f^2*i*z
- 4*a^2*b^4*g*i^2*z - 4*b^4*c^2*d^2*i*z + 16*a^3*c^3*e*h^2*z - 16*a^2*c^4*
d^2*i*z + 16*a^2*c^4*e^2*g*z + 4*b^3*c^3*d^2*g*z - 16*a^2*c^4*e*f^2*z - 4*b
^2*c^4*d^2*e*z + 4*a*b^5*e*i^2*z - 16*a^4*b*c*i^3*z + 16*a*c^5*d^2*e*z + 4*
a^3*b^3*i^3*z + 16*a^3*c^3*g^3*z + 4*a^2*b^2*c*d*g*h*i + 12*a^2*b*c^2*d*f*g
*i - 4*a^2*b*c^2*e*f*g*h - 4*a^2*b*c^2*d*e*h*i + 4*a*b^2*c^2*d*e*f*i - 4*a^
3*b*c*f*g*h*i - 4*a*b^3*c*d*f*g*i - 4*a*b*c^3*d*e*f*g + 2*a^2*b^2*c*f^2*g*i
- 4*a^2*b^2*c*e*g^2*i - 2*a^2*b*c^2*e^2*g*i - 8*a*b^2*c^2*d^2*g*i + 2*a^2*
b^2*c*e*g*h^2 - 2*a^2*b*c^2*e*f^2*i - 8*a^2*b^2*c*d*f*i^2 - 2*a^2*b*c^2*d*g
^2*h + 2*a*b^2*c^2*e^2*f*h - 4*a*b^2*c^2*d*f^2*h - 2*a^2*b*c^2*d*f*h^2 + 2*
a*b^2*c^2*d*f*g^2 + 8*a^3*c^2*e*f*h*i - 8*a^3*c^2*d*g*h*i + 8*a^2*c^3*d*e*g
*h - 8*a^2*c^3*d*e*f*i - 2*a^3*b*c*e*h^2*i + 6*a^3*b*c*d*h*i^2 - 2*a^3*b*c*
e*g*i^2 + 2*a*b^3*c*e^2*g*i + 6*a*b*c^3*d^2*e*i + 2*a*b^3*c*d*f*h^2 - 2*a*b
*c^3*d^2*f*h - 2*a*b*c^3*d*e^2*h + 4*a^2*b^2*c*e^2*i^2 - 5*a^2*b*c^2*d^2*i^
2 + 3*a^2*b*c^2*e^2*h^2 + 4*a*b^2*c^2*d^2*h^2 - 4*a^3*c^2*f^2*g*i + 2*a^3*b
^2*f*h*i^2 + 4*a^3*c^2*f*g^2*h + 4*a^3*c^2*e*g^2*i - 4*a^3*c^2*e*g*h^2 + 4*
a^2*c^3*d^2*g*i + 2*a^2*b^3*e*g*i^2 - 2*a^2*b^3*d*h*i^2 + 4*a^3*c^2*d*f*i^2
- 4*a^2*c^3*e^2*f*h + 2*b^3*c^2*d^2*f*h - 2*b^3*c^2*d^2*e*i + 4*a^2*c^3*e*
f^2*g + 4*a^2*c^3*d*f^2*h - 4*a^2*c^3*d*f*g^2 + 3*a^3*b*c*f^2*i^2 + 2*b^2*c
^3*d^2*e*g + 2*a^2*b*c^2*f^3*h - 2*a*b^2*c^2*e^3*i + 5*a*b^3*c*d^2*i^2 - 2*
a^2*b^2*c*d*h^3 + 2*a^2*b*c^2*e*g^3 + 3*a*b*c^3*d^2*g^2 + 4*a^4*c*g*h^2*i -

4*a^4*c*f*h*i^2 + 2*b^4*c*d^2*g*i + 2*a^3*b*c*g^3*i + 2*a*b^4*d*f*i^2 - 4*
a*c^4*d^2*e*g + 2*a^3*b*c*f*h^3 + 4*a*c^4*d*e^2*f + 2*a*b*c^3*e^3*g + 2*a*b
*c^3*d*f^3 - a^2*b^2*c*f^2*h^2 - a^2*b*c^2*f^2*g^2 - a*b^2*c^2*e^2*g^2 + 2*
a^4*b*g*i^3 + 4*a^4*c*e*i^3 + 4*a*c^4*d^3*h + 2*b*c^4*d^3*f - a^3*b*c*g^2*h
^2 - a*b^3*c*e^2*h^2 - 6*a^3*c^2*e^2*i^2 - 2*a^3*c^2*f^2*h^2 - a*b*c^3*e^2*
f^2 - 6*a^2*c^3*d^2*h^2 - 2*a^2*c^3*e^2*g^2 - 2*a^4*c*g^2*i^2 + 4*a^2*c^3*e
^3*i - 2*b^2*c^3*d^3*h - 2*a^3*b^2*e*i^3 + 4*a^3*c^2*d*h^3 - 2*a*c^4*d^2*f^
2 - a^3*b^2*g^2*i^2 - a^2*b^3*f^2*i^2 - b^3*c^2*d^2*g^2 - b^2*c^3*d^2*f^2 -
a^4*b*h^2*i^2 - b^4*c*d^2*h^2 - a*b^4*e^2*i^2 - b*c^4*d^2*e^2 - b^5*d^2*i^
2 - a^3*c^2*g^4 - a^2*c^3*f^4 - a^4*c*h^4 - a*c^4*e^4 - a^5*i^4 - c^5*d^4,
z, l)*(root(128*a^2*b^2*c^5*z^4 - 16*a*b^4*c^4*z^4 - 256*a^3*c^6*z^4 + 128*
a^2*b^3*c^3*i*z^3 - 128*a^2*b^2*c^4*g*z^3 - 256*a^3*b*c^4*i*z^3 - 16*a*b^5*
c^2*i*z^3 + 16*a*b^4*c^3*g*z^3 + 256*a^3*c^5*g*z^3 + 160*a^3*b*c^3*g*i*z^2
+ 8*a*b^4*c^2*f*h*z^2 + 8*a*b^4*c^2*e*i*z^2 + 32*a^2*b*c^4*e*g*z^2 + 32*a^2
*b*c^4*d*h*z^2 - 8*a*b^3*c^3*e*g*z^2 - 8*a*b^3*c^3*d*h*z^2 + 16*a*b^2*c^4*d
*f*z^2 + 8*a*b^5*c*g*i*z^2 - 72*a^2*b^3*c^2*g*i*z^2 - 48*a^2*b^2*c^3*f*h*z^
2 - 48*a^2*b^2*c^3*e*i*z^2 + 32*a^2*b^4*c*i^2*z^2 - 48*a^3*b*c^3*h^2*z^2 -
4*a*b^4*c^2*g^2*z^2 + 16*a^2*b*c^4*f^2*z^2 - 4*a*b^3*c^3*f^2*z^2 + 8*a*b^2*
c^4*e^2*z^2 + 64*a^3*c^4*f*h*z^2 + 64*a^3*c^4*e*i*z^2 - 64*a^2*c^5*d*f*z^2
- 4*a*b^5*c*h^2*z^2 + 16*a*b*c^5*d^2*z^2 - 56*a^3*b^2*c^2*i^2*z^2 + 28*a^2*
b^3*c^2*h^2*z^2 + 40*a^2*b^2*c^3*g^2*z^2 - 32*a^4*c^3*i^2*z^2 - 96*a^3*c^4*
g^2*z^2 - 32*a^2*c^5*e^2*z^2 - 4*b^3*c^4*d^2*z^2 - 4*a*b^6*i^2*z^2 + 32*a^2
*b*c^3*e*f*h*z - 32*a^2*b*c^3*d*f*i*z - 8*a*b^3*c^2*e*f*h*z + 8*a*b^3*c^2*d
*f*i*z - 8*a*b^2*c^3*d*f*g*z + 8*a*b^2*c^3*d*e*h*z - 8*a*b^4*c*e*g*i*z + 40
*a^2*b^2*c^2*e*g*i*z + 8*a^2*b^2*c^2*f*g*h*z - 8*a^2*b^2*c^2*d*h*i*z + 4*a^
3*b^2*c*h^2*i*z - 32*a^3*b*c^2*g^2*i*z + 12*a^3*b^2*c*g*i^2*z + 8*a^2*b^3*c
*g^2*i*z + 16*a^3*b*c^2*g*h^2*z - 4*a^2*b^3*c*g*h^2*z + 32*a^3*b*c^2*e*i^2*
z - 24*a^2*b^3*c*e*i^2*z - 16*a^2*b*c^3*e^2*i*z + 4*a*b^3*c^2*e^2*i*z + 20*
a*b^2*c^3*d^2*i*z - 16*a^2*b*c^3*e*g^2*z + 4*a*b^3*c^2*e*g^2*z - 4*a*b^2*c^
3*e^2*g*z + 4*a*b^2*c^3*e*f^2*z - 32*a^3*c^3*f*g*h*z - 32*a^3*c^3*e*g*i*z +
32*a^3*c^3*d*h*i*z + 32*a^2*c^4*d*f*g*z - 32*a^2*c^4*d*e*h*z + 4*a*b^4*c*e
*h^2*z - 16*a*b*c^4*d^2*g*z - 4*a^2*b^2*c^2*f^2*i*z - 20*a^2*b^2*c^2*e*h^2*
z - 4*a^2*b^2*c^2*g^3*z - 16*a^4*c^2*h^2*i*z + 16*a^4*c^2*g*i^2*z + 16*a^3*
c^3*f^2*i*z - 4*a^2*b^4*g*i^2*z - 4*b^4*c^2*d^2*i*z + 16*a^3*c^3*e*h^2*z -
16*a^2*c^4*d^2*i*z + 16*a^2*c^4*e^2*g*z + 4*b^3*c^3*d^2*g*z - 16*a^2*c^4*e*
f^2*z - 4*b^2*c^4*d^2*e*z + 4*a*b^5*e*i^2*z - 16*a^4*b*c*i^3*z + 16*a*c^5*d
^2*e*z + 4*a^3*b^3*i^3*z + 16*a^3*c^3*g^3*z + 4*a^2*b^2*c*d*g*h*i + 12*a^2*
b*c^2*d*f*g*i - 4*a^2*b*c^2*e*f*g*h - 4*a^2*b*c^2*d*e*h*i + 4*a*b^2*c^2*d*e
*f*i - 4*a^3*b*c*f*g*h*i - 4*a*b^3*c*d*f*g*i - 4*a*b*c^3*d*e*f*g + 2*a^2*b^
2*c*f^2*g*i - 4*a^2*b^2*c*e*g^2*i - 2*a^2*b*c^2*e^2*g*i - 8*a*b^2*c^2*d^2*g
*i + 2*a^2*b^2*c*e*g*h^2 - 2*a^2*b*c^2*e*f^2*i - 8*a^2*b^2*c*d*f*i^2 - 2*a^
2*b*c^2*d*g^2*h + 2*a*b^2*c^2*e^2*f*h - 4*a*b^2*c^2*d*f^2*h - 2*a^2*b*c^2*d
*f*h^2 + 2*a*b^2*c^2*d*f*g^2 + 8*a^3*c^2*e*f*h*i - 8*a^3*c^2*d*g*h*i + 8*a^
2*c^3*d*e*g*h - 8*a^2*c^3*d*e*f*i - 2*a^3*b*c*e*h^2*i + 6*a^3*b*c*d*h*i^2 -
2*a^3*b*c*e*g*i^2 + 2*a*b^3*c*e^2*g*i + 6*a*b*c^3*d^2*e*i + 2*a*b^3*c*d*f*

h^2 - 2*a*b*c^3*d^2*f*h - 2*a*b*c^3*d*e^2*h + 4*a^2*b^2*c*e^2*i^2 - 5*a^2*b
*c^2*d^2*i^2 + 3*a^2*b*c^2*e^2*h^2 + 4*a*b^2*c^2*d^2*h^2 - 4*a^3*c^2*f^2*g*
i + 2*a^3*b^2*f*h*i^2 + 4*a^3*c^2*f*g^2*h + 4*a^3*c^2*e*g^2*i - 4*a^3*c^2*e
*g*h^2 + 4*a^2*c^3*d^2*g*i + 2*a^2*b^3*e*g*i^2 - 2*a^2*b^3*d*h*i^2 + 4*a^3*
c^2*d*f*i^2 - 4*a^2*c^3*e^2*f*h + 2*b^3*c^2*d^2*f*h - 2*b^3*c^2*d^2*e*i + 4
*a^2*c^3*e*f^2*g + 4*a^2*c^3*d*f^2*h - 4*a^2*c^3*d*f*g^2 + 3*a^3*b*c*f^2*i^
2 + 2*b^2*c^3*d^2*e*g + 2*a^2*b*c^2*f^3*h - 2*a*b^2*c^2*e^3*i + 5*a*b^3*c*d
^2*i^2 - 2*a^2*b^2*c*d*h^3 + 2*a^2*b*c^2*e*g^3 + 3*a*b*c^3*d^2*g^2 + 4*a^4*
c*g*h^2*i - 4*a^4*c*f*h*i^2 + 2*b^4*c*d^2*g*i + 2*a^3*b*c*g^3*i + 2*a*b^4*d
*f*i^2 - 4*a*c^4*d^2*e*g + 2*a^3*b*c*f*h^3 + 4*a*c^4*d*e^2*f + 2*a*b*c^3*e^
3*g + 2*a*b*c^3*d*f^3 - a^2*b^2*c*f^2*h^2 - a^2*b*c^2*f^2*g^2 - a*b^2*c^2*e
^2*g^2 + 2*a^4*b*g*i^3 + 4*a^4*c*e*i^3 + 4*a*c^4*d^3*h + 2*b*c^4*d^3*f - a^
3*b*c*g^2*h^2 - a*b^3*c*e^2*h^2 - 6*a^3*c^2*e^2*i^2 - 2*a^3*c^2*f^2*h^2 - a
*b*c^3*e^2*f^2 - 6*a^2*c^3*d^2*h^2 - 2*a^2*c^3*e^2*g^2 - 2*a^4*c*g^2*i^2 +
4*a^2*c^3*e^3*i - 2*b^2*c^3*d^3*h - 2*a^3*b^2*e*i^3 + 4*a^3*c^2*d*h^3 - 2*a
*c^4*d^2*f^2 - a^3*b^2*g^2*i^2 - a^2*b^3*f^2*i^2 - b^3*c^2*d^2*g^2 - b^2*c^
3*d^2*f^2 - a^4*b*h^2*i^2 - b^4*c*d^2*h^2 - a*b^4*e^2*i^2 - b*c^4*d^2*e^2 -
b^5*d^2*i^2 - a^3*c^2*g^4 - a^2*c^3*f^4 - a^4*c*h^4 - a*c^4*e^4 - a^5*i^4
- c^5*d^4, z, l)*((x*(4*b^2*c^4*e - 8*b^3*c^3*g + 16*a^2*c^4*i + 8*b^4*c^2*
i - 16*a*c^5*e + 32*a*b*c^4*g - 36*a*b^2*c^3*i))/c^2 - (4*b^2*c^4*d + 16*a^
2*c^4*h - 16*a*c^5*d - 4*a*b^2*c^3*h)/c^2 + (root(128*a^2*b^2*c^5*z^4 - 16*
a*b^4*c^4*z^4 - 256*a^3*c^6*z^4 + 128*a^2*b^3*c^3*i*z^3 - 128*a^2*b^2*c^4*g
*z^3 - 256*a^3*b*c^4*i*z^3 - 16*a*b^5*c^2*i*z^3 + 16*a*b^4*c^3*g*z^3 + 256*
a^3*c^5*g*z^3 + 160*a^3*b*c^3*g*i*z^2 + 8*a*b^4*c^2*f*h*z^2 + 8*a*b^4*c^2*e
*i*z^2 + 32*a^2*b*c^4*e*g*z^2 + 32*a^2*b*c^4*d*h*z^2 - 8*a*b^3*c^3*e*g*z^2
- 8*a*b^3*c^3*d*h*z^2 + 16*a*b^2*c^4*d*f*z^2 + 8*a*b^5*c*g*i*z^2 - 72*a^2*b
^3*c^2*g*i*z^2 - 48*a^2*b^2*c^3*f*h*z^2 - 48*a^2*b^2*c^3*e*i*z^2 + 32*a^2*b
^4*c*i^2*z^2 - 48*a^3*b*c^3*h^2*z^2 - 4*a*b^4*c^2*g^2*z^2 + 16*a^2*b*c^4*f^
2*z^2 - 4*a*b^3*c^3*f^2*z^2 + 8*a*b^2*c^4*e^2*z^2 + 64*a^3*c^4*f*h*z^2 + 64
*a^3*c^4*e*i*z^2 - 64*a^2*c^5*d*f*z^2 - 4*a*b^5*c*h^2*z^2 + 16*a*b*c^5*d^2*
z^2 - 56*a^3*b^2*c^2*i^2*z^2 + 28*a^2*b^3*c^2*h^2*z^2 + 40*a^2*b^2*c^3*g^2*
z^2 - 32*a^4*c^3*i^2*z^2 - 96*a^3*c^4*g^2*z^2 - 32*a^2*c^5*e^2*z^2 - 4*b^3*
c^4*d^2*z^2 - 4*a*b^6*i^2*z^2 + 32*a^2*b*c^3*e*f*h*z - 32*a^2*b*c^3*d*f*i*z
- 8*a*b^3*c^2*e*f*h*z + 8*a*b^3*c^2*d*f*i*z - 8*a*b^2*c^3*d*f*g*z + 8*a*b^
2*c^3*d*e*h*z - 8*a*b^4*c*e*g*i*z + 40*a^2*b^2*c^2*e*g*i*z + 8*a^2*b^2*c^2*
f*g*h*z - 8*a^2*b^2*c^2*d*h*i*z + 4*a^3*b^2*c*h^2*i*z - 32*a^3*b*c^2*g^2*i*
z + 12*a^3*b^2*c*g*i^2*z + 8*a^2*b^3*c*g^2*i*z + 16*a^3*b*c^2*g*h^2*z - 4*a
^2*b^3*c*g*h^2*z + 32*a^3*b*c^2*e*i^2*z - 24*a^2*b^3*c*e*i^2*z - 16*a^2*b*c
^3*e^2*i*z + 4*a*b^3*c^2*e^2*i*z + 20*a*b^2*c^3*d^2*i*z - 16*a^2*b*c^3*e*g^
2*z + 4*a*b^3*c^2*e*g^2*z - 4*a*b^2*c^3*e^2*g*z + 4*a*b^2*c^3*e*f^2*z - 32*
a^3*c^3*f*g*h*z - 32*a^3*c^3*e*g*i*z + 32*a^3*c^3*d*h*i*z + 32*a^2*c^4*d*f*
g*z - 32*a^2*c^4*d*e*h*z + 4*a*b^4*c*e*h^2*z - 16*a*b*c^4*d^2*g*z - 4*a^2*b
^2*c^2*f^2*i*z - 20*a^2*b^2*c^2*e*h^2*z - 4*a^2*b^2*c^2*g^3*z - 16*a^4*c^2*
h^2*i*z + 16*a^4*c^2*g*i^2*z + 16*a^3*c^3*f^2*i*z - 4*a^2*b^4*g*i^2*z - 4*b
^4*c^2*d^2*i*z + 16*a^3*c^3*e*h^2*z - 16*a^2*c^4*d^2*i*z + 16*a^2*c^4*e^2*g

```
*z + 4*b^3*c^3*d^2*g*z - 16*a^2*c^4*e*f^2*z - 4*b^2*c^4*d^2*e*z + 4*a*b^5*e
*i^2*z - 16*a^4*b*c*i^3*z + 16*a*c^5*d^2*e*z + 4*a^3*b^3*i^3*z + 16*a^3*c^3
*g^3*z + 4*a^2*b^2*c*d*g*h*i + 12*a^2*b*c^2*d*f*g*i - 4*a^2*b*c^2*e*f*g*h -
 4*a^2*b*c^2*d*e*h*i + 4*a*b^2*c^2*d*e*f*i - 4*a^3*b*c*f*g*h*i - 4*a*b^3*c*
d*f*g*i - 4*a*b*c^3*d*e*f*g + 2*a^2*b^2*c*f^2*g*i - 4*a^2*b^2*c*e*g^2*i - 2
*a^2*b*c^2*e^2*g*i - 8*a*b^2*c^2*d^2*g*i + 2*a^2*b^2*c*e*g*h^2 - 2*a^2*b*c^
2*e*f^2*i - 8*a^2*b^2*c*d*f*i^2 - 2*a^2*b*c^2*d*g^2*h + 2*a*b^2*c^2*e^2*f*h
 - 4*a*b^2*c^2*d*f^2*h - 2*a^2*b*c^2*d*f*h^2 + 2*a*b^2*c^2*d*f*g^2 + 8*a^3*
c^2*e*f*h*i - 8*a^3*c^2*d*g*h*i + 8*a^2*c^3*d*e*g*h - 8*a^2*c^3*d*e*f*i - 2
*a^3*b*c*e*h^2*i + 6*a^3*b*c*d*h*i^2 - 2*a^3*b*c*e*g*i^2 + 2*a*b^3*c*e^2*g*
i + 6*a*b*c^3*d^2*e*i + 2*a*b^3*c*d*f*h^2 - 2*a*b*c^3*d^2*f*h - 2*a*b*c^3*d
*e^2*h + 4*a^2*b^2*c*e^2*i^2 - 5*a^2*b*c^2*d^2*i^2 + 3*a^2*b*c^2*e^2*h^2 +
4*a*b^2*c^2*d^2*h^2 - 4*a^3*c^2*f^2*g*i + 2*a^3*b^2*f*h*i^2 + 4*a^3*c^2*f*g
^2*h + 4*a^3*c^2*e*g^2*i - 4*a^3*c^2*e*g*h^2 + 4*a^2*c^3*d^2*g*i + 2*a^2*b^
3*e*g*i^2 - 2*a^2*b^3*d*h*i^2 + 4*a^3*c^2*d*f*i^2 - 4*a^2*c^3*e^2*f*h + 2*b
^3*c^2*d^2*f*h - 2*b^3*c^2*d^2*e*i + 4*a^2*c^3*e*f^2*g + 4*a^2*c^3*d*f^2*h
- 4*a^2*c^3*d*f*g^2 + 3*a^3*b*c*f^2*i^2 + 2*b^2*c^3*d^2*e*g + 2*a^2*b*c^2*f
^3*h - 2*a*b^2*c^2*e^3*i + 5*a*b^3*c*d^2*i^2 - 2*a^2*b^2*c*d*h^3 + 2*a^2*b*
c^2*e*g^3 + 3*a*b*c^3*d^2*g^2 + 4*a^4*c*g*h^2*i - 4*a^4*c*f*h*i^2 + 2*b^4*c
*d^2*g*i + 2*a^3*b*c*g^3*i + 2*a*b^4*d*f*i^2 - 4*a*c^4*d^2*e*g + 2*a^3*b*c*
f*h^3 + 4*a*c^4*d*e^2*f + 2*a*b*c^3*e^3*g + 2*a*b*c^3*d*f^3 - a^2*b^2*c*f^2
*h^2 - a^2*b*c^2*f^2*g^2 - a*b^2*c^2*e^2*g^2 + 2*a^4*b*g*i^3 + 4*a^4*c*e*i^
3 + 4*a*c^4*d^3*h + 2*b*c^4*d^3*f - a^3*b*c*g^2*h^2 - a*b^3*c*e^2*h^2 - 6*a
^3*c^2*e^2*i^2 - 2*a^3*c^2*f^2*h^2 - a*b*c^3*e^2*f^2 - 6*a^2*c^3*d^2*h^2 -
2*a^2*c^3*e^2*g^2 - 2*a^4*c*g^2*i^2 + 4*a^2*c^3*e^3*i - 2*b^2*c^3*d^3*h - 2
*a^3*b^2*e*i^3 + 4*a^3*c^2*d*h^3 - 2*a*c^4*d^2*f^2 - a^3*b^2*g^2*i^2 - a^2*
b^3*f^2*i^2 - b^3*c^2*d^2*g^2 - b^2*c^3*d^2*f^2 - a^4*b*h^2*i^2 - b^4*c*d^2
*h^2 - a*b^4*e^2*i^2 - b*c^4*d^2*e^2 - b^5*d^2*i^2 - a^3*c^2*g^4 - a^2*c^3*
f^4 - a^4*c*h^4 - a*c^4*e^4 - a^5*i^4 - c^5*d^4, z, l)*x*(8*b^3*c^4 - 32*a*
b*c^5))/c^2) - (4*b*c^4*d*e + 8*a*c^4*d*g - 8*a*c^4*e*f - 4*b^2*c^3*d*g + 4
*b^3*c^2*d*i + 8*a^2*c^3*f*i - 8*a^2*c^3*g*h - 4*a*b^2*c^2*f*i + 4*a^2*b*c^
2*h*i - 12*a*b*c^3*d*i + 4*a*b*c^3*e*h + 4*a*b*c^3*f*g)/c^2 + (x*(4*c^5*d^2
 + 2*b^5*i^2 - 4*a*c^4*f^2 - 2*b*c^4*e^2 + 2*b^4*c*h^2 + 2*b^2*c^3*f^2 + 4*
a^2*c^3*h^2 + 2*b^3*c^2*g^2 - 8*a*b^2*c^2*h^2 + 6*a^2*b*c^2*i^2 - 4*b*c^4*d
*f - 8*a*c^4*d*h + 8*a*c^4*e*g - 4*b^4*c*g*i - 10*a*b*c^3*g^2 - 10*a*b^3*c*
i^2 + 4*b^2*c^3*d*h - 4*b^3*c^2*f*h - 8*a^2*c^3*g*i + 20*a*b^2*c^2*g*i - 4*
a*b*c^3*e*i + 12*a*b*c^3*f*h))/c^2))*root(128*a^2*b^2*c^5*z^4 - 16*a*b^4*c^
4*z^4 - 256*a^3*c^6*z^4 + 128*a^2*b^3*c^3*i*z^3 - 128*a^2*b^2*c^4*g*z^3 - 2
56*a^3*b*c^4*i*z^3 - 16*a*b^5*c^2*i*z^3 + 16*a*b^4*c^3*g*z^3 + 256*a^3*c^5*
g*z^3 + 160*a^3*b*c^3*g*i*z^2 + 8*a*b^4*c^2*f*h*z^2 + 8*a*b^4*c^2*e*i*z^2 +
 32*a^2*b*c^4*e*g*z^2 + 32*a^2*b*c^4*d*h*z^2 - 8*a*b^3*c^3*e*g*z^2 - 8*a*b^
3*c^3*d*h*z^2 + 16*a*b^2*c^4*d*f*z^2 + 8*a*b^5*c*g*i*z^2 - 72*a^2*b^3*c^2*g
*i*z^2 - 48*a^2*b^2*c^3*f*h*z^2 - 48*a^2*b^2*c^3*e*i*z^2 + 32*a^2*b^4*c*i^2
*z^2 - 48*a^3*b*c^3*h^2*z^2 - 4*a*b^4*c^2*g^2*z^2 + 16*a^2*b*c^4*f^2*z^2 -
4*a*b^3*c^3*f^2*z^2 + 8*a*b^2*c^4*e^2*z^2 + 64*a^3*c^4*f*h*z^2 + 64*a^3*c^4
```

```
*e*i*z^2 - 64*a^2*c^5*d*f*z^2 - 4*a*b^5*c*h^2*z^2 + 16*a*b*c^5*d^2*z^2 - 56
*a^3*b^2*c^2*i^2*z^2 + 28*a^2*b^3*c^2*h^2*z^2 + 40*a^2*b^2*c^3*g^2*z^2 - 32
*a^4*c^3*i^2*z^2 - 96*a^3*c^4*g^2*z^2 - 32*a^2*c^5*e^2*z^2 - 4*b^3*c^4*d^2*
z^2 - 4*a*b^6*i^2*z^2 + 32*a^2*b*c^3*e*f*h*z - 32*a^2*b*c^3*d*f*i*z - 8*a*b
^3*c^2*e*f*h*z + 8*a*b^3*c^2*d*f*i*z - 8*a*b^2*c^3*d*f*g*z + 8*a*b^2*c^3*d*
e*h*z - 8*a*b^4*c*e*g*i*z + 40*a^2*b^2*c^2*e*g*i*z + 8*a^2*b^2*c^2*f*g*h*z
- 8*a^2*b^2*c^2*d*h*i*z + 4*a^3*b^2*c*h^2*i*z - 32*a^3*b*c^2*g^2*i*z + 12*a
^3*b^2*c*g*i^2*z + 8*a^2*b^3*c*g^2*i*z + 16*a^3*b*c^2*g*h^2*z - 4*a^2*b^3*c
*g*h^2*z + 32*a^3*b*c^2*e*i^2*z - 24*a^2*b^3*c*e*i^2*z - 16*a^2*b*c^3*e^2*i
*z + 4*a*b^3*c^2*e^2*i*z + 20*a*b^2*c^3*d^2*i*z - 16*a^2*b*c^3*e*g^2*z + 4*
a*b^3*c^2*e*g^2*z - 4*a*b^2*c^3*e^2*g*z + 4*a*b^2*c^3*e*f^2*z - 32*a^3*c^3*
f*g*h*z - 32*a^3*c^3*e*g*i*z + 32*a^3*c^3*d*h*i*z + 32*a^2*c^4*d*f*g*z - 32
*a^2*c^4*d*e*h*z + 4*a*b^4*c*e*h^2*z - 16*a*b*c^4*d^2*g*z - 4*a^2*b^2*c^2*f
^2*i*z - 20*a^2*b^2*c^2*e*h^2*z - 4*a^2*b^2*c^2*g^3*z - 16*a^4*c^2*h^2*i*z
+ 16*a^4*c^2*g*i^2*z + 16*a^3*c^3*f^2*i*z - 4*a^2*b^4*g*i^2*z - 4*b^4*c^2*d
^2*i*z + 16*a^3*c^3*e*h^2*z - 16*a^2*c^4*d^2*i*z + 16*a^2*c^4*e^2*g*z + 4*b
^3*c^3*d^2*g*z - 16*a^2*c^4*e*f^2*z - 4*b^2*c^4*d^2*e*z + 4*a*b^5*e*i^2*z -
 16*a^4*b*c*i^3*z + 16*a*c^5*d^2*e*z + 4*a^3*b^3*i^3*z + 16*a^3*c^3*g^3*z +
 4*a^2*b^2*c*d*g*h*i + 12*a^2*b*c^2*d*f*g*i - 4*a^2*b*c^2*e*f*g*h - 4*a^2*b
*c^2*d*e*h*i + 4*a*b^2*c^2*d*e*f*i - 4*a^3*b*c*f*g*h*i - 4*a*b^3*c*d*f*g*i
- 4*a*b*c^3*d*e*f*g + 2*a^2*b^2*c*f^2*g*i - 4*a^2*b^2*c*e*g^2*i - 2*a^2*b*c
^2*e^2*g*i - 8*a*b^2*c^2*d^2*g*i + 2*a^2*b^2*c*e*g*h^2 - 2*a^2*b*c^2*e*f^2*
i - 8*a^2*b^2*c*d*f*i^2 - 2*a^2*b*c^2*d*g^2*h + 2*a*b^2*c^2*e^2*f*h - 4*a*b
^2*c^2*d*f^2*h - 2*a^2*b*c^2*d*f*h^2 + 2*a*b^2*c^2*d*f*g^2 + 8*a^3*c^2*e*f*
h*i - 8*a^3*c^2*d*g*h*i + 8*a^2*c^3*d*e*g*h - 8*a^2*c^3*d*e*f*i - 2*a^3*b*c
*e*h^2*i + 6*a^3*b*c*d*h*i^2 - 2*a^3*b*c*e*g*i^2 + 2*a*b^3*c*e^2*g*i + 6*a*
b*c^3*d^2*e*i + 2*a*b^3*c*d*f*h^2 - 2*a*b*c^3*d^2*f*h - 2*a*b*c^3*d*e^2*h +
 4*a^2*b^2*c*e^2*i^2 - 5*a^2*b*c^2*d^2*i^2 + 3*a^2*b*c^2*e^2*h^2 + 4*a*b^2*
c^2*d^2*h^2 - 4*a^3*c^2*f^2*g*i + 2*a^3*b^2*f*h*i^2 + 4*a^3*c^2*f*g^2*h + 4
*a^3*c^2*e*g^2*i - 4*a^3*c^2*e*g*h^2 + 4*a^2*c^3*d^2*g*i + 2*a^2*b^3*e*g*i^
2 - 2*a^2*b^3*d*h*i^2 + 4*a^3*c^2*d*f*i^2 - 4*a^2*c^3*e^2*f*h + 2*b^3*c^2*d
^2*f*h - 2*b^3*c^2*d^2*e*i + 4*a^2*c^3*e*f^2*g + 4*a^2*c^3*d*f^2*h - 4*a^2*
c^3*d*f*g^2 + 3*a^3*b*c*f^2*i^2 + 2*b^2*c^3*d^2*e*g + 2*a^2*b*c^2*f^3*h - 2
*a*b^2*c^2*e^3*i + 5*a*b^3*c*d^2*i^2 - 2*a^2*b^2*c*d*h^3 + 2*a^2*b*c^2*e*g^
3 + 3*a*b*c^3*d^2*g^2 + 4*a^4*c*g*h^2*i - 4*a^4*c*f*h*i^2 + 2*b^4*c*d^2*g*i
 + 2*a^3*b*c*g^3*i + 2*a*b^4*d*f*i^2 - 4*a*c^4*d^2*e*g + 2*a^3*b*c*f*h^3 +
4*a*c^4*d*e^2*f + 2*a*b*c^3*e^3*g + 2*a*b*c^3*d*f^3 - a^2*b^2*c*f^2*h^2 - a
^2*b*c^2*f^2*g^2 - a*b^2*c^2*e^2*g^2 + 2*a^4*b*g*i^3 + 4*a^4*c*e*i^3 + 4*a*
c^4*d^3*h + 2*b*c^4*d^3*f - a^3*b*c*g^2*h^2 - a*b^3*c*e^2*h^2 - 6*a^3*c^2*e
^2*i^2 - 2*a^3*c^2*f^2*h^2 - a*b*c^3*e^2*f^2 - 6*a^2*c^3*d^2*h^2 - 2*a^2*c^
3*e^2*g^2 - 2*a^4*c*g^2*i^2 + 4*a^2*c^3*e^3*i - 2*b^2*c^3*d^3*h - 2*a^3*b^2
*e*i^3 + 4*a^3*c^2*d*h^3 - 2*a*c^4*d^2*f^2 - a^3*b^2*g^2*i^2 - a^2*b^3*f^2*
i^2 - b^3*c^2*d^2*g^2 - b^2*c^3*d^2*f^2 - a^4*b*h^2*i^2 - b^4*c*d^2*h^2 - a
*b^4*e^2*i^2 - b*c^4*d^2*e^2 - b^5*d^2*i^2 - a^3*c^2*g^4 - a^2*c^3*f^4 - a^
4*c*h^4 - a*c^4*e^4 - a^5*i^4 - c^5*d^4, z, l), l, 1, 4) + (h*x)/c + (i*x^2
```

```
)/(2*c)
```

3.25 $\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4} \, dx$

Optimal result

Integrand size = 55, antiderivative size = 545

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4} \, dx \\
&= \frac{(c^2h+b^2m-c(bk+am))\,x}{c^3} + \frac{(cj-bl)x^2}{2c^2} + \frac{(ck-bm)x^3}{3c^2} + \frac{lx^4}{4c} + \frac{mx^5}{5c} \\
&\quad + \frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)+\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\arctan\Big(}{\sqrt{2}c^{7/2}\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)-\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\arctan\Big(}{\sqrt{2}c^{7/2}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad - \frac{(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^3\sqrt{b^2-4ac}} \\
&\quad + \frac{(c^2g+b^2l-c(bj+al))\log\left(a+bx^2+cx^4\right)}{4c^3}
\end{aligned}
$$

```
[Out] (c^2*h+b^2*m-c*(a*m+b*k))*x/c^3+1/2*(-b*l+c*j)*x^2/c^2+1/3*(-b*m+c*k)*x^3/c
^2+1/4*l*x^4/c+1/5*m*x^5/c+1/4*(c^2*g+b^2*l-c*(a*l+b*j))*ln(c*x^4+b*x^2+a)/
c^3-1/2*(2*c^3*e-c^2*(2*a*j+b*g)-b^3*l+b*c*(3*a*l+b*j))*arctanh((2*c*x^2+b)
/(-4*a*c+b^2)^(1/2))/c^3/(-4*a*c+b^2)^(1/2)+1/2*arctan(x*2^(1/2)*c^(1/2)/(b
-(-4*a*c+b^2)^(1/2))^(1/2))*(c^3*f-c^2*(a*k+b*h)-b^3*m+b*c*(2*a*m+b*k)+(2*c
^4*d-c^3*(2*a*h+b*f)+b^4*m-b^2*c*(4*a*m+b*k)+c^2*(2*a^2*m+3*a*b*k+b^2*h))/(
-4*a*c+b^2)^(1/2))/c^(7/2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/2*arctan(
x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*(c^3*f-c^2*(a*k+b*h)-b^3*m+
b*c*(2*a*m+b*k)+(-2*c^4*d+c^3*(2*a*h+b*f)-b^4*m+b^2*c*(4*a*m+b*k)-c^2*(2*a^
2*m+3*a*b*k+b^2*h))/(-4*a*c+b^2)^(1/2))/c^(7/2)*2^(1/2)/(b+(-4*a*c+b^2)^(1/
2))^(1/2)
```

Rubi [A] (verified)

Time = 2.85 (sec) , antiderivative size = 545, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.182, Rules used = {1687, 1690, 1180, 211, 1677, 1671, 648, 632, 212, 642}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx$$
$$=\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{c^2(2a^2m+3abk+b^2h)-b^2c(4am+bk)-c^3(2ah+bf)+b^4m+2c^4d}{\sqrt{b^2-4ac}}-c^2(ak+bh)+bc(2am+bk)+b^3(\right.}{\sqrt{2}c^{7/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{c^2(2a^2m+3abk+b^2h)-b^2c(4am+bk)-c^3(2ah+bf)+b^4m+2c^4d}{\sqrt{b^2-4ac}}-c^2(ak+bh)+bc(2am+bk)+\right.}{\sqrt{2}c^{7/2}\sqrt{\sqrt{b^2-4ac}+b}}$$
$$-\frac{\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)\left(-c^2(2aj+bg)+bc(3al+bj)+b^3(-l)+2c^3e\right)}{2c^3\sqrt{b^2-4ac}}$$
$$+\frac{\log\left(a+bx^2+cx^4\right)\left(-c(al+bj)+b^2l+c^2g\right)}{4c^3}$$
$$+\frac{x(-c(am+bk)+b^2m+c^2h)}{c^3}+\frac{x^2(cj-bl)}{2c^2}+\frac{x^3(ck-bm)}{3c^2}+\frac{lx^4}{4c}+\frac{mx^5}{5c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4),x]
```

```
[Out] ((c^2*h + b^2*m - c*(b*k + a*m))*x)/c^3 + ((c*j - b*l)*x^2)/(2*c^2) + ((c*k
 - b*m)*x^3)/(3*c^2) + (l*x^4)/(4*c) + (m*x^5)/(5*c) + ((c^3*f - c^2*(b*h +
 a*k) - b^3*m + b*c*(b*k + 2*a*m) + (2*c^4*d - c^3*(b*f + 2*a*h) + b^4*m -
b^2*c*(b*k + 4*a*m) + c^2*(b^2*h + 3*a*b*k + 2*a^2*m))/Sqrt[b^2 - 4*a*c])*A
rcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(7/2)*Sq
rt[b - Sqrt[b^2 - 4*a*c]]) + ((c^3*f - c^2*(b*h + a*k) - b^3*m + b*c*(b*k +
 2*a*m) - (2*c^4*d - c^3*(b*f + 2*a*h) + b^4*m - b^2*c*(b*k + 4*a*m) + c^2*
(b^2*h + 3*a*b*k + 2*a^2*m))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/
Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(7/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]])
 - ((2*c^3*e - c^2*(b*g + 2*a*j) - b^3*l + b*c*(b*j + 3*a*l))*ArcTanh[(b +
2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(2*c^3*Sqrt[b^2 - 4*a*c]) + ((c^2*g + b^2*l -
c*(b*j + a*l))*Log[a + b*x^2 + c*x^4])/(4*c^3)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
```

```
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
```

```
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2+jx^4+lx^6)}{a+bx^2+cx^4}\,dx + \int \frac{d+fx^2+hx^4+kx^6+mx^8}{a+bx^2+cx^4}\,dx \\
&= \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+jx^2+lx^3}{a+bx+cx^2}\,dx, x, x^2\right) \\
&\quad + \int \left(\frac{c^2h+b^2m-c(bk+am)}{c^3} + \frac{(ck-bm)x^2}{c^2} + \frac{mx^4}{c} \right. \\
&\quad \left. + \frac{c^3d-ac^2h-ab^2m+ac(bk+am)+\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)\right)x^2}{c^3\left(a+bx^2+cx^4\right)}\right) dx \\
&= \frac{\left(c^2h+b^2m-c(bk+am)\right)x}{c^3} + \frac{(ck-bm)x^3}{3c^2} + \frac{mx^5}{5c} + \frac{1}{2}\text{Subst}\left(\int \left(\frac{cj-bl}{c^2} + \frac{lx}{c} \right.\right. \\
&\quad \left.\left. + \frac{c^2e-acj+abl+\left(c^2g+b^2l-c(bj+al)\right)x}{c^2\left(a+bx+cx^2\right)}\right) dx, x, x^2\right) \\
&\quad + \frac{\int \frac{c^3d-ac^2h-ab^2m+ac(bk+am)+\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)\right)x^2}{a+bx^2+cx^4}\,dx}{c^3} \\
&= \frac{\left(c^2h+b^2m-c(bk+am)\right)x}{c^3} + \frac{(cj-bl)x^2}{2c^2} + \frac{(ck-bm)x^3}{3c^2} \\
&\quad + \frac{lx^4}{4c} + \frac{mx^5}{5c} + \frac{\text{Subst}\left(\int \frac{c^2e-acj+abl+\left(c^2g+b^2l-c(bj+al)\right)x}{a+bx+cx^2}\,dx, x, x^2\right)}{2c^2} \\
&\quad + \frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)-\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2\left(b^2h+3abk+2a^2m\right)}{\sqrt{b^2-4ac}}\right)\int}{2c^3} \\
&\quad + \frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)+\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2\left(b^2h+3abk+2a^2m\right)}{\sqrt{b^2-4ac}}\right)\int}{2c^3}
\end{aligned}$$

$$= \frac{(c^2h+b^2m-c(bk+am))\,x}{c^3}+\frac{(cj-bl)x^2}{2c^2}+\frac{(ck-bm)x^3}{3c^2}+\frac{lx^4}{4c}+\frac{mx^5}{5c}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)+\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)-\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+\frac{(c^2g+b^2l-c(bj+al))\,\mathrm{Subst}\left(\int\frac{b+2cx}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^3}$$
$$+\frac{(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))\,\mathrm{Subst}\left(\int\frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^3}$$

$$= \frac{(c^2h+b^2m-c(bk+am))\,x}{c^3}+\frac{(cj-bl)x^2}{2c^2}+\frac{(ck-bm)x^3}{3c^2}+\frac{lx^4}{4c}+\frac{mx^5}{5c}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)+\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)-\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+\frac{(c^2g+b^2l-c(bj+al))\log\left(a+bx^2+cx^4\right)}{4c^3}$$
$$-\frac{(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))\,\mathrm{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{2c^3}$$

$$= \frac{(c^2h+b^2m-c(bk+am))\,x}{c^3}+\frac{(cj-bl)x^2}{2c^2}+\frac{(ck-bm)x^3}{3c^2}+\frac{lx^4}{4c}+\frac{mx^5}{5c}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)+\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\left(c^3f-c^2(bh+ak)-b^3m+bc(bk+2am)-\frac{2c^4d-c^3(bf+2ah)+b^4m-b^2c(bk+4am)+c^2(b^2h+3abk+2a^2m)}{\sqrt{b^2-4ac}}\right)\mathrm{t}}{\sqrt{2}c^{7/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$-\frac{(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^3\sqrt{b^2-4ac}}$$
$$+\frac{(c^2g+b^2l-c(bj+al))\log\left(a+bx^2+cx^4\right)}{4c^3}$$

Mathematica [A] (verified)

Time = 0.73 (sec) , antiderivative size = 816, normalized size of antiderivative = 1.50

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx$$
$$= \frac{(c^2h+b^2m-c(bk+am))\,x}{c^3}+\frac{(cj-bl)x^2}{2c^2}+\frac{(ck-bm)x^3}{3c^2}+\frac{lx^4}{4c}+\frac{mx^5}{5c}$$
$$+\frac{\left(2c^4d+c^3\left(-bf+\sqrt{b^2-4ac}f-2ah\right)+b^3\left(b-\sqrt{b^2-4ac}\right)m+c^2\left(b^2h-b\sqrt{b^2-4ac}h+3abk-a\sqrt{b^2-\cdots}\right.\right.}{\sqrt{2}c^{7/2}\sqrt{b^2-4ac}\sqrt{b-\cdots}}$$
$$-\frac{\left(2c^4d-c^3\left(bf+\sqrt{b^2-4ac}f+2ah\right)+b^3\left(b+\sqrt{b^2-4ac}\right)m+c^2\left(b^2h+b\sqrt{b^2-4ac}h+3abk+a\sqrt{b^2-\cdots}\right.\right.}{\sqrt{2}c^{7/2}\sqrt{b^2-4ac}\sqrt{b+\cdots}}$$
$$+\frac{\left(2c^3e+c^2\left(-bg+\sqrt{b^2-4ac}g-2aj\right)+b^2\left(-b+\sqrt{b^2-4ac}\right)l+c\left(b^2j-b\sqrt{b^2-4ac}j+3abl-a\sqrt{b^2-\cdots}\right.\right.}{4c^3\sqrt{b^2-4ac}}$$
$$+\frac{\left(-2c^3e+c^2\left(bg+\sqrt{b^2-4ac}g+2aj\right)+b^2\left(b+\sqrt{b^2-4ac}\right)l-c\left(b^2j+b\sqrt{b^2-4ac}j+3abl+a\sqrt{b^2-4a\cdots}\right.\right.}{4c^3\sqrt{b^2-4ac}}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)
/(a + b*x^2 + c*x^4),x]
```

```
[Out] ((c^2*h + b^2*m - c*(b*k + a*m))*x)/c^3 + ((c*j - b*l)*x^2)/(2*c^2) + ((c*k
 - b*m)*x^3)/(3*c^2) + (l*x^4)/(4*c) + (m*x^5)/(5*c) + ((2*c^4*d + c^3*(-(b
*f) + Sqrt[b^2 - 4*a*c]*f - 2*a*h) + b^3*(b - Sqrt[b^2 - 4*a*c])*m + c^2*(b
^2*h - b*Sqrt[b^2 - 4*a*c]*h + 3*a*b*k - a*Sqrt[b^2 - 4*a*c]*k + 2*a^2*m) +
 b*c*(-(b^2*k) + b*Sqrt[b^2 - 4*a*c]*k - 4*a*b*m + 2*a*Sqrt[b^2 - 4*a*c]*m)
)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(7/2)
*Sqrt[b^2 - 4*a*c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) - ((2*c^4*d - c^3*(b*f + Sq
rt[b^2 - 4*a*c]*f + 2*a*h) + b^3*(b + Sqrt[b^2 - 4*a*c])*m + c^2*(b^2*h + b
*Sqrt[b^2 - 4*a*c]*h + 3*a*b*k + a*Sqrt[b^2 - 4*a*c]*k + 2*a^2*m) - b*c*(b^
2*k + b*Sqrt[b^2 - 4*a*c]*k + 4*a*b*m + 2*a*Sqrt[b^2 - 4*a*c]*m))*ArcTan[(S
qrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*c^(7/2)*Sqrt[b^2 -
 4*a*c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((2*c^3*e + c^2*(-(b*g) + Sqrt[b^2 -
 4*a*c]*g - 2*a*j) + b^2*(-b + Sqrt[b^2 - 4*a*c])*l + c*(b^2*j - b*Sqrt[b^2
 - 4*a*c]*j + 3*a*b*l - a*Sqrt[b^2 - 4*a*c]*l))*Log[-b + Sqrt[b^2 - 4*a*c]
- 2*c*x^2])/(4*c^3*Sqrt[b^2 - 4*a*c]) + ((-2*c^3*e + c^2*(b*g + Sqrt[b^2 -
4*a*c]*g + 2*a*j) + b^2*(b + Sqrt[b^2 - 4*a*c])*l - c*(b^2*j + b*Sqrt[b^2 -
 4*a*c]*j + 3*a*b*l + a*Sqrt[b^2 - 4*a*c]*l))*Log[b + Sqrt[b^2 - 4*a*c] + 2
*c*x^2])/(4*c^3*Sqrt[b^2 - 4*a*c])
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.76 (sec) , antiderivative size = 246, normalized size of antiderivative = 0.45

method	result
risch	$\frac{m\,x^{5}}{5c}+\frac{l\,x^{4}}{4c}-\frac{bm\,x^{3}}{3c^{2}}+\frac{k\,x^{3}}{3c}-\frac{bl\,x^{2}}{2c^{2}}+\frac{j\,x^{2}}{2c}-\frac{amx}{c^{2}}+\frac{b^{2}mx}{c^{3}}-\frac{bkx}{c^{2}}+\frac{hx}{c}+\frac{\sum_{\_R=\mathrm{RootOf}\left(c\_Z^{4}+\_Z^{2}b+a\right)}\frac{\left(c\left(-acl+b^{2}\cdots\right.\right.}{\cdots}}{\cdots}$
default	$-\frac{-\frac{1}{5}m\,x^{5}c^{2}-\frac{1}{4}l\,x^{4}c^{2}+\frac{1}{3}bcm\,x^{3}-\frac{1}{3}c^{2}k\,x^{3}+\frac{1}{2}bcl\,x^{2}-\frac{1}{2}c^{2}j\,x^{2}+acmx-b^{2}mx+bckx-c^{2}hx}{c^{3}}+\frac{-\frac{\sqrt{-4ac+b^{2}}\left(\frac{\left(-\sqrt{-4ac+b^{2}}\,a\,c^{2}l+\sqrt{-\cdots}\right.}{\cdots}\right.}{\cdots}}{\cdots}$

```
[In] int((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/5*m*x^5/c+1/4*l*x^4/c-1/3/c^2*b*m*x^3+1/3/c*k*x^3-1/2/c^2*b*l*x^2+1/2/c*j
*x^2-1/c^2*a*m*x+1/c^3*b^2*m*x-1/c^2*b*k*x+h*x/c+1/2/c^3*sum((c*(-a*c*l+b^2
*l-b*c*j+c^2*g)*_R^3+_R^2*(2*a*b*c*m-a*c^2*k-b^3*m+b^2*c*k-b*c^2*h+c^3*f)+c
*(a*b*l-a*c*j+c^2*e)*_R+a^2*c*m-a*b^2*m+a*b*c*k-a*c^2*h+c^3*d)/(2*_R^3*c+_R
*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx=\text{Timed out}$$

```
[In] integrate((m*x**8+l*x**7+k*x**6+j*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+
b*x**2+a),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx$$
$$=\int \frac{mx^8+lx^7+kx^6+jx^5+hx^4+gx^3+fx^2+ex+d}{cx^4+bx^2+a}\,dx$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
,x, algorithm="maxima")
[Out] 1/60*(12*c^2*m*x^5 + 15*c^2*l*x^4 + 20*(c^2*k - b*c*m)*x^3 + 30*(c^2*j - b*
c*l)*x^2 + 60*(c^2*h - b*c*k + (b^2 - a*c)*m)*x)/c^3 - integrate(-(c^3*d -
a*c^2*h + a*b*c*k + (c^3*g - b*c^2*j + (b^2*c - a*c^2)*l)*x^3 + (c^3*f - b*
c^2*h + (b^2*c - a*c^2)*k - (b^3 - 2*a*b*c)*m)*x^2 - (a*b^2 - a^2*c)*m + (c
^3*e - a*c^2*j + a*b*c*l)*x)/(c*x^4 + b*x^2 + a), x)/c^3
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 11830 vs. $2(495)=990$.

Time = 2.10 (sec) , antiderivative size = 11830, normalized size of antiderivative = 21.71

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx=\text{Too large to display}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
,x, algorithm="giac")
[Out] 1/8*((2*b^4*c^5 - 16*a*b^2*c^6 + 32*a^2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^3*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^2*c^5 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
```

- 4*a*c)*c)*a*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*b^2*c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a*c^6 - 2*(b^2 - 4*a*c)*b^2*c^5 + 8*(b^2 - 4*a*c)*a*c^6)*c^2*f - (2*b^5*c^4
 - 16*a*b^3*c^5 + 32*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*b^5*c^2 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*b^4*c^3 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^2*b*c^4 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*
c^4 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^5 -
 2*(b^2 - 4*a*c)*b^3*c^4 + 8*(b^2 - 4*a*c)*a*b*c^5)*c^2*h + (2*b^6*c^3 - 18
*a*b^4*c^4 + 48*a^2*b^2*c^5 - 32*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*b^6*c + 9*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*b^5*c^2 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^2*b^2*c^3 - 10*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a*b^3*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b^4*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^3*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
2*b*c^4 + 5*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2
*c^4 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^5
- 2*(b^2 - 4*a*c)*b^4*c^3 + 10*(b^2 - 4*a*c)*a*b^2*c^4 - 8*(b^2 - 4*a*c)*a^
2*c^5)*c^2*k - (2*b^7*c^2 - 20*a*b^5*c^3 + 64*a^2*b^3*c^4 - 64*a^3*b*c^5 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7 + 10*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c - 32*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 12*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b^5*c^2 + 12*(b^2 - 4*a
*c)*a*b^3*c^3 - 16*(b^2 - 4*a*c)*a^2*b*c^4)*c^2*m + 2*(sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*b^4*c^5 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b
^2*c^6 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^6 - 2*b^4*c^6 + 16
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^7 + 8*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b*c^7 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^7
 + 16*a*b^2*c^7 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^8 - 32*a^2*
c^8 + 2*(b^2 - 4*a*c)*b^2*c^6 - 8*(b^2 - 4*a*c)*a*c^7)*d*abs(c) - 2*(sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^4 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^2*b^2*c^5 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*
c^5 - 2*a*b^4*c^5 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^6 + 8*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^6 + sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a*b^2*c^6 + 16*a^2*b^2*c^6 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2

- 4*a*c)*c)*a^2*c^7 - 32*a^3*c^7 + 2*(b^2 - 4*a*c)*a*b^2*c^5 - 8*(b^2 - 4*a*c)*a^2*c^6)*h*abs(c) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^3 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^4 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^4 - 2*a*b^5*c^4 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^5 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5 + 16*a^2*b^3*c^5 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^6 - 32*a^3*b*c^6 + 2*(b^2 - 4*a*c)*a*b^3*c^4 - 8*(b^2 - 4*a*c)*a^2*b*c^5)*k*abs(c) - 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c^2 - 9*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^3 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^3 - 2*a*b^6*c^3 + 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^4 + 10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^4 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^4 + 18*a^2*b^4*c^4 - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^5 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^5 - 5*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^5 - 48*a^3*b^2*c^5 + 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^6 + 32*a^4*c^6 + 2*(b^2 - 4*a*c)*a*b^4*c^3 - 10*(b^2 - 4*a*c)*a^2*b^2*c^4 + 8*(b^2 - 4*a*c)*a^3*c^5)*m*abs(c) + 2*(2*b^3*c^8 - 8*a*b*c^9 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^6 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^7 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^8 - 2*(b^2 - 4*a*c)*b*c^8)*d - (2*b^4*c^7 - 8*a*b^2*c^8 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c^7 - 2*(b^2 - 4*a*c)*b^2*c^7)*f + (2*b^5*c^6 - 12*a*b^3*c^7 + 16*a^2*b*c^8 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^4 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^5 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^6 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^6 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^7 - 2*(b^2 - 4*a*c)*b^3*c^6 + 4*(b^2 - 4*a*c)*a*b*c^7)*h - (2*b^6*c^5 - 14*a*b^4*c^6 + 24*a^2*b^2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c^3 + 7*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^4 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^5 - 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^5 + 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 - 2*(b^2 - 4*a*c)*b^4*c^5 + 6*(b^2 - 4*a*c)*a*b^2*c^6)*k + (2*b^7*c^4 - 16*a*b^5*c^5 + 36*a^2*b^3*c^6 - 16*a^3*b*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7*c^2 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*

```
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^4 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*b^5*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^3*b*c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^2*b^2*c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a*b^3*c^5 - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*b*c^6 - 2*(b^2 - 4*a*c)*b^5*c^4 + 8*(b^2 - 4*a*c)*a*b^3*c^5 -
 4*(b^2 - 4*a*c)*a^2*b*c^6)*m)*arctan(2*sqrt(1/2)*x/sqrt((b*c^11 + sqrt(b^2
*c^22 - 4*a*c^23))/c^12))/((a*b^4*c^5 - 8*a^2*b^2*c^6 - 2*a*b^3*c^6 + 16*a^
3*c^7 + 8*a^2*b*c^7 + a*b^2*c^7 - 4*a^2*c^8)*c^2) - 1/8*((2*b^4*c^5 - 16*a*
b^2*c^6 + 32*a^2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*b^4*c^3 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a*b^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^
3*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^
5 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^5 + 4*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^6 - 2*(b^2 - 4*a*c)
*b^2*c^5 + 8*(b^2 - 4*a*c)*a*c^6)*c^2*f - (2*b^5*c^4 - 16*a*b^3*c^5 + 32*a^
2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2
 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 +
2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 16*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 8*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 4*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^
4 + 8*(b^2 - 4*a*c)*a*b*c^5)*c^2*h + (2*b^6*c^3 - 18*a*b^4*c^4 + 48*a^2*b^2
*c^5 - 32*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*b^6*c + 9*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^
4*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2
 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3
 - 10*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 -
 sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 + 16*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 8*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 5*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 4*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^5 - 2*(b^2 - 4*a*c)*b^4*c
^3 + 10*(b^2 - 4*a*c)*a*b^2*c^4 - 8*(b^2 - 4*a*c)*a^2*c^5)*c^2*k - (2*b^7*c
^2 - 20*a*b^5*c^3 + 64*a^2*b^3*c^4 - 64*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7 + 10*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*b^6*c - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*b^5*c^2 + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
```

```
*c)*c)*a^3*b*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^2*c^3 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a*b^3*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b^5*c^2 + 12*(b^2 - 4*a*c)*a*b^3*c^3 - 16*(b^2
 - 4*a*c)*a^2*b*c^4)*c^2*m - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4
*c^5 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 - 2*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^6 + 2*b^4*c^6 + 16*sqrt(2)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^2*c^7 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c
^7 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^7 - 16*a*b^2*c^7 - 4*sqr
t(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^8 + 32*a^2*c^8 - 2*(b^2 - 4*a*c)*b
^2*c^6 + 8*(b^2 - 4*a*c)*a*c^7)*d*abs(c) + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*a*b^4*c^4 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^
5 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5 + 2*a*b^4*c^5 + 16*
sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^6 + 8*sqrt(2)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^2*b*c^6 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*
c^6 - 16*a^2*b^2*c^6 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^7 +
32*a^3*c^7 - 2*(b^2 - 4*a*c)*a*b^2*c^5 + 8*(b^2 - 4*a*c)*a^2*c^6)*h*abs(c)
- 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^3 - 8*sqrt(2)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a*b^4*c^4 + 2*a*b^5*c^4 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^3*b*c^5 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^5 + sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5 - 16*a^2*b^3*c^5 - 4*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^6 + 32*a^3*b*c^6 - 2*(b^2 - 4*a*c)*a*b^
3*c^4 + 8*(b^2 - 4*a*c)*a^2*b*c^5)*k*abs(c) + 2*(sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a*b^6*c^2 - 9*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4
*c^3 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^3 + 2*a*b^6*c^3 +
24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^4 + 10*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^4 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a*b^4*c^4 - 18*a^2*b^4*c^4 - 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^4*c^5 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^5 - 5*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^5 + 48*a^3*b^2*c^5 + 4*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^6 - 32*a^4*c^6 - 2*(b^2 - 4*a*c)*a*b^4*
c^3 + 10*(b^2 - 4*a*c)*a^2*b^2*c^4 - 8*(b^2 - 4*a*c)*a^3*c^5)*m*abs(c) + 2*
(2*b^3*c^8 - 8*a*b*c^9 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*b^3*c^6 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a*b*c^7 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^
2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^8 - 2
*(b^2 - 4*a*c)*b*c^8)*d - (2*b^4*c^7 - 8*a*b^2*c^8 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*b^2*c^7 - 2*(b^2 - 4*a*c)*b^2*c^7)*f + (2*b^5*c^6 - 12
*a*b^3*c^7 + 16*a^2*b*c^8 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*b^5*c^4 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^3*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
```

c)*b^4*c^5 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
2*b*c^6 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2
*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^6 +
2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^7 - 2*(b^
2 - 4*a*c)*b^3*c^6 + 4*(b^2 - 4*a*c)*a*b*c^7)*h - (2*b^6*c^5 - 14*a*b^4*c^6
 + 24*a^2*b^2*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*b^6*c^3 + 7*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*
b^4*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c
^4 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c
^5 - 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^5 + 3*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^6 - 2*(b^2 -
 4*a*c)*b^4*c^5 + 6*(b^2 - 4*a*c)*a*b^2*c^6)*k + (2*b^7*c^4 - 16*a*b^5*c^5
+ 36*a^2*b^3*c^6 - 16*a^3*b*c^7 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*b^7*c^2 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a*b^5*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*b^6*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^3*c^4 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a*b^4*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b
^5*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*
c^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c
^5 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^5
- 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^6 - 2
*(b^2 - 4*a*c)*b^5*c^4 + 8*(b^2 - 4*a*c)*a*b^3*c^5 - 4*(b^2 - 4*a*c)*a^2*b*
c^6)*m)*arctan(2*sqrt(1/2)*x/sqrt((b*c^11 - sqrt(b^2*c^22 - 4*a*c^23))/c^12
))/((a*b^4*c^5 - 8*a^2*b^2*c^6 - 2*a*b^3*c^6 + 16*a^3*c^7 + 8*a^2*b*c^7 + a
*b^2*c^7 - 4*a^2*c^8)*c^2) + 1/4*(c^2*g - b*c*j + b^2*l - a*c*l)*log(abs(c*
x^4 + b*x^2 + a))/c^3 + 1/16*(2*(b^5*c^3 - 8*a*b^3*c^4 - 2*b^4*c^4 + 16*a^2
*b*c^5 + 8*a*b^2*c^5 + b^3*c^5 - 4*a*b*c^6 + (b^4*c^3 - 8*a*b^2*c^4 - 2*b^3
*c^4 + 16*a^2*c^5 + 8*a*b*c^5 + b^2*c^5 - 4*a*c^6)*sqrt(b^2 - 4*a*c))*e*abs
(c) - (b^6*c^2 - 8*a*b^4*c^3 - 2*b^5*c^3 + 16*a^2*b^2*c^4 + 8*a*b^3*c^4 + b
^4*c^4 - 4*a*b^2*c^5 + (b^5*c^2 - 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4 +
8*a*b^2*c^4 + b^3*c^4 - 4*a*b*c^5)*sqrt(b^2 - 4*a*c))*g*abs(c) + (b^7*c - 1
0*a*b^5*c^2 - 2*b^6*c^2 + 32*a^2*b^3*c^3 + 12*a*b^4*c^3 + b^5*c^3 - 32*a^3*
b*c^4 - 16*a^2*b^2*c^4 - 6*a*b^3*c^4 + 8*a^2*b*c^5 + (b^6*c - 10*a*b^4*c^2
- 2*b^5*c^2 + 32*a^2*b^2*c^3 + 12*a*b^3*c^3 + b^4*c^3 - 32*a^3*c^4 - 16*a^2
*b*c^4 - 6*a*b^2*c^4 + 8*a^2*c^5)*sqrt(b^2 - 4*a*c))*j*abs(c) - (b^8 - 11*a
*b^6*c - 2*b^7*c + 40*a^2*b^4*c^2 + 14*a*b^5*c^2 + b^6*c^2 - 48*a^3*b^2*c^3
 - 24*a^2*b^3*c^3 - 7*a*b^4*c^3 + 12*a^2*b^2*c^4 + (b^7 - 11*a*b^5*c - 2*b^
6*c + 40*a^2*b^3*c^2 + 14*a*b^4*c^2 + b^5*c^2 - 48*a^3*b*c^3 - 24*a^2*b^2*c
^3 - 7*a*b^3*c^3 + 12*a^2*b*c^4)*sqrt(b^2 - 4*a*c))*l*abs(c) - 2*(b^5*c^4 -
 8*a*b^3*c^5 - 2*b^4*c^5 + 16*a^2*b*c^6 + 8*a*b^2*c^6 + b^3*c^6 - 4*a*b*c^7
 - (b^4*c^4 - 4*a*b^2*c^5 - 2*b^3*c^5 + b^2*c^6)*sqrt(b^2 - 4*a*c))*e + (b^
6*c^3 - 8*a*b^4*c^4 - 2*b^5*c^4 + 16*a^2*b^2*c^5 + 8*a*b^3*c^5 + b^4*c^5 -
4*a*b^2*c^6 + (b^5*c^3 - 4*a*b^3*c^4 - 2*b^4*c^4 + b^3*c^5)*sqrt(b^2 - 4*a*

```
c))*g - (b^7*c^2 - 10*a*b^5*c^3 - 2*b^6*c^3 + 32*a^2*b^3*c^4 + 12*a*b^4*c^4
 + b^5*c^4 - 32*a^3*b*c^5 - 16*a^2*b^2*c^5 - 6*a*b^3*c^5 + 8*a^2*b*c^6 - (b
^6*c^2 - 6*a*b^4*c^3 - 2*b^5*c^3 + 8*a^2*b^2*c^4 + 4*a*b^3*c^4 + b^4*c^4 -
2*a*b^2*c^5)*sqrt(b^2 - 4*a*c))*j + (b^8*c - 11*a*b^6*c^2 - 2*b^7*c^2 + 40*
a^2*b^4*c^3 + 14*a*b^5*c^3 + b^6*c^3 - 48*a^3*b^2*c^4 - 24*a^2*b^3*c^4 - 7*
a*b^4*c^4 + 12*a^2*b^2*c^5 - (b^7*c - 7*a*b^5*c^2 - 2*b^6*c^2 + 12*a^2*b^3*
c^3 + 6*a*b^4*c^3 + b^5*c^3 - 3*a*b^3*c^4)*sqrt(b^2 - 4*a*c))*l)*log(x^2 +
1/2*(b*c^11 + sqrt(b^2*c^22 - 4*a*c^23))/c^12)/((a*b^4*c^2 - 8*a^2*b^2*c^3
- 2*a*b^3*c^3 + 16*a^3*c^4 + 8*a^2*b*c^4 + a*b^2*c^4 - 4*a^2*c^5)*c^2*abs(c
)) + 1/16*(2*(b^5*c^3 - 8*a*b^3*c^4 - 2*b^4*c^4 + 16*a^2*b*c^5 + 8*a*b^2*c^
5 + b^3*c^5 - 4*a*b*c^6 + (b^4*c^3 - 8*a*b^2*c^4 - 2*b^3*c^4 + 16*a^2*c^5 +
 8*a*b*c^5 + b^2*c^5 - 4*a*c^6)*sqrt(b^2 - 4*a*c))*e*abs(c) - (b^6*c^2 - 8*
a*b^4*c^3 - 2*b^5*c^3 + 16*a^2*b^2*c^4 + 8*a*b^3*c^4 + b^4*c^4 - 4*a*b^2*c^
5 - (b^5*c^2 - 8*a*b^3*c^3 - 2*b^4*c^3 + 16*a^2*b*c^4 + 8*a*b^2*c^4 + b^3*c
^4 - 4*a*b*c^5)*sqrt(b^2 - 4*a*c))*g*abs(c) + (b^7*c - 10*a*b^5*c^2 - 2*b^6
*c^2 + 32*a^2*b^3*c^3 + 12*a*b^4*c^3 + b^5*c^3 - 32*a^3*b*c^4 - 16*a^2*b^2*
c^4 - 6*a*b^3*c^4 + 8*a^2*b*c^5 - (b^6*c - 10*a*b^4*c^2 - 2*b^5*c^2 + 32*a^
2*b^2*c^3 + 12*a*b^3*c^3 + b^4*c^3 - 32*a^3*c^4 - 16*a^2*b*c^4 - 6*a*b^2*c^
4 + 8*a^2*c^5)*sqrt(b^2 - 4*a*c))*j*abs(c) - (b^8 - 11*a*b^6*c - 2*b^7*c +
40*a^2*b^4*c^2 + 14*a*b^5*c^2 + b^6*c^2 - 48*a^3*b^2*c^3 - 24*a^2*b^3*c^3 -
 7*a*b^4*c^3 + 12*a^2*b^2*c^4 - (b^7 - 11*a*b^5*c - 2*b^6*c + 40*a^2*b^3*c^
2 + 14*a*b^4*c^2 + b^5*c^2 - 48*a^3*b*c^3 - 24*a^2*b^2*c^3 - 7*a*b^3*c^3 +
12*a^2*b*c^4)*sqrt(b^2 - 4*a*c))*l*abs(c) - 2*(b^5*c^4 - 8*a*b^3*c^5 - 2*b^
4*c^5 + 16*a^2*b*c^6 + 8*a*b^2*c^6 + b^3*c^6 - 4*a*b*c^7 + (b^4*c^4 - 4*a*b
^2*c^5 - 2*b^3*c^5 + b^2*c^6)*sqrt(b^2 - 4*a*c))*e + (b^6*c^3 - 8*a*b^4*c^4
 - 2*b^5*c^4 + 16*a^2*b^2*c^5 + 8*a*b^3*c^5 + b^4*c^5 - 4*a*b^2*c^6 - (b^5*
c^3 - 4*a*b^3*c^4 - 2*b^4*c^4 + b^3*c^5)*sqrt(b^2 - 4*a*c))*g - (b^7*c^2 -
10*a*b^5*c^3 - 2*b^6*c^3 + 32*a^2*b^3*c^4 + 12*a*b^4*c^4 + b^5*c^4 - 32*a^3
*b*c^5 - 16*a^2*b^2*c^5 - 6*a*b^3*c^5 + 8*a^2*b*c^6 - (b^6*c^2 - 6*a*b^4*c^
3 - 2*b^5*c^3 + 8*a^2*b^2*c^4 + 4*a*b^3*c^4 + b^4*c^4 - 2*a*b^2*c^5)*sqrt(b
^2 - 4*a*c))*j + (b^8*c - 11*a*b^6*c^2 - 2*b^7*c^2 + 40*a^2*b^4*c^3 + 14*a*
b^5*c^3 + b^6*c^3 - 48*a^3*b^2*c^4 - 24*a^2*b^3*c^4 - 7*a*b^4*c^4 + 12*a^2*
b^2*c^5 - (b^7*c - 7*a*b^5*c^2 - 2*b^6*c^2 + 12*a^2*b^3*c^3 + 6*a*b^4*c^3 +
 b^5*c^3 - 3*a*b^3*c^4)*sqrt(b^2 - 4*a*c))*l)*log(x^2 + 1/2*(b*c^11 - sqrt(
b^2*c^22 - 4*a*c^23))/c^12)/((a*b^4*c^2 - 8*a^2*b^2*c^3 - 2*a*b^3*c^3 + 16*
a^3*c^4 + 8*a^2*b*c^4 + a*b^2*c^4 - 4*a^2*c^5)*c^2*abs(c)) + 1/60*(12*c^4*m
*x^5 + 15*c^4*l*x^4 + 20*c^4*k*x^3 - 20*b*c^3*m*x^3 + 30*c^4*j*x^2 - 30*b*c
^3*l*x^2 + 60*c^4*h*x - 60*b*c^3*k*x + 60*b^2*c^2*m*x - 60*a*c^3*m*x)/c^5
```

Mupad [B] (verification not implemented)

Time = 10.69 (sec) , antiderivative size = 49150, normalized size of antiderivative = 90.18

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^2+cx^4}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4),x)
```

```
[Out] x^2*(j/(2*c) - (b*l)/(2*c^2)) - x*((b*(k/c - (b*m)/c^2))/c - h/c + (a*m)/c^
2) + x^3*(k/(3*c) - (b*m)/(3*c^2)) + symsum(log((c^7*d*e^2 - a*c^6*f^3 - c^
7*d^2*f + b^7*d*m^2 + a^4*c^3*k^3 + a^4*b^3*m^3 + a^2*b*c^4*h^3 + b^2*c^5*d
*g^2 + b^3*c^4*d*h^2 + a^2*c^5*d*j^2 - a^2*c^5*f*h^2 + a^2*c^5*g^2*h + b^4*
c^3*d*j^2 - a^3*c^4*d*l^2 - b^2*c^5*d^2*k + b^5*c^2*d*k^2 + 3*a^2*c^5*f^2*k
 - 3*a^3*c^4*f*k^2 + a^2*c^5*e^2*m - a^3*c^4*h*j^2 + b^3*c^4*d^2*m + a^3*c^
4*h^2*k - a^4*c^3*f*m^2 + a^2*b^5*h*m^2 - a^3*c^4*g^2*m + a^4*c^3*h*l^2 - a
^3*b^4*k*m^2 + a^4*c^3*j^2*m + a^5*c^2*k*m^2 - a^5*c^2*l^2*m - a^3*b^2*c^2*
k^3 - a*c^6*d*g^2 + b*c^6*d*f^2 - a*c^6*e^2*h + b*c^6*d^2*h + a*c^6*d^2*k -
 2*a^5*b*c*m^3 + b^6*c*d*l^2 - a*b^6*f*m^2 - 2*a*b*c^5*d*h^2 - a*b*c^5*f*g^
2 + 2*a*b*c^5*f^2*h + a*b*c^5*e^2*k - 2*a*b*c^5*d^2*m - 6*a*b^5*c*d*m^2 - 2
*b^2*c^5*d*f*h - a*b^5*c*f*l^2 + 2*b^2*c^5*d*e*j - 2*b^3*c^4*d*e*l + 2*b^3*
c^4*d*f*k - 2*b^3*c^4*d*g*j - 2*a^2*c^5*d*f*m + 2*a^2*c^5*d*g*l - 2*a^2*c^5
*d*h*k - 2*a^2*c^5*e*f*l - 2*a^2*c^5*e*g*k + 2*a^2*c^5*e*h*j - 2*a^2*c^5*f*
g*j - 2*b^4*c^3*d*f*m + 2*b^4*c^3*d*g*l - 2*b^4*c^3*d*h*k + 2*b^5*c^2*d*h*m
 + 2*a^3*c^4*f*h*m - 2*a^3*c^4*g*h*l - 2*b^5*c^2*d*j*l + 2*a^3*c^4*d*k*m -
2*a^3*c^4*e*j*m + 2*a^3*c^4*e*k*l + 2*a^3*c^4*f*j*l + 2*a^3*c^4*g*j*k + 2*a
^4*c^3*g*l*m - 2*a^4*c^3*h*k*m - 2*a^4*c^3*j*k*l - 3*a*b^2*c^4*d*j^2 - a*b^
2*c^4*f*h^2 - 4*a*b^3*c^3*d*k^2 + 3*a^2*b*c^4*d*k^2 - a*b^3*c^3*f*j^2 - 5*a
*b^4*c^2*d*l^2 + 2*a^2*b*c^4*f*j^2 - 2*a*b^2*c^4*f^2*k - a*b^4*c^2*f*k^2 -
4*a^3*b*c^3*d*m^2 - a*b^2*c^4*e^2*m - 3*a^3*b*c^3*f*l^2 + 2*a*b^3*c^3*f^2*m
 - 5*a^2*b*c^4*f^2*m + 5*a^2*b^4*c*f*m^2 + a^2*b^4*c*h*l^2 - 4*a^3*b*c^3*h^
2*m - a^3*b*c^3*j^2*k - 4*a^3*b^3*c*h*m^2 + 5*a^4*b*c^2*h*m^2 - a^3*b^3*c*k
*l^2 + 2*a^4*b*c^2*k*l^2 + 2*a^3*b^3*c*k^2*m - 3*a^4*b*c^2*k^2*m + a^4*b^2*
c*k*m^2 + a^4*b^2*c*l^2*m - 2*b*c^6*d*e*g + 2*a*c^6*d*f*h + 2*a*c^6*e*f*g -
 2*a*c^6*d*e*j - 2*b^6*c*d*k*m + 6*a^2*b^2*c^3*d*l^2 + 3*a^2*b^2*c^3*f*k^2
+ 10*a^2*b^3*c^2*d*m^2 + a^2*b^2*c^3*h*j^2 + 4*a^2*b^3*c^2*f*l^2 - 2*a^2*b^
2*c^3*h^2*k + a^2*b^3*c^2*h*k^2 - 6*a^3*b^2*c^2*f*m^2 - 3*a^3*b^2*c^2*h*l^2
 + 2*a^2*b^3*c^2*h^2*m + 4*a*b*c^5*d*e*l - 4*a*b*c^5*d*f*k + 4*a*b*c^5*d*g*
j - 2*a*b*c^5*e*f*j + 2*a*b^5*c*f*k*m + 6*a*b^2*c^4*d*f*m - 6*a*b^2*c^4*d*g
*l + 6*a*b^2*c^4*d*h*k + 2*a*b^2*c^4*e*f*l + 2*a*b^2*c^4*f*g*j - 8*a*b^3*c^
3*d*h*m - 2*a*b^3*c^3*f*g*l + 2*a*b^3*c^3*f*h*k + 6*a^2*b*c^4*d*h*m + 2*a^2
*b*c^4*e*g*m - 2*a^2*b*c^4*e*h*l + 4*a^2*b*c^4*f*g*l - 2*a^2*b*c^4*f*h*k -
2*a^2*b*c^4*g*h*j + 8*a*b^3*c^3*d*j*l - 6*a^2*b*c^4*d*j*l - 2*a*b^4*c^2*f*h
*m + 10*a*b^4*c^2*d*k*m + 2*a*b^4*c^2*f*j*l + 8*a^3*b*c^3*f*k*m - 2*a^3*b*c
```

^3*g*k*l + 4*a^3*b*c^3*h*j*l - 2*a^2*b^4*c*h*k*m - 2*a^4*b*c^2*j*l*m + 4*a^
2*b^2*c^3*f*h*m + 2*a^2*b^2*c^3*g*h*l - 12*a^2*b^2*c^3*d*k*m - 6*a^2*b^2*c^
3*f*j*l - 8*a^2*b^3*c^2*f*k*m - 2*a^2*b^3*c^2*h*j*l + 4*a^3*b^2*c^2*h*k*m +
2*a^3*b^2*c^2*j*k*l)/c^5 - root(128*a^2*b^2*c^8*z^4 - 16*a*b^4*c^7*z^4 - 2
56*a^3*c^9*z^4 + 384*a^3*b^2*c^6*l*z^3 - 144*a^2*b^4*c^5*l*z^3 + 128*a^2*b^
3*c^6*j*z^3 - 128*a^2*b^2*c^7*g*z^3 + 16*a*b^6*c^4*l*z^3 - 256*a^3*b*c^7*j*
z^3 - 16*a*b^5*c^5*j*z^3 + 16*a*b^4*c^6*g*z^3 - 256*a^4*c^7*l*z^3 + 256*a^3
*c^8*g*z^3 - 96*a^4*b*c^5*j*l*z^2 + 8*a*b^7*c^2*j*l*z^2 + 160*a^4*b*c^5*h*m
*z^2 - 8*a*b^7*c^2*h*m*z^2 + 8*a*b^6*c^3*h*k*z^2 - 8*a*b^6*c^3*g*l*z^2 + 8*
a*b^6*c^3*f*m*z^2 + 160*a^3*b*c^6*g*j*z^2 - 96*a^3*b*c^6*f*k*z^2 - 96*a^3*b
*c^6*e*l*z^2 - 96*a^3*b*c^6*d*m*z^2 + 8*a*b^5*c^4*g*j*z^2 - 8*a*b^5*c^4*f*k
*z^2 - 8*a*b^5*c^4*e*l*z^2 - 8*a*b^5*c^4*d*m*z^2 + 8*a*b^4*c^5*e*j*z^2 + 8*
a*b^4*c^5*d*k*z^2 + 8*a*b^4*c^5*f*h*z^2 + 32*a^2*b*c^7*e*g*z^2 + 32*a^2*b*c
^7*d*h*z^2 - 8*a*b^3*c^6*e*g*z^2 - 8*a*b^3*c^6*d*h*z^2 + 16*a*b^2*c^7*d*f*z
^2 + 8*a*b^8*c*k*m*z^2 - 304*a^4*b^2*c^4*k*m*z^2 + 264*a^3*b^4*c^3*k*m*z^2
- 80*a^2*b^6*c^2*k*m*z^2 + 184*a^3*b^3*c^4*j*l*z^2 - 72*a^2*b^5*c^3*j*l*z^2
- 200*a^3*b^3*c^4*h*m*z^2 + 72*a^2*b^5*c^3*h*m*z^2 - 240*a^3*b^2*c^5*g*l*z
^2 + 144*a^3*b^2*c^5*h*k*z^2 + 144*a^3*b^2*c^5*f*m*z^2 + 80*a^2*b^4*c^4*g*l
*z^2 - 64*a^2*b^4*c^4*h*k*z^2 - 64*a^2*b^4*c^4*f*m*z^2 - 72*a^2*b^3*c^5*g*j
*z^2 + 56*a^2*b^3*c^5*f*k*z^2 + 56*a^2*b^3*c^5*e*l*z^2 + 56*a^2*b^3*c^5*d*m
*z^2 - 48*a^2*b^2*c^6*e*j*z^2 - 48*a^2*b^2*c^6*d*k*z^2 - 48*a^2*b^2*c^6*f*h
*z^2 - 112*a^5*b*c^4*m^2*z^2 + 44*a^2*b^7*c*m^2*z^2 + 80*a^4*b*c^5*k^2*z^2
- 4*a*b^7*c^2*k^2*z^2 - 4*a*b^6*c^3*j^2*z^2 - 48*a^3*b*c^6*h^2*z^2 - 4*a*b^
5*c^4*h^2*z^2 - 4*a*b^4*c^5*g^2*z^2 + 16*a^2*b*c^7*f^2*z^2 - 4*a*b^3*c^6*f^
2*z^2 + 8*a*b^2*c^7*e^2*z^2 + 64*a^5*c^5*k*m*z^2 + 192*a^4*c^6*g*l*z^2 - 64
*a^4*c^6*h*k*z^2 - 64*a^4*c^6*f*m*z^2 + 64*a^3*c^7*e*j*z^2 + 64*a^3*c^7*d*k
*z^2 + 64*a^3*c^7*f*h*z^2 - 4*a*b^8*c*l^2*z^2 - 64*a^2*c^8*d*f*z^2 + 16*a*b
*c^8*d^2*z^2 + 252*a^4*b^3*c^3*m^2*z^2 - 168*a^3*b^5*c^2*m^2*z^2 + 168*a^4*
b^2*c^4*l^2*z^2 - 132*a^3*b^4*c^3*l^2*z^2 + 40*a^2*b^6*c^2*l^2*z^2 - 100*a^
3*b^3*c^4*k^2*z^2 + 36*a^2*b^5*c^3*k^2*z^2 - 56*a^3*b^2*c^5*j^2*z^2 + 32*a^
2*b^4*c^4*j^2*z^2 + 28*a^2*b^3*c^5*h^2*z^2 + 40*a^2*b^2*c^6*g^2*z^2 - 96*a^
5*c^5*l^2*z^2 - 32*a^4*c^6*j^2*z^2 - 96*a^3*c^7*g^2*z^2 - 32*a^2*c^8*e^2*z^
2 - 4*b^3*c^7*d^2*z^2 - 4*a*b^9*m^2*z^2 + 32*a^5*b*c^3*h*l*m*z + 8*a^2*b^6*
c*g*k*m*z + 96*a^4*b*c^4*e*k*m*z + 32*a^4*b*c^4*h*j*k*z + 32*a^4*b*c^4*g*j*
l*z + 32*a^4*b*c^4*f*j*m*z - 64*a^4*b*c^4*g*h*m*z - 8*a*b^6*c^2*e*j*l*z + 8
*a*b^6*c^2*e*h*m*z - 64*a^3*b*c^5*e*h*k*z + 64*a^3*b*c^5*e*g*l*z - 64*a^3*b
*c^5*e*f*m*z + 32*a^3*b*c^5*f*g*k*z - 32*a^3*b*c^5*d*h*l*z + 32*a^3*b*c^5*d
*g*m*z - 8*a*b^5*c^3*e*h*k*z + 8*a*b^5*c^3*e*g*l*z - 8*a*b^5*c^3*e*f*m*z -
8*a*b^4*c^4*e*g*j*z + 8*a*b^4*c^4*e*f*k*z - 8*a*b^4*c^4*d*f*l*z + 8*a*b^4*c
^4*d*e*m*z - 32*a^2*b*c^6*d*f*j*z + 32*a^2*b*c^6*d*e*k*z + 8*a*b^3*c^5*d*f*
j*z - 8*a*b^3*c^5*d*e*k*z + 32*a^2*b*c^6*e*f*h*z - 8*a*b^3*c^5*e*f*h*z - 8*
a*b^2*c^6*d*f*g*z + 8*a*b^2*c^6*d*e*h*z - 8*a*b^7*c*e*k*m*z - 40*a^5*b^2*c^
2*k*l*m*z + 48*a^4*b^3*c^2*j*k*m*z - 8*a^4*b^3*c^2*h*l*m*z + 104*a^4*b^2*c^
3*g*k*m*z - 56*a^3*b^4*c^2*g*k*m*z - 40*a^4*b^2*c^3*h*j*m*z + 8*a^4*b^2*c^3
*h*k*l*z + 8*a^4*b^2*c^3*f*l*m*z + 8*a^3*b^4*c^2*h*j*m*z - 152*a^3*b^3*c^3*

e*k*m*z + 64*a^2*b^5*c^2*e*k*m*z - 40*a^3*b^3*c^3*g*j*l*z - 8*a^3*b^3*c^3*h
*j*k*z - 8*a^3*b^3*c^3*f*j*m*z + 8*a^2*b^5*c^2*g*j*l*z + 48*a^3*b^3*c^3*g*h
*m*z - 8*a^2*b^5*c^2*g*h*m*z - 104*a^3*b^2*c^4*e*j*l*z + 56*a^2*b^4*c^3*e*j
*l*z + 8*a^3*b^2*c^4*f*j*k*z - 8*a^3*b^2*c^4*d*k*l*z + 8*a^3*b^2*c^4*d*j*m*
z + 104*a^3*b^2*c^4*e*h*m*z - 56*a^2*b^4*c^3*e*h*m*z - 40*a^3*b^2*c^4*g*h*k
*z - 40*a^3*b^2*c^4*f*g*m*z - 8*a^3*b^2*c^4*f*h*l*z + 8*a^2*b^4*c^3*g*h*k*z
 + 8*a^2*b^4*c^3*f*g*m*z + 48*a^2*b^3*c^4*e*h*k*z - 48*a^2*b^3*c^4*e*g*l*z
+ 48*a^2*b^3*c^4*e*f*m*z - 8*a^2*b^3*c^4*f*g*k*z + 8*a^2*b^3*c^4*d*h*l*z -
8*a^2*b^3*c^4*d*g*m*z + 40*a^2*b^2*c^5*e*g*j*z - 40*a^2*b^2*c^5*e*f*k*z + 4
0*a^2*b^2*c^5*d*f*l*z - 40*a^2*b^2*c^5*d*e*m*z - 8*a^2*b^2*c^5*d*h*j*z + 8*
a^2*b^2*c^5*d*g*k*z + 8*a^2*b^2*c^5*f*g*h*z + 8*a^4*b^4*c*k*l*m*z - 64*a^5*
b*c^3*j*k*m*z - 8*a^3*b^5*c*j*k*m*z - 32*a^6*b*c^2*l*m^2*z + 24*a^5*b^3*c*l
*m^2*z - 28*a^4*b^4*c*j*m^2*z + 16*a^5*b*c^3*k^2*l*z + 4*a^3*b^5*c*j*l^2*z
+ 48*a^5*b*c^3*g*m^2*z + 32*a^3*b^5*c*g*m^2*z - 4*a^2*b^6*c*g*l^2*z - 36*a^
2*b^6*c*e*m^2*z - 32*a^4*b*c^4*g*k^2*z - 16*a^3*b*c^5*f^2*l*z - 48*a^4*b*c^
4*e*l^2*z - 32*a^3*b*c^5*g^2*j*z - 4*a*b^4*c^4*e^2*l*z + 32*a^2*b*c^6*d^2*l
*z - 24*a*b^3*c^5*d^2*l*z + 4*a*b^6*c^2*e*k^2*z + 32*a^3*b*c^5*e*j^2*z + 16
*a^3*b*c^5*g*h^2*z - 16*a^2*b*c^6*e^2*j*z + 4*a*b^5*c^3*e*j^2*z + 4*a*b^3*c
^5*e^2*j*z + 20*a*b^2*c^6*d^2*j*z + 4*a*b^4*c^4*e*h^2*z - 16*a^2*b*c^6*e*g^
2*z + 4*a*b^3*c^5*e*g^2*z - 4*a*b^2*c^6*e^2*g*z + 4*a*b^2*c^6*e*f^2*z + 32*
a^6*c^3*k*l*m*z - 32*a^5*c^4*h*k*l*z + 32*a^5*c^4*h*j*m*z - 32*a^5*c^4*g*k*
m*z - 32*a^5*c^4*f*l*m*z - 32*a^4*c^5*f*j*k*z + 32*a^4*c^5*e*j*l*z + 32*a^4
*c^5*d*k*l*z - 32*a^4*c^5*d*j*m*z + 32*a^4*c^5*g*h*k*z + 32*a^4*c^5*f*h*l*z
 + 32*a^4*c^5*f*g*m*z - 32*a^4*c^5*e*h*m*z - 32*a^3*c^6*e*g*j*z + 32*a^3*c^
6*e*f*k*z + 32*a^3*c^6*d*h*j*z - 32*a^3*c^6*d*g*k*z - 32*a^3*c^6*d*f*l*z +
32*a^3*c^6*d*e*m*z - 32*a^3*c^6*f*g*h*z + 4*a*b^7*c*e*l^2*z + 32*a^2*c^7*d*
f*g*z - 32*a^2*c^7*d*e*h*z - 16*a*b*c^7*d^2*g*z + 52*a^5*b^2*c^2*j*m^2*z -
4*a^4*b^3*c^2*k^2*l*z + 36*a^4*b^2*c^3*j^2*l*z - 16*a^4*b^3*c^2*j*l^2*z - 8
*a^3*b^4*c^2*j^2*l*z - 20*a^4*b^2*c^3*j*k^2*z + 4*a^3*b^4*c^2*j*k^2*z - 76*
a^4*b^3*c^2*g*m^2*z - 60*a^4*b^2*c^3*g*l^2*z + 44*a^3*b^2*c^4*g^2*l*z + 28*
a^3*b^4*c^2*g*l^2*z - 8*a^2*b^4*c^3*g^2*l*z + 104*a^3*b^4*c^2*e*m^2*z - 100
*a^4*b^2*c^3*e*m^2*z + 24*a^3*b^3*c^3*g*k^2*z + 4*a^3*b^2*c^4*h^2*j*z - 4*a
^2*b^5*c^2*g*k^2*z + 4*a^2*b^3*c^4*f^2*l*z + 76*a^3*b^3*c^3*e*l^2*z - 32*a^
2*b^5*c^2*e*l^2*z + 20*a^2*b^2*c^5*e^2*l*z + 12*a^3*b^2*c^4*g*j^2*z + 8*a^2
*b^3*c^4*g^2*j*z - 4*a^2*b^4*c^3*g*j^2*z + 52*a^3*b^2*c^4*e*k^2*z - 28*a^2*
b^4*c^3*e*k^2*z - 4*a^2*b^2*c^5*f^2*j*z - 24*a^2*b^3*c^4*e*j^2*z - 4*a^2*b^
3*c^4*g*h^2*z - 20*a^2*b^2*c^5*e*h^2*z + 20*a^5*b^2*c^2*l^3*z + 4*a^3*b^3*c
^3*j^3*z - 4*a^2*b^2*c^5*g^3*z - 4*a^4*b^5*l*m^2*z - 16*a^6*c^3*j*m^2*z - 1
6*a^5*c^4*j^2*l*z + 4*a^3*b^6*j*m^2*z + 16*a^5*c^4*j*k^2*z + 48*a^5*c^4*g*l
^2*z - 48*a^4*c^5*g^2*l*z - 4*a^2*b^7*g*m^2*z + 16*a^5*c^4*e*m^2*z - 16*a^4
*c^5*h^2*j*z + 16*a^4*c^5*g*j^2*z - 16*a^3*c^6*e^2*l*z + 4*b^5*c^4*d^2*l*z
- 16*a^4*c^5*e*k^2*z + 16*a^3*c^6*f^2*j*z - 4*b^4*c^5*d^2*j*z - 16*a^2*c^7*
d^2*j*z - 4*a^4*b^4*c*l^3*z + 16*a^3*c^6*e*h^2*z - 16*a^4*b*c^4*j^3*z + 16*
a^2*c^7*e^2*g*z + 4*b^3*c^6*d^2*g*z - 16*a^2*c^7*e*f^2*z - 4*b^2*c^7*d^2*e*
z + 4*a*b^8*e*m^2*z + 16*a*c^8*d^2*e*z - 16*a^6*c^3*l^3*z + 16*a^3*c^6*g^3*

```
z + 4*a^5*b^2*c*g*k*l*m + 12*a^5*b*c^2*g*j*k*m + 12*a^5*b*c^2*e*k*l*m - 4*a
^5*b*c^2*h*j*k*l - 4*a^5*b*c^2*f*j*l*m - 4*a^4*b^3*c*g*j*k*m - 4*a^4*b^3*c*
e*k*l*m - 4*a^5*b*c^2*g*h*l*m + 4*a^3*b^4*c*e*j*k*m - 4*a^3*b^4*c*f*h*k*m +
 12*a^4*b*c^3*d*j*k*l - 20*a^4*b*c^3*e*g*k*m + 12*a^4*b*c^3*f*h*j*l + 12*a^
4*b*c^3*e*h*j*m + 12*a^4*b*c^3*d*h*k*m - 4*a^4*b*c^3*g*h*j*k - 4*a^4*b*c^3*
f*g*k*l - 4*a^4*b*c^3*f*g*j*m - 4*a^4*b*c^3*e*h*k*l - 4*a^4*b*c^3*e*f*l*m -
 4*a^4*b*c^3*d*g*l*m - 4*a^2*b^5*c*e*g*k*m + 4*a^2*b^5*c*d*h*k*m - 20*a^3*b
*c^4*d*f*j*l - 4*a^3*b*c^4*e*f*j*k - 4*a^3*b*c^4*d*g*j*k - 4*a^3*b*c^4*d*e*
k*l - 4*a^3*b*c^4*d*e*j*m - 4*a*b^5*c^2*d*f*j*l + 12*a^3*b*c^4*e*g*h*k + 12
*a^3*b*c^4*e*f*g*m + 12*a^3*b*c^4*d*g*h*l + 12*a^3*b*c^4*d*f*h*m - 4*a^3*b*
c^4*f*g*h*j - 4*a^3*b*c^4*e*f*h*l + 4*a*b^5*c^2*d*f*h*m - 4*a*b^4*c^3*d*f*h
*k + 4*a*b^4*c^3*d*f*g*l + 12*a^2*b*c^5*d*f*g*j + 12*a^2*b*c^5*d*e*f*l - 4*
a^2*b*c^5*d*e*h*j - 4*a^2*b*c^5*d*e*g*k - 4*a*b^3*c^4*d*f*g*j - 4*a*b^3*c^4
*d*e*f*l - 4*a^2*b*c^5*e*f*g*h + 4*a*b^2*c^5*d*e*f*j - 4*a^6*b*c*j*k*l*m -
4*a*b^6*c*d*f*k*m - 4*a*b*c^6*d*e*f*g - 16*a^4*b^2*c^2*e*j*k*m + 4*a^4*b^2*
c^2*f*j*k*l + 4*a^4*b^2*c^2*d*j*l*m + 12*a^4*b^2*c^2*f*h*k*m + 4*a^4*b^2*c^
2*g*h*j*m + 4*a^4*b^2*c^2*e*h*l*m - 4*a^3*b^3*c^2*d*j*k*l + 20*a^3*b^3*c^2*
e*g*k*m - 16*a^3*b^3*c^2*d*h*k*m - 4*a^3*b^3*c^2*f*h*j*l - 4*a^3*b^3*c^2*e*
h*j*m - 40*a^3*b^2*c^3*d*f*k*m + 24*a^2*b^4*c^2*d*f*k*m - 16*a^3*b^2*c^3*d*
h*j*l + 12*a^3*b^2*c^3*e*g*j*l + 4*a^3*b^2*c^3*e*h*j*k + 4*a^3*b^2*c^3*e*f*
j*m + 4*a^3*b^2*c^3*d*g*k*l - 4*a^2*b^4*c^2*e*g*j*l + 4*a^2*b^4*c^2*d*h*j*l
 - 16*a^3*b^2*c^3*e*g*h*m + 4*a^3*b^2*c^3*f*g*h*l + 4*a^2*b^4*c^2*e*g*h*m +
 20*a^2*b^3*c^3*d*f*j*l - 16*a^2*b^3*c^3*d*f*h*m - 4*a^2*b^3*c^3*e*g*h*k -
4*a^2*b^3*c^3*e*f*g*m - 4*a^2*b^3*c^3*d*g*h*l - 16*a^2*b^2*c^4*d*f*g*l + 12
*a^2*b^2*c^4*d*f*h*k + 4*a^2*b^2*c^4*e*f*g*k + 4*a^2*b^2*c^4*d*g*h*j + 4*a^
2*b^2*c^4*d*e*h*l + 4*a^2*b^2*c^4*d*e*g*m + 2*a^5*b^2*c*j^2*k*m - 4*a^5*b^2
*c*h*k^2*m - 2*a^5*b*c^2*h^2*k*m + 2*a^4*b^3*c*h^2*k*m + 2*a^5*b^2*c*h*k*l^
2 + 2*a^5*b^2*c*f*l^2*m - 2*a^5*b*c^2*h*j^2*m + 2*a^3*b^4*c*g^2*k*m + 14*a^
4*b*c^3*f^2*k*m - 10*a^5*b*c^2*f*k^2*m - 8*a^5*b^2*c*g*j*m^2 - 8*a^5*b^2*c*
e*l*m^2 + 4*a^5*b^2*c*f*k*m^2 + 4*a^4*b^3*c*f*k^2*m - 2*a^5*b*c^2*g*k^2*l +
 2*a^2*b^5*c*f^2*k*m + 6*a^5*b*c^2*f*k*l^2 + 6*a^5*b*c^2*d*l^2*m - 2*a^5*b*
c^2*g*j*l^2 + 2*a^4*b^3*c*g*j*l^2 - 2*a^4*b^3*c*f*k*l^2 - 2*a^4*b^3*c*d*l^2
*m - 2*a^4*b*c^3*g^2*j*l - 14*a*b^5*c^2*d^2*k*m - 10*a^5*b*c^2*e*j*m^2 + 10
*a^4*b^3*c*e*j*m^2 - 10*a^3*b*c^4*d^2*k*m - 6*a^4*b^3*c*d*k*m^2 + 6*a^4*b*c
^3*g^2*h*m - 4*a^3*b^4*c*d*k^2*m - 2*a^5*b*c^2*d*k*m^2 + 14*a^5*b*c^2*f*h*m
^2 + 14*a^3*b*c^4*e^2*j*l - 10*a^4*b^3*c*f*h*m^2 - 10*a^4*b*c^3*f*h^2*m - 1
0*a^4*b*c^3*e*j^2*l - 2*a^4*b*c^3*g*h^2*l - 2*a^4*b*c^3*f*j^2*k - 2*a^4*b*c
^3*d*j^2*m - 2*a^3*b^4*c*e*j*l^2 + 2*a^3*b^4*c*d*k*l^2 + 2*a*b^5*c^2*e^2*j*
l - 12*a*b^4*c^3*d^2*j*l - 10*a^3*b*c^4*e^2*h*m + 6*a^4*b*c^3*e*j*k^2 + 2*a
^3*b^4*c*f*h*l^2 - 2*a*b^5*c^2*e^2*h*m - 12*a^3*b^4*c*e*g*m^2 + 12*a^3*b^4*
c*d*h*m^2 + 12*a*b^4*c^3*d^2*h*m + 6*a^3*b*c^4*f^2*g*l - 2*a^4*b*c^3*f*h*k^
2 - 2*a^3*b*c^4*f^2*h*k + 14*a^4*b*c^3*e*g*l^2 - 10*a^4*b*c^3*d*h*l^2 - 10*
a^3*b*c^4*e*g^2*l - 2*a^3*b*c^4*f*g^2*k - 2*a^3*b*c^4*d*g^2*m + 2*a^2*b^5*c
*e*g*l^2 - 2*a^2*b^5*c*d*h*l^2 + 2*a*b^4*c^3*e^2*h*k - 2*a*b^4*c^3*e^2*g*l
+ 2*a*b^4*c^3*e^2*f*m - 14*a^2*b^5*c*d*f*m^2 + 14*a^2*b*c^5*d^2*h*k - 10*a^
```

```
4*b*c^3*d*f*m^2 - 10*a^3*b*c^4*d*h^2*k - 10*a^2*b*c^5*d^2*g*l - 10*a*b^3*c^
4*d^2*h*k + 10*a*b^3*c^4*d^2*g*l - 6*a*b^3*c^4*d^2*f*m - 4*a*b^4*c^3*d*f^2*
m - 2*a^3*b*c^4*e*h^2*j - 2*a^2*b*c^5*d^2*f*m + 6*a^3*b*c^4*d*h*j^2 + 6*a^2
*b*c^5*e^2*f*k + 6*a^2*b*c^5*d*e^2*m - 2*a^3*b*c^4*e*g*j^2 - 2*a^2*b*c^5*e^
2*g*j + 2*a*b^3*c^4*e^2*g*j - 2*a*b^3*c^4*e^2*f*k - 2*a*b^3*c^4*d*e^2*m + 1
4*a^3*b*c^4*d*f*k^2 - 10*a^2*b*c^5*d*f^2*k - 8*a*b^2*c^5*d^2*g*j - 8*a*b^2*
c^5*d^2*e*l + 4*a*b^3*c^4*d*f^2*k + 4*a*b^2*c^5*d^2*f*k - 2*a^2*b*c^5*e*f^2
*j + 2*a*b^5*c^2*d*f*k^2 + 2*a*b^4*c^3*d*f*j^2 + 2*a*b^2*c^5*d*e^2*k - 2*a^
2*b*c^5*d*g^2*h + 2*a*b^2*c^5*e^2*f*h - 4*a*b^2*c^5*d*f^2*h - 2*a^2*b*c^5*d
*f*h^2 + 2*a*b^3*c^4*d*f*h^2 + 2*a*b^2*c^5*d*f*g^2 + 8*a^6*c^2*h*j*l*m - 8*
a^6*c^2*g*k*l*m - 8*a^5*c^3*f*j*k*l + 8*a^5*c^3*e*j*k*m - 8*a^5*c^3*d*j*l*m
 + 8*a^5*c^3*g*h*k*l - 8*a^5*c^3*g*h*j*m - 8*a^5*c^3*f*h*k*m + 8*a^5*c^3*f*
g*l*m - 8*a^5*c^3*e*h*l*m - 2*a^6*b*c*h*l^2*m + 8*a^4*c^4*f*g*j*k - 8*a^4*c
^4*e*h*j*k - 8*a^4*c^4*e*g*j*l + 8*a^4*c^4*e*f*k*l - 8*a^4*c^4*e*f*j*m + 8*
a^4*c^4*d*h*j*l - 8*a^4*c^4*d*g*k*l + 8*a^4*c^4*d*g*j*m + 8*a^4*c^4*d*f*k*m
 + 8*a^4*c^4*d*e*l*m + 6*a^6*b*c*g*l*m^2 - 2*a^6*b*c*h*k*m^2 - 8*a^4*c^4*f*
g*h*l + 8*a^4*c^4*e*g*h*m + 2*a*b^6*c*e^2*k*m + 8*a^3*c^5*d*e*j*k + 8*a^3*c
^5*e*f*h*j - 8*a^3*c^5*e*f*g*k - 8*a^3*c^5*d*g*h*j - 8*a^3*c^5*d*f*h*k + 8*
a^3*c^5*d*f*g*l - 8*a^3*c^5*d*e*h*l - 8*a^3*c^5*d*e*g*m - 8*a^2*c^6*d*e*f*j
 + 8*a^2*c^6*d*e*g*h + 2*a*b^6*c*d*f*l^2 + 6*a*b*c^6*d^2*e*j - 2*a*b*c^6*d^
2*f*h - 2*a*b*c^6*d*e^2*h - 8*a^4*b^2*c^2*g^2*k*m - 10*a^3*b^3*c^2*f^2*k*m
+ 2*a^4*b^2*c^2*h^2*j*l + 18*a^3*b^2*c^3*e^2*k*m - 12*a^2*b^4*c^2*e^2*k*m -
 4*a^4*b^2*c^2*g*j^2*l + 2*a^3*b^3*c^2*g^2*j*l + 28*a^2*b^3*c^3*d^2*k*m + 1
4*a^4*b^2*c^2*d*k^2*m - 8*a^3*b^2*c^3*f^2*j*l + 2*a^4*b^2*c^2*g*j*k^2 + 2*a
^4*b^2*c^2*e*k^2*l - 2*a^3*b^3*c^2*g^2*h*m + 2*a^2*b^4*c^2*f^2*j*l - 10*a^2
*b^3*c^3*e^2*j*l - 8*a^4*b^2*c^2*d*k*l^2 + 4*a^4*b^2*c^2*e*j*l^2 + 4*a^3*b^
3*c^2*f*h^2*m + 4*a^3*b^3*c^2*e*j^2*l + 4*a^3*b^2*c^3*f^2*h*m - 2*a^2*b^4*c
^2*f^2*h*m + 18*a^2*b^2*c^4*d^2*j*l + 10*a^2*b^3*c^3*e^2*h*m - 8*a^4*b^2*c^
2*f*h*l^2 - 2*a^3*b^3*c^2*e*j*k^2 + 2*a^3*b^2*c^3*g^2*h*k + 2*a^3*b^2*c^3*f
*g^2*m - 22*a^4*b^2*c^2*d*h*m^2 - 22*a^2*b^2*c^4*d^2*h*m + 18*a^4*b^2*c^2*e
*g*m^2 + 16*a^3*b^2*c^3*d*h^2*m - 4*a^3*b^2*c^3*f*h^2*k - 4*a^2*b^4*c^2*d*h
^2*m + 2*a^3*b^3*c^2*f*h*k^2 + 2*a^3*b^2*c^3*d*j^2*k + 2*a^2*b^3*c^3*f^2*h*
k - 2*a^2*b^3*c^3*f^2*g*l - 10*a^3*b^3*c^2*e*g*l^2 + 10*a^3*b^3*c^2*d*h*l^2
 - 8*a^2*b^2*c^4*e^2*h*k - 8*a^2*b^2*c^4*e^2*f*m + 4*a^2*b^3*c^3*e*g^2*l +
4*a^2*b^2*c^4*e^2*g*l + 2*a^3*b^2*c^3*f*h*j^2 + 28*a^3*b^3*c^2*d*f*m^2 + 14
*a^2*b^2*c^4*d*f^2*m - 8*a^3*b^2*c^3*e*g*k^2 + 4*a^3*b^2*c^3*d*h*k^2 + 4*a^
2*b^3*c^3*d*h^2*k + 2*a^2*b^4*c^2*e*g*k^2 - 2*a^2*b^4*c^2*d*h*k^2 + 2*a^2*b
^2*c^4*f^2*g*j + 2*a^2*b^2*c^4*e*f^2*l + 18*a^3*b^2*c^3*d*f*l^2 - 12*a^2*b^
4*c^2*d*f*l^2 - 4*a^2*b^2*c^4*e*g^2*j + 2*a^2*b^3*c^3*e*g*j^2 - 2*a^2*b^3*c
^3*d*h*j^2 - 10*a^2*b^3*c^3*d*f*k^2 - 8*a^2*b^2*c^4*d*f*j^2 + 2*a^2*b^2*c^4
*e*g*h^2 + 4*a^5*b^2*c*h^2*m^2 - 2*a^4*b^2*c^2*h^3*m - 5*a^5*b*c^2*g^2*m^2
+ 5*a^4*b^3*c*g^2*m^2 + 3*a^5*b*c^2*h^2*l^2 + 6*a^3*b^4*c*f^2*m^2 - 2*a^3*b
^2*c^3*g^3*l + 2*a^2*b^3*c^3*f^3*m + 7*a^4*b*c^3*e^2*m^2 + 7*a^2*b^5*c*e^2*
m^2 - 5*a^4*b*c^3*f^2*l^2 + 3*a^4*b*c^3*g^2*k^2 - 2*a^4*b^2*c^2*f*k^3 - 2*a
^2*b^2*c^4*f^3*k + 7*a^3*b*c^4*d^2*l^2 + 7*a*b^5*c^2*d^2*l^2 - 5*a^3*b*c^4*
```

e^2*k^2 + 3*a^3*b*c^4*f^2*j^2 + 6*a*b^4*c^3*d^2*k^2 + 2*a^3*b^3*c^2*d*k^3 -
2*a^3*b^2*c^3*e*j^3 - 5*a^2*b*c^5*d^2*j^2 + 5*a*b^3*c^4*d^2*j^2 + 3*a^2*b*
c^5*e^2*h^2 + 4*a*b^2*c^5*d^2*h^2 - 2*a^2*b^2*c^4*d*h^3 - 4*a^6*c^2*j^2*k*m
+ 2*a^6*b^2*j*l*m^2 + 4*a^6*c^2*j*k^2*l + 4*a^6*c^2*h*k^2*m - 4*a^6*c^2*h*
k*l^2 - 4*a^6*c^2*f*l^2*m + 4*a^5*c^3*g^2*k*m + 2*a^5*b^3*h*k*m^2 - 2*a^5*b
^3*g*l*m^2 + 4*a^6*c^2*g*j*m^2 + 4*a^6*c^2*f*k*m^2 + 4*a^6*c^2*e*l*m^2 - 4*
a^5*c^3*h^2*j*l + 4*a^5*c^3*h*j^2*k + 4*a^5*c^3*g*j^2*l + 4*a^5*c^3*f*j^2*m
- 4*a^4*c^4*e^2*k*m + 2*a^4*b^4*g*j*m^2 - 2*a^4*b^4*f*k*m^2 + 2*a^4*b^4*e*
l*m^2 - 4*a^5*c^3*g*j*k^2 - 4*a^5*c^3*e*k^2*l - 4*a^5*c^3*d*k^2*m + 4*a^4*c
^4*f^2*j*l + 4*a^5*c^3*e*j*l^2 + 4*a^5*c^3*d*k*l^2 + 4*a^4*c^4*f^2*h*m + 2*
b^6*c^2*d^2*j*l - 2*a^3*b^5*e*j*m^2 + 2*a^3*b^5*d*k*m^2 + 4*a^5*c^3*f*h*l^2
- 4*a^4*c^4*g^2*h*k - 4*a^4*c^4*f*g^2*m - 4*a^3*c^5*d^2*j*l - 2*b^6*c^2*d^
2*h*m + 2*a^3*b^5*f*h*m^2 + 12*a^5*c^3*d*h*m^2 - 12*a^4*c^4*d*h^2*m + 12*a^
3*c^5*d^2*h*m - 4*a^5*c^3*e*g*m^2 + 4*a^4*c^4*g*h^2*j + 4*a^4*c^4*f*h^2*k +
4*a^4*c^4*e*h^2*l - 4*a^4*c^4*d*j^2*k + 3*a^6*b*c*j^2*m^2 - 4*a^4*c^4*f*h*
j^2 + 4*a^3*c^5*e^2*h*k + 4*a^3*c^5*e^2*g*l + 4*a^3*c^5*e^2*f*m + 2*b^5*c^3
*d^2*h*k - 2*b^5*c^3*d^2*g*l + 2*b^5*c^3*d^2*f*m + 2*a^5*b*c^2*j^3*l + 2*a^
2*b^6*e*g*m^2 - 2*a^2*b^6*d*h*m^2 + 4*a^4*c^4*e*g*k^2 + 4*a^4*c^4*d*h*k^2 -
4*a^3*c^5*f^2*g*j - 4*a^3*c^5*e*f^2*l - 4*a^3*c^5*d*f^2*m - 4*a^4*c^4*d*f*
l^2 + 4*a^3*c^5*e*g^2*j + 4*a^3*c^5*d*g^2*k + 2*b^4*c^4*d^2*g*j - 2*b^4*c^4
*d^2*f*k + 2*b^4*c^4*d^2*e*l - 6*a^3*b*c^4*f^3*m + 4*a^3*c^5*f*g^2*h + 4*a^
2*c^6*d^2*g*j + 4*a^2*c^6*d^2*f*k + 4*a^2*c^6*d^2*e*l - 2*a^5*b^2*c*g*l^3 +
2*a^5*b*c^2*h*k^3 + 2*a^4*b*c^3*h^3*k - 4*a^3*c^5*e*g*h^2 + 4*a^3*c^5*d*f*
j^2 - 4*a^2*c^6*d*e^2*k - 2*b^3*c^5*d^2*e*j + 8*a^5*b^2*c*d*m^3 + 8*a*b^6*c
*d^2*m^2 + 8*a*b^2*c^5*d^3*m - 6*a^5*b*c^2*e*l^3 - 6*a^2*b*c^5*e^3*l - 4*a^
2*c^6*e^2*f*h + 2*b^3*c^5*d^2*f*h + 2*a^4*b^3*c*e*l^3 + 2*a^4*b*c^3*g*j^3 +
2*a^3*b*c^4*g^3*j + 2*a*b^3*c^4*e^3*l + 4*a^2*c^6*e*f^2*g + 4*a^2*c^6*d*f^
2*h - 6*a^4*b*c^3*d*k^3 - 4*a^2*c^6*d*f*g^2 + 2*b^2*c^6*d^2*e*g - 2*a*b^2*c
^5*e^3*j + 2*a^3*b*c^4*f*h^3 + 2*a^2*b*c^5*f^3*h + 2*a^2*b*c^5*e*g^3 + 3*a*
b*c^6*d^2*g^2 - 9*a^4*b^2*c^2*f^2*m^2 + 4*a^4*b^2*c^2*g^2*l^2 - 14*a^3*b^3*
c^2*e^2*m^2 + 5*a^3*b^3*c^2*f^2*l^2 - 20*a^2*b^4*c^2*d^2*m^2 + 16*a^3*b^2*c
^3*d^2*m^2 - 9*a^3*b^2*c^3*e^2*l^2 + 6*a^2*b^4*c^2*e^2*l^2 + 4*a^3*b^2*c^3*
f^2*k^2 - 14*a^2*b^3*c^3*d^2*l^2 + 5*a^2*b^3*c^3*e^2*k^2 - 9*a^2*b^2*c^4*d^
2*k^2 + 4*a^2*b^2*c^4*e^2*j^2 + 4*a^7*c*k*l^2*m - 4*a^7*c*j*l*m^2 + 2*b^7*c
*d^2*k*m + 2*a^6*b*c*k^3*m + 2*a^6*b*c*j*l^3 + 2*a*b^7*d*f*m^2 - 6*a^6*b*c*
f*m^3 - 6*a*b*c^6*d^3*k - 4*a*c^7*d^2*e*g + 4*a*c^7*d*e^2*f + 2*a*b*c^6*e^3
*g + 2*a*b*c^6*d*f^3 - a^5*b^2*c*j^2*l^2 - a^5*b*c^2*j^2*k^2 - a^4*b^3*c*h^
2*l^2 - a^3*b^4*c*g^2*l^2 - a^4*b*c^3*h^2*j^2 - a^2*b^5*c*f^2*l^2 - a*b^5*c
^2*e^2*k^2 - a^3*b*c^4*g^2*h^2 - a*b^4*c^3*e^2*j^2 - a^2*b*c^5*f^2*g^2 - a*
b^3*c^4*e^2*h^2 - a*b^2*c^5*e^2*g^2 + 2*a^7*b*k*m^3 + 4*a^7*c*h*m^3 + 4*a*c
^7*d^3*h + 2*b*c^7*d^3*f - a^6*b*c*k^2*l^2 - 2*a^6*c^2*j^2*l^2 - 6*a^6*c^2*
h^2*m^2 - a*b^6*c*e^2*l^2 - 6*a^5*c^3*g^2*l^2 - 2*a^5*c^3*h^2*k^2 - 2*a^5*c
^3*f^2*m^2 - 6*a^4*c^4*f^2*k^2 - 6*a^4*c^4*d^2*m^2 - 2*a^4*c^4*g^2*j^2 - 2*
a^4*c^4*e^2*l^2 - 6*a^3*c^5*e^2*j^2 - 2*a^3*c^5*d^2*k^2 - 2*a^3*c^5*f^2*h^2
- a*b*c^6*e^2*f^2 - 6*a^2*c^6*d^2*h^2 - 2*a^2*c^6*e^2*g^2 - a^4*b^2*c^2*h^

```
2*k^2 - a^3*b^3*c^2*g^2*k^2 - a^3*b^2*c^3*g^2*j^2 - a^2*b^4*c^2*f^2*k^2 - a
^2*b^3*c^3*f^2*j^2 - a^2*b^2*c^4*f^2*h^2 - 2*a^7*c*k^2*m^2 + 4*a^5*c^3*h^3*
m - 2*a^6*b^2*h*m^3 + 4*a^6*c^2*g*l^3 + 4*a^4*c^4*g^3*l - 2*b^4*c^4*d^3*m +
 2*a^5*b^3*f*m^3 - 4*a^6*c^2*d*m^3 + 4*a^5*c^3*f*k^3 + 4*a^3*c^5*f^3*k - 4*
a^2*c^6*d^3*m + 2*b^3*c^5*d^3*k - 2*a^4*b^4*d*m^3 + 4*a^4*c^4*e*j^3 + 4*a^2
*c^6*e^3*j - 2*b^2*c^6*d^3*h + 4*a^3*c^5*d*h^3 - 2*a*c^7*d^2*f^2 - a^6*b^2*
k^2*m^2 - a^5*b^3*j^2*m^2 - a^4*b^4*h^2*m^2 - a^3*b^5*g^2*m^2 - a^2*b^6*f^2
*m^2 - b^6*c^2*d^2*k^2 - b^5*c^3*d^2*j^2 - b^4*c^4*d^2*h^2 - b^3*c^5*d^2*g^
2 - b^2*c^6*d^2*f^2 - a^7*b*l^2*m^2 - b^7*c*d^2*l^2 - a*b^7*e^2*m^2 - b*c^7
*d^2*e^2 - b^8*d^2*m^2 - a^6*c^2*k^4 - a^5*c^3*j^4 - a^4*c^4*h^4 - a^3*c^5*
g^4 - a^2*c^6*f^4 - a^7*c*l^4 - a*c^7*e^4 - a^8*m^4 - c^8*d^4, z, k1)*(root
(128*a^2*b^2*c^8*z^4 - 16*a*b^4*c^7*z^4 - 256*a^3*c^9*z^4 + 384*a^3*b^2*c^6
*l*z^3 - 144*a^2*b^4*c^5*l*z^3 + 128*a^2*b^3*c^6*j*z^3 - 128*a^2*b^2*c^7*g*
z^3 + 16*a*b^6*c^4*l*z^3 - 256*a^3*b*c^7*j*z^3 - 16*a*b^5*c^5*j*z^3 + 16*a*
b^4*c^6*g*z^3 - 256*a^4*c^7*l*z^3 + 256*a^3*c^8*g*z^3 - 96*a^4*b*c^5*j*l*z^
2 + 8*a*b^7*c^2*j*l*z^2 + 160*a^4*b*c^5*h*m*z^2 - 8*a*b^7*c^2*h*m*z^2 + 8*a
*b^6*c^3*h*k*z^2 - 8*a*b^6*c^3*g*l*z^2 + 8*a*b^6*c^3*f*m*z^2 + 160*a^3*b*c^
6*g*j*z^2 - 96*a^3*b*c^6*f*k*z^2 - 96*a^3*b*c^6*e*l*z^2 - 96*a^3*b*c^6*d*m*
z^2 + 8*a*b^5*c^4*g*j*z^2 - 8*a*b^5*c^4*f*k*z^2 - 8*a*b^5*c^4*e*l*z^2 - 8*a
*b^5*c^4*d*m*z^2 + 8*a*b^4*c^5*e*j*z^2 + 8*a*b^4*c^5*d*k*z^2 + 8*a*b^4*c^5*
f*h*z^2 + 32*a^2*b*c^7*e*g*z^2 + 32*a^2*b*c^7*d*h*z^2 - 8*a*b^3*c^6*e*g*z^2
 - 8*a*b^3*c^6*d*h*z^2 + 16*a*b^2*c^7*d*f*z^2 + 8*a*b^8*c*k*m*z^2 - 304*a^4
*b^2*c^4*k*m*z^2 + 264*a^3*b^4*c^3*k*m*z^2 - 80*a^2*b^6*c^2*k*m*z^2 + 184*a
^3*b^3*c^4*j*l*z^2 - 72*a^2*b^5*c^3*j*l*z^2 - 200*a^3*b^3*c^4*h*m*z^2 + 72*
a^2*b^5*c^3*h*m*z^2 - 240*a^3*b^2*c^5*g*l*z^2 + 144*a^3*b^2*c^5*h*k*z^2 + 1
44*a^3*b^2*c^5*f*m*z^2 + 80*a^2*b^4*c^4*g*l*z^2 - 64*a^2*b^4*c^4*h*k*z^2 -
64*a^2*b^4*c^4*f*m*z^2 - 72*a^2*b^3*c^5*g*j*z^2 + 56*a^2*b^3*c^5*f*k*z^2 +
56*a^2*b^3*c^5*e*l*z^2 + 56*a^2*b^3*c^5*d*m*z^2 - 48*a^2*b^2*c^6*e*j*z^2 -
48*a^2*b^2*c^6*d*k*z^2 - 48*a^2*b^2*c^6*f*h*z^2 - 112*a^5*b*c^4*m^2*z^2 + 4
4*a^2*b^7*c*m^2*z^2 + 80*a^4*b*c^5*k^2*z^2 - 4*a*b^7*c^2*k^2*z^2 - 4*a*b^6*
c^3*j^2*z^2 - 48*a^3*b*c^6*h^2*z^2 - 4*a*b^5*c^4*h^2*z^2 - 4*a*b^4*c^5*g^2*
z^2 + 16*a^2*b*c^7*f^2*z^2 - 4*a*b^3*c^6*f^2*z^2 + 8*a*b^2*c^7*e^2*z^2 + 64
*a^5*c^5*k*m*z^2 + 192*a^4*c^6*g*l*z^2 - 64*a^4*c^6*h*k*z^2 - 64*a^4*c^6*f*
m*z^2 + 64*a^3*c^7*e*j*z^2 + 64*a^3*c^7*d*k*z^2 + 64*a^3*c^7*f*h*z^2 - 4*a*
b^8*c*l^2*z^2 - 64*a^2*c^8*d*f*z^2 + 16*a*b*c^8*d^2*z^2 + 252*a^4*b^3*c^3*m
^2*z^2 - 168*a^3*b^5*c^2*m^2*z^2 + 168*a^4*b^2*c^4*l^2*z^2 - 132*a^3*b^4*c^
3*l^2*z^2 + 40*a^2*b^6*c^2*l^2*z^2 - 100*a^3*b^3*c^4*k^2*z^2 + 36*a^2*b^5*c
^3*k^2*z^2 - 56*a^3*b^2*c^5*j^2*z^2 + 32*a^2*b^4*c^4*j^2*z^2 + 28*a^2*b^3*c
^5*h^2*z^2 + 40*a^2*b^2*c^6*g^2*z^2 - 96*a^5*c^5*l^2*z^2 - 32*a^4*c^6*j^2*z
^2 - 96*a^3*c^7*g^2*z^2 - 32*a^2*c^8*e^2*z^2 - 4*b^3*c^7*d^2*z^2 - 4*a*b^9*
m^2*z^2 + 32*a^5*b*c^3*h*l*m*z + 8*a^2*b^6*c*g*k*m*z + 96*a^4*b*c^4*e*k*m*z
 + 32*a^4*b*c^4*h*j*k*z + 32*a^4*b*c^4*g*j*l*z + 32*a^4*b*c^4*f*j*m*z - 64*
a^4*b*c^4*g*h*m*z - 8*a*b^6*c^2*e*j*l*z + 8*a*b^6*c^2*e*h*m*z - 64*a^3*b*c^
5*e*h*k*z + 64*a^3*b*c^5*e*g*l*z - 64*a^3*b*c^5*e*f*m*z + 32*a^3*b*c^5*f*g*
k*z - 32*a^3*b*c^5*d*h*l*z + 32*a^3*b*c^5*d*g*m*z - 8*a*b^5*c^3*e*h*k*z + 8
```

*a*b^5*c^3*e*g*l*z - 8*a*b^5*c^3*e*f*m*z - 8*a*b^4*c^4*e*g*j*z + 8*a*b^4*c^
4*e*f*k*z - 8*a*b^4*c^4*d*f*l*z + 8*a*b^4*c^4*d*e*m*z - 32*a^2*b*c^6*d*f*j*
z + 32*a^2*b*c^6*d*e*k*z + 8*a*b^3*c^5*d*f*j*z - 8*a*b^3*c^5*d*e*k*z + 32*a
^2*b*c^6*e*f*h*z - 8*a*b^3*c^5*e*f*h*z - 8*a*b^2*c^6*d*f*g*z + 8*a*b^2*c^6*
d*e*h*z - 8*a*b^7*c*e*k*m*z - 40*a^5*b^2*c^2*k*l*m*z + 48*a^4*b^3*c^2*j*k*m
*z - 8*a^4*b^3*c^2*h*l*m*z + 104*a^4*b^2*c^3*g*k*m*z - 56*a^3*b^4*c^2*g*k*m
*z - 40*a^4*b^2*c^3*h*j*m*z + 8*a^4*b^2*c^3*h*k*l*z + 8*a^4*b^2*c^3*f*l*m*z
 + 8*a^3*b^4*c^2*h*j*m*z - 152*a^3*b^3*c^3*e*k*m*z + 64*a^2*b^5*c^2*e*k*m*z
 - 40*a^3*b^3*c^3*g*j*l*z - 8*a^3*b^3*c^3*h*j*k*z - 8*a^3*b^3*c^3*f*j*m*z +
 8*a^2*b^5*c^2*g*j*l*z + 48*a^3*b^3*c^3*g*h*m*z - 8*a^2*b^5*c^2*g*h*m*z - 1
04*a^3*b^2*c^4*e*j*l*z + 56*a^2*b^4*c^3*e*j*l*z + 8*a^3*b^2*c^4*f*j*k*z - 8
*a^3*b^2*c^4*d*k*l*z + 8*a^3*b^2*c^4*d*j*m*z + 104*a^3*b^2*c^4*e*h*m*z - 56
*a^2*b^4*c^3*e*h*m*z - 40*a^3*b^2*c^4*g*h*k*z - 40*a^3*b^2*c^4*f*g*m*z - 8*
a^3*b^2*c^4*f*h*l*z + 8*a^2*b^4*c^3*g*h*k*z + 8*a^2*b^4*c^3*f*g*m*z + 48*a^
2*b^3*c^4*e*h*k*z - 48*a^2*b^3*c^4*e*g*l*z + 48*a^2*b^3*c^4*e*f*m*z - 8*a^2
*b^3*c^4*f*g*k*z + 8*a^2*b^3*c^4*d*h*l*z - 8*a^2*b^3*c^4*d*g*m*z + 40*a^2*b
^2*c^5*e*g*j*z - 40*a^2*b^2*c^5*e*f*k*z + 40*a^2*b^2*c^5*d*f*l*z - 40*a^2*b
^2*c^5*d*e*m*z - 8*a^2*b^2*c^5*d*h*j*z + 8*a^2*b^2*c^5*d*g*k*z + 8*a^2*b^2*
c^5*f*g*h*z + 8*a^4*b^4*c*k*l*m*z - 64*a^5*b*c^3*j*k*m*z - 8*a^3*b^5*c*j*k*
m*z - 32*a^6*b*c^2*l*m^2*z + 24*a^5*b^3*c*l*m^2*z - 28*a^4*b^4*c*j*m^2*z +
16*a^5*b*c^3*k^2*l*z + 4*a^3*b^5*c*j*l^2*z + 48*a^5*b*c^3*g*m^2*z + 32*a^3*
b^5*c*g*m^2*z - 4*a^2*b^6*c*g*l^2*z - 36*a^2*b^6*c*e*m^2*z - 32*a^4*b*c^4*g
*k^2*z - 16*a^3*b*c^5*f^2*l*z - 48*a^4*b*c^4*e*l^2*z - 32*a^3*b*c^5*g^2*j*z
 - 4*a*b^4*c^4*e^2*l*z + 32*a^2*b*c^6*d^2*l*z - 24*a*b^3*c^5*d^2*l*z + 4*a*
b^6*c^2*e*k^2*z + 32*a^3*b*c^5*e*j^2*z + 16*a^3*b*c^5*g*h^2*z - 16*a^2*b*c^
6*e^2*j*z + 4*a*b^5*c^3*e*j^2*z + 4*a*b^3*c^5*e^2*j*z + 20*a*b^2*c^6*d^2*j*
z + 4*a*b^4*c^4*e*h^2*z - 16*a^2*b*c^6*e*g^2*z + 4*a*b^3*c^5*e*g^2*z - 4*a*
b^2*c^6*e^2*g*z + 4*a*b^2*c^6*e*f^2*z + 32*a^6*c^3*k*l*m*z - 32*a^5*c^4*h*k
*l*z + 32*a^5*c^4*h*j*m*z - 32*a^5*c^4*g*k*m*z - 32*a^5*c^4*f*l*m*z - 32*a^
4*c^5*f*j*k*z + 32*a^4*c^5*e*j*l*z + 32*a^4*c^5*d*k*l*z - 32*a^4*c^5*d*j*m*
z + 32*a^4*c^5*g*h*k*z + 32*a^4*c^5*f*h*l*z + 32*a^4*c^5*f*g*m*z - 32*a^4*c
^5*e*h*m*z - 32*a^3*c^6*e*g*j*z + 32*a^3*c^6*e*f*k*z + 32*a^3*c^6*d*h*j*z -
 32*a^3*c^6*d*g*k*z - 32*a^3*c^6*d*f*l*z + 32*a^3*c^6*d*e*m*z - 32*a^3*c^6*
f*g*h*z + 4*a*b^7*c*e*l^2*z + 32*a^2*c^7*d*f*g*z - 32*a^2*c^7*d*e*h*z - 16*
a*b*c^7*d^2*g*z + 52*a^5*b^2*c^2*j*m^2*z - 4*a^4*b^3*c^2*k^2*l*z + 36*a^4*b
^2*c^3*j^2*l*z - 16*a^4*b^3*c^2*j*l^2*z - 8*a^3*b^4*c^2*j^2*l*z - 20*a^4*b^
2*c^3*j*k^2*z + 4*a^3*b^4*c^2*j*k^2*z - 76*a^4*b^3*c^2*g*m^2*z - 60*a^4*b^2
*c^3*g*l^2*z + 44*a^3*b^2*c^4*g^2*l*z + 28*a^3*b^4*c^2*g*l^2*z - 8*a^2*b^4*
c^3*g^2*l*z + 104*a^3*b^4*c^2*e*m^2*z - 100*a^4*b^2*c^3*e*m^2*z + 24*a^3*b^
3*c^3*g*k^2*z + 4*a^3*b^2*c^4*h^2*j*z - 4*a^2*b^5*c^2*g*k^2*z + 4*a^2*b^3*c
^4*f^2*l*z + 76*a^3*b^3*c^3*e*l^2*z - 32*a^2*b^5*c^2*e*l^2*z + 20*a^2*b^2*c
^5*e^2*l*z + 12*a^3*b^2*c^4*g*j^2*z + 8*a^2*b^3*c^4*g^2*j*z - 4*a^2*b^4*c^3
*g*j^2*z + 52*a^3*b^2*c^4*e*k^2*z - 28*a^2*b^4*c^3*e*k^2*z - 4*a^2*b^2*c^5*
f^2*j*z - 24*a^2*b^3*c^4*e*j^2*z - 4*a^2*b^3*c^4*g*h^2*z - 20*a^2*b^2*c^5*e
*h^2*z + 20*a^5*b^2*c^2*l^3*z + 4*a^3*b^3*c^3*j^3*z - 4*a^2*b^2*c^5*g^3*z -

4*a^4*b^5*l*m^2*z - 16*a^6*c^3*j*m^2*z - 16*a^5*c^4*j^2*l*z + 4*a^3*b^6*j*
m^2*z + 16*a^5*c^4*j*k^2*z + 48*a^5*c^4*g*l^2*z - 48*a^4*c^5*g^2*l*z - 4*a^
2*b^7*g*m^2*z + 16*a^5*c^4*e*m^2*z - 16*a^4*c^5*h^2*j*z + 16*a^4*c^5*g*j^2*
z - 16*a^3*c^6*e^2*l*z + 4*b^5*c^4*d^2*l*z - 16*a^4*c^5*e*k^2*z + 16*a^3*c^
6*f^2*j*z - 4*b^4*c^5*d^2*j*z - 16*a^2*c^7*d^2*j*z - 4*a^4*b^4*c*l^3*z + 16
*a^3*c^6*e*h^2*z - 16*a^4*b*c^4*j^3*z + 16*a^2*c^7*e^2*g*z + 4*b^3*c^6*d^2*
g*z - 16*a^2*c^7*e*f^2*z - 4*b^2*c^7*d^2*e*z + 4*a*b^8*e*m^2*z + 16*a*c^8*d
^2*e*z - 16*a^6*c^3*l^3*z + 16*a^3*c^6*g^3*z + 4*a^5*b^2*c*g*k*l*m + 12*a^5
*b*c^2*g*j*k*m + 12*a^5*b*c^2*e*k*l*m - 4*a^5*b*c^2*h*j*k*l - 4*a^5*b*c^2*f
*j*l*m - 4*a^4*b^3*c*g*j*k*m - 4*a^4*b^3*c*e*k*l*m - 4*a^5*b*c^2*g*h*l*m +
4*a^3*b^4*c*e*j*k*m - 4*a^3*b^4*c*f*h*k*m + 12*a^4*b*c^3*d*j*k*l - 20*a^4*b
*c^3*e*g*k*m + 12*a^4*b*c^3*f*h*j*l + 12*a^4*b*c^3*e*h*j*m + 12*a^4*b*c^3*d
*h*k*m - 4*a^4*b*c^3*g*h*j*k - 4*a^4*b*c^3*f*g*k*l - 4*a^4*b*c^3*f*g*j*m -
4*a^4*b*c^3*e*h*k*l - 4*a^4*b*c^3*e*f*l*m - 4*a^4*b*c^3*d*g*l*m - 4*a^2*b^5
*c*e*g*k*m + 4*a^2*b^5*c*d*h*k*m - 20*a^3*b*c^4*d*f*j*l - 4*a^3*b*c^4*e*f*j
*k - 4*a^3*b*c^4*d*g*j*k - 4*a^3*b*c^4*d*e*k*l - 4*a^3*b*c^4*d*e*j*m - 4*a*
b^5*c^2*d*f*j*l + 12*a^3*b*c^4*e*g*h*k + 12*a^3*b*c^4*e*f*g*m + 12*a^3*b*c^
4*d*g*h*l + 12*a^3*b*c^4*d*f*h*m - 4*a^3*b*c^4*f*g*h*j - 4*a^3*b*c^4*e*f*h*
l + 4*a*b^5*c^2*d*f*h*m - 4*a*b^4*c^3*d*f*h*k + 4*a*b^4*c^3*d*f*g*l + 12*a^
2*b*c^5*d*f*g*j + 12*a^2*b*c^5*d*e*f*l - 4*a^2*b*c^5*d*e*h*j - 4*a^2*b*c^5*
d*e*g*k - 4*a*b^3*c^4*d*f*g*j - 4*a*b^3*c^4*d*e*f*l - 4*a^2*b*c^5*e*f*g*h +
4*a*b^2*c^5*d*e*f*j - 4*a^6*b*c*j*k*l*m - 4*a*b^6*c*d*f*k*m - 4*a*b*c^6*d*
e*f*g - 16*a^4*b^2*c^2*e*j*k*m + 4*a^4*b^2*c^2*f*j*k*l + 4*a^4*b^2*c^2*d*j*
l*m + 12*a^4*b^2*c^2*f*h*k*m + 4*a^4*b^2*c^2*g*h*j*m + 4*a^4*b^2*c^2*e*h*l*
m - 4*a^3*b^3*c^2*d*j*k*l + 20*a^3*b^3*c^2*e*g*k*m - 16*a^3*b^3*c^2*d*h*k*m
- 4*a^3*b^3*c^2*f*h*j*l - 4*a^3*b^3*c^2*e*h*j*m - 40*a^3*b^2*c^3*d*f*k*m +
24*a^2*b^4*c^2*d*f*k*m - 16*a^3*b^2*c^3*d*h*j*l + 12*a^3*b^2*c^3*e*g*j*l +
4*a^3*b^2*c^3*e*h*j*k + 4*a^3*b^2*c^3*e*f*j*m + 4*a^3*b^2*c^3*d*g*k*l - 4*
a^2*b^4*c^2*e*g*j*l + 4*a^2*b^4*c^2*d*h*j*l - 16*a^3*b^2*c^3*e*g*h*m + 4*a^
3*b^2*c^3*f*g*h*l + 4*a^2*b^4*c^2*e*g*h*m + 20*a^2*b^3*c^3*d*f*j*l - 16*a^2
*b^3*c^3*d*f*h*m - 4*a^2*b^3*c^3*e*g*h*k - 4*a^2*b^3*c^3*e*f*g*m - 4*a^2*b^
3*c^3*d*g*h*l - 16*a^2*b^2*c^4*d*f*g*l + 12*a^2*b^2*c^4*d*f*h*k + 4*a^2*b^2
*c^4*e*f*g*k + 4*a^2*b^2*c^4*d*g*h*j + 4*a^2*b^2*c^4*d*e*h*l + 4*a^2*b^2*c^
4*d*e*g*m + 2*a^5*b^2*c*j^2*k*m - 4*a^5*b^2*c*h*k^2*m - 2*a^5*b*c^2*h^2*k*m
+ 2*a^4*b^3*c*h^2*k*m + 2*a^5*b^2*c*h*k*l^2 + 2*a^5*b^2*c*f*l^2*m - 2*a^5*
b*c^2*h*j^2*m + 2*a^3*b^4*c*g^2*k*m + 14*a^4*b*c^3*f^2*k*m - 10*a^5*b*c^2*f
*k^2*m - 8*a^5*b^2*c*g*j*m^2 - 8*a^5*b^2*c*e*l*m^2 + 4*a^5*b^2*c*f*k*m^2 +
4*a^4*b^3*c*f*k^2*m - 2*a^5*b*c^2*g*k^2*l + 2*a^2*b^5*c*f^2*k*m + 6*a^5*b*c
^2*f*k*l^2 + 6*a^5*b*c^2*d*l^2*m - 2*a^5*b*c^2*g*j*l^2 + 2*a^4*b^3*c*g*j*l^
2 - 2*a^4*b^3*c*f*k*l^2 - 2*a^4*b^3*c*d*l^2*m - 2*a^4*b*c^3*g^2*j*l - 14*a*
b^5*c^2*d^2*k*m - 10*a^5*b*c^2*e*j*m^2 + 10*a^4*b^3*c*e*j*m^2 - 10*a^3*b*c^
4*d^2*k*m - 6*a^4*b^3*c*d*k*m^2 + 6*a^4*b*c^3*g^2*h*m - 4*a^3*b^4*c*d*k^2*m
- 2*a^5*b*c^2*d*k*m^2 + 14*a^5*b*c^2*f*h*m^2 + 14*a^3*b*c^4*e^2*j*l - 10*a
^4*b^3*c*f*h*m^2 - 10*a^4*b*c^3*f*h^2*m - 10*a^4*b*c^3*e*j^2*l - 2*a^4*b*c^
3*g*h^2*l - 2*a^4*b*c^3*f*j^2*k - 2*a^4*b*c^3*d*j^2*m - 2*a^3*b^4*c*e*j*l^2

+ 2*a^3*b^4*c*d*k*l^2 + 2*a*b^5*c^2*e^2*j*l - 12*a*b^4*c^3*d^2*j*l - 10*a^
3*b*c^4*e^2*h*m + 6*a^4*b*c^3*e*j*k^2 + 2*a^3*b^4*c*f*h*l^2 - 2*a*b^5*c^2*e
^2*h*m - 12*a^3*b^4*c*e*g*m^2 + 12*a^3*b^4*c*d*h*m^2 + 12*a*b^4*c^3*d^2*h*m
+ 6*a^3*b*c^4*f^2*g*l - 2*a^4*b*c^3*f*h*k^2 - 2*a^3*b*c^4*f^2*h*k + 14*a^4
*b*c^3*e*g*l^2 - 10*a^4*b*c^3*d*h*l^2 - 10*a^3*b*c^4*e*g^2*l - 2*a^3*b*c^4*
f*g^2*k - 2*a^3*b*c^4*d*g^2*m + 2*a^2*b^5*c*e*g*l^2 - 2*a^2*b^5*c*d*h*l^2 +
2*a*b^4*c^3*e^2*h*k - 2*a*b^4*c^3*e^2*g*l + 2*a*b^4*c^3*e^2*f*m - 14*a^2*b
^5*c*d*f*m^2 + 14*a^2*b*c^5*d^2*h*k - 10*a^4*b*c^3*d*f*m^2 - 10*a^3*b*c^4*d
*h^2*k - 10*a^2*b*c^5*d^2*g*l - 10*a*b^3*c^4*d^2*h*k + 10*a*b^3*c^4*d^2*g*l
- 6*a*b^3*c^4*d^2*f*m - 4*a*b^4*c^3*d*f^2*m - 2*a^3*b*c^4*e*h^2*j - 2*a^2*
b*c^5*d^2*f*m + 6*a^3*b*c^4*d*h*j^2 + 6*a^2*b*c^5*e^2*f*k + 6*a^2*b*c^5*d*e
^2*m - 2*a^3*b*c^4*e*g*j^2 - 2*a^2*b*c^5*e^2*g*j + 2*a*b^3*c^4*e^2*g*j - 2*
a*b^3*c^4*e^2*f*k - 2*a*b^3*c^4*d*e^2*m + 14*a^3*b*c^4*d*f*k^2 - 10*a^2*b*c
^5*d*f^2*k - 8*a*b^2*c^5*d^2*g*j - 8*a*b^2*c^5*d^2*e*l + 4*a*b^3*c^4*d*f^2*
k + 4*a*b^2*c^5*d^2*f*k - 2*a^2*b*c^5*e*f^2*j + 2*a*b^5*c^2*d*f*k^2 + 2*a*b
^4*c^3*d*f*j^2 + 2*a*b^2*c^5*d*e^2*k - 2*a^2*b*c^5*d*g^2*h + 2*a*b^2*c^5*e^
2*f*h - 4*a*b^2*c^5*d*f^2*h - 2*a^2*b*c^5*d*f*h^2 + 2*a*b^3*c^4*d*f*h^2 + 2
*a*b^2*c^5*d*f*g^2 + 8*a^6*c^2*h*j*l*m - 8*a^6*c^2*g*k*l*m - 8*a^5*c^3*f*j*
k*l + 8*a^5*c^3*e*j*k*m - 8*a^5*c^3*d*j*l*m + 8*a^5*c^3*g*h*k*l - 8*a^5*c^3
*g*h*j*m - 8*a^5*c^3*f*h*k*m + 8*a^5*c^3*f*g*l*m - 8*a^5*c^3*e*h*l*m - 2*a^
6*b*c*h*l^2*m + 8*a^4*c^4*f*g*j*k - 8*a^4*c^4*e*h*j*k - 8*a^4*c^4*e*g*j*l +
8*a^4*c^4*e*f*k*l - 8*a^4*c^4*e*f*j*m + 8*a^4*c^4*d*h*j*l - 8*a^4*c^4*d*g*
k*l + 8*a^4*c^4*d*g*j*m + 8*a^4*c^4*d*f*k*m + 8*a^4*c^4*d*e*l*m + 6*a^6*b*c
*g*l*m^2 - 2*a^6*b*c*h*k*m^2 - 8*a^4*c^4*f*g*h*l + 8*a^4*c^4*e*g*h*m + 2*a*
b^6*c*e^2*k*m + 8*a^3*c^5*d*e*j*k + 8*a^3*c^5*e*f*h*j - 8*a^3*c^5*e*f*g*k -
8*a^3*c^5*d*g*h*j - 8*a^3*c^5*d*f*h*k + 8*a^3*c^5*d*f*g*l - 8*a^3*c^5*d*e*
h*l - 8*a^3*c^5*d*e*g*m - 8*a^2*c^6*d*e*f*j + 8*a^2*c^6*d*e*g*h + 2*a*b^6*c
*d*f*l^2 + 6*a*b*c^6*d^2*e*j - 2*a*b*c^6*d^2*f*h - 2*a*b*c^6*d*e^2*h - 8*a^
4*b^2*c^2*g^2*k*m - 10*a^3*b^3*c^2*f^2*k*m + 2*a^4*b^2*c^2*h^2*j*l + 18*a^3
*b^2*c^3*e^2*k*m - 12*a^2*b^4*c^2*e^2*k*m - 4*a^4*b^2*c^2*g*j^2*l + 2*a^3*b
^3*c^2*g^2*j*l + 28*a^2*b^3*c^3*d^2*k*m + 14*a^4*b^2*c^2*d*k^2*m - 8*a^3*b^
2*c^3*f^2*j*l + 2*a^4*b^2*c^2*g*j*k^2 + 2*a^4*b^2*c^2*e*k^2*l - 2*a^3*b^3*c
^2*g^2*h*m + 2*a^2*b^4*c^2*f^2*j*l - 10*a^2*b^3*c^3*e^2*j*l - 8*a^4*b^2*c^2
*d*k*l^2 + 4*a^4*b^2*c^2*e*j*l^2 + 4*a^3*b^3*c^2*f*h^2*m + 4*a^3*b^3*c^2*e*
j^2*l + 4*a^3*b^2*c^3*f^2*h*m - 2*a^2*b^4*c^2*f^2*h*m + 18*a^2*b^2*c^4*d^2*
j*l + 10*a^2*b^3*c^3*e^2*h*m - 8*a^4*b^2*c^2*f*h*l^2 - 2*a^3*b^3*c^2*e*j*k^
2 + 2*a^3*b^2*c^3*g^2*h*k + 2*a^3*b^2*c^3*f*g^2*m - 22*a^4*b^2*c^2*d*h*m^2
- 22*a^2*b^2*c^4*d^2*h*m + 18*a^4*b^2*c^2*e*g*m^2 + 16*a^3*b^2*c^3*d*h^2*m
- 4*a^3*b^2*c^3*f*h^2*k - 4*a^2*b^4*c^2*d*h^2*m + 2*a^3*b^3*c^2*f*h*k^2 + 2
*a^3*b^2*c^3*d*j^2*k + 2*a^2*b^3*c^3*f^2*h*k - 2*a^2*b^3*c^3*f^2*g*l - 10*a
^3*b^3*c^2*e*g*l^2 + 10*a^3*b^3*c^2*d*h*l^2 - 8*a^2*b^2*c^4*e^2*h*k - 8*a^2
*b^2*c^4*e^2*f*m + 4*a^2*b^3*c^3*e*g^2*l + 4*a^2*b^2*c^4*e^2*g*l + 2*a^3*b^
2*c^3*f*h*j^2 + 28*a^3*b^3*c^2*d*f*m^2 + 14*a^2*b^2*c^4*d*f^2*m - 8*a^3*b^2
*c^3*e*g*k^2 + 4*a^3*b^2*c^3*d*h*k^2 + 4*a^2*b^3*c^3*d*h^2*k + 2*a^2*b^4*c^
2*e*g*k^2 - 2*a^2*b^4*c^2*d*h*k^2 + 2*a^2*b^2*c^4*f^2*g*j + 2*a^2*b^2*c^4*e

*f^2*l + 18*a^3*b^2*c^3*d*f*l^2 - 12*a^2*b^4*c^2*d*f*l^2 - 4*a^2*b^2*c^4*e*
g^2*j + 2*a^2*b^3*c^3*e*g*j^2 - 2*a^2*b^3*c^3*d*h*j^2 - 10*a^2*b^3*c^3*d*f*
k^2 - 8*a^2*b^2*c^4*d*f*j^2 + 2*a^2*b^2*c^4*e*g*h^2 + 4*a^5*b^2*c*h^2*m^2 -
2*a^4*b^2*c^2*h^3*m - 5*a^5*b*c^2*g^2*m^2 + 5*a^4*b^3*c*g^2*m^2 + 3*a^5*b*
c^2*h^2*l^2 + 6*a^3*b^4*c*f^2*m^2 - 2*a^3*b^2*c^3*g^3*l + 2*a^2*b^3*c^3*f^3
*m + 7*a^4*b*c^3*e^2*m^2 + 7*a^2*b^5*c*e^2*m^2 - 5*a^4*b*c^3*f^2*l^2 + 3*a^
4*b*c^3*g^2*k^2 - 2*a^4*b^2*c^2*f*k^3 - 2*a^2*b^2*c^4*f^3*k + 7*a^3*b*c^4*d
^2*l^2 + 7*a*b^5*c^2*d^2*l^2 - 5*a^3*b*c^4*e^2*k^2 + 3*a^3*b*c^4*f^2*j^2 +
6*a*b^4*c^3*d^2*k^2 + 2*a^3*b^3*c^2*d*k^3 - 2*a^3*b^2*c^3*e*j^3 - 5*a^2*b*c
^5*d^2*j^2 + 5*a*b^3*c^4*d^2*j^2 + 3*a^2*b*c^5*e^2*h^2 + 4*a*b^2*c^5*d^2*h^
2 - 2*a^2*b^2*c^4*d*h^3 - 4*a^6*c^2*j^2*k*m + 2*a^6*b^2*j*l*m^2 + 4*a^6*c^2
*j*k^2*l + 4*a^6*c^2*h*k^2*m - 4*a^6*c^2*h*k*l^2 - 4*a^6*c^2*f*l^2*m + 4*a^
5*c^3*g^2*k*m + 2*a^5*b^3*h*k*m^2 - 2*a^5*b^3*g*l*m^2 + 4*a^6*c^2*g*j*m^2 +
4*a^6*c^2*f*k*m^2 + 4*a^6*c^2*e*l*m^2 - 4*a^5*c^3*h^2*j*l + 4*a^5*c^3*h*j^
2*k + 4*a^5*c^3*g*j^2*l + 4*a^5*c^3*f*j^2*m - 4*a^4*c^4*e^2*k*m + 2*a^4*b^4
*g*j*m^2 - 2*a^4*b^4*f*k*m^2 + 2*a^4*b^4*e*l*m^2 - 4*a^5*c^3*g*j*k^2 - 4*a^
5*c^3*e*k^2*l - 4*a^5*c^3*d*k^2*m + 4*a^4*c^4*f^2*j*l + 4*a^5*c^3*e*j*l^2 +
4*a^5*c^3*d*k*l^2 + 4*a^4*c^4*f^2*h*m + 2*b^6*c^2*d^2*j*l - 2*a^3*b^5*e*j*
m^2 + 2*a^3*b^5*d*k*m^2 + 4*a^5*c^3*f*h*l^2 - 4*a^4*c^4*g^2*h*k - 4*a^4*c^4
*f*g^2*m - 4*a^3*c^5*d^2*j*l - 2*b^6*c^2*d^2*h*m + 2*a^3*b^5*f*h*m^2 + 12*a
^5*c^3*d*h*m^2 - 12*a^4*c^4*d*h^2*m + 12*a^3*c^5*d^2*h*m - 4*a^5*c^3*e*g*m^
2 + 4*a^4*c^4*g*h^2*j + 4*a^4*c^4*f*h^2*k + 4*a^4*c^4*e*h^2*l - 4*a^4*c^4*d
*j^2*k + 3*a^6*b*c*j^2*m^2 - 4*a^4*c^4*f*h*j^2 + 4*a^3*c^5*e^2*h*k + 4*a^3*
c^5*e^2*g*l + 4*a^3*c^5*e^2*f*m + 2*b^5*c^3*d^2*h*k - 2*b^5*c^3*d^2*g*l + 2
*b^5*c^3*d^2*f*m + 2*a^5*b*c^2*j^3*l + 2*a^2*b^6*e*g*m^2 - 2*a^2*b^6*d*h*m^
2 + 4*a^4*c^4*e*g*k^2 + 4*a^4*c^4*d*h*k^2 - 4*a^3*c^5*f^2*g*j - 4*a^3*c^5*e
*f^2*l - 4*a^3*c^5*d*f^2*m - 4*a^4*c^4*d*f*l^2 + 4*a^3*c^5*e*g^2*j + 4*a^3*
c^5*d*g^2*k + 2*b^4*c^4*d^2*g*j - 2*b^4*c^4*d^2*f*k + 2*b^4*c^4*d^2*e*l - 6
*a^3*b*c^4*f^3*m + 4*a^3*c^5*f*g^2*h + 4*a^2*c^6*d^2*g*j + 4*a^2*c^6*d^2*f*
k + 4*a^2*c^6*d^2*e*l - 2*a^5*b^2*c*g*l^3 + 2*a^5*b*c^2*h*k^3 + 2*a^4*b*c^3
*h^3*k - 4*a^3*c^5*e*g*h^2 + 4*a^3*c^5*d*f*j^2 - 4*a^2*c^6*d*e^2*k - 2*b^3*
c^5*d^2*e*j + 8*a^5*b^2*c*d*m^3 + 8*a*b^6*c*d^2*m^2 + 8*a*b^2*c^5*d^3*m - 6
*a^5*b*c^2*e*l^3 - 6*a^2*b*c^5*e^3*l - 4*a^2*c^6*e^2*f*h + 2*b^3*c^5*d^2*f*
h + 2*a^4*b^3*c*e*l^3 + 2*a^4*b*c^3*g*j^3 + 2*a^3*b*c^4*g^3*j + 2*a*b^3*c^4
*e^3*l + 4*a^2*c^6*e*f^2*g + 4*a^2*c^6*d*f^2*h - 6*a^4*b*c^3*d*k^3 - 4*a^2*
c^6*d*f*g^2 + 2*b^2*c^6*d^2*e*g - 2*a*b^2*c^5*e^3*j + 2*a^3*b*c^4*f*h^3 + 2
*a^2*b*c^5*f^3*h + 2*a^2*b*c^5*e*g^3 + 3*a*b*c^6*d^2*g^2 - 9*a^4*b^2*c^2*f^
2*m^2 + 4*a^4*b^2*c^2*g^2*l^2 - 14*a^3*b^3*c^2*e^2*m^2 + 5*a^3*b^3*c^2*f^2*
l^2 - 20*a^2*b^4*c^2*d^2*m^2 + 16*a^3*b^2*c^3*d^2*m^2 - 9*a^3*b^2*c^3*e^2*l
^2 + 6*a^2*b^4*c^2*e^2*l^2 + 4*a^3*b^2*c^3*f^2*k^2 - 14*a^2*b^3*c^3*d^2*l^2
+ 5*a^2*b^3*c^3*e^2*k^2 - 9*a^2*b^2*c^4*d^2*k^2 + 4*a^2*b^2*c^4*e^2*j^2 +
4*a^7*c*k*l^2*m - 4*a^7*c*j*l*m^2 + 2*b^7*c*d^2*k*m + 2*a^6*b*c*k^3*m + 2*a
^6*b*c*j*l^3 + 2*a*b^7*d*f*m^2 - 6*a^6*b*c*f*m^3 - 6*a*b*c^6*d^3*k - 4*a*c^
7*d^2*e*g + 4*a*c^7*d*e^2*f + 2*a*b*c^6*e^3*g + 2*a*b*c^6*d*f^3 - a^5*b^2*c
*j^2*l^2 - a^5*b*c^2*j^2*k^2 - a^4*b^3*c*h^2*l^2 - a^3*b^4*c*g^2*l^2 - a^4*

b*c^3*h^2*j^2 - a^2*b^5*c*f^2*l^2 - a*b^5*c^2*e^2*k^2 - a^3*b*c^4*g^2*h^2 - a*b^4*c^3*e^2*j^2 - a^2*b*c^5*f^2*g^2 - a*b^3*c^4*e^2*h^2 - a*b^2*c^5*e^2*g^2 + 2*a^7*b*k*m^3 + 4*a^7*c*h*m^3 + 4*a*c^7*d^3*h + 2*b*c^7*d^3*f - a^6*b*c*k^2*l^2 - 2*a^6*c^2*j^2*l^2 - 6*a^6*c^2*h^2*m^2 - a*b^6*c*e^2*l^2 - 6*a^5*c^3*g^2*l^2 - 2*a^5*c^3*h^2*k^2 - 2*a^5*c^3*f^2*m^2 - 6*a^4*c^4*f^2*k^2 - 6*a^4*c^4*d^2*m^2 - 2*a^4*c^4*g^2*j^2 - 2*a^4*c^4*e^2*l^2 - 6*a^3*c^5*e^2*j^2 - 2*a^3*c^5*d^2*k^2 - 2*a^3*c^5*f^2*h^2 - a*b*c^6*e^2*f^2 - 6*a^2*c^6*d^2*h^2 - 2*a^2*c^6*e^2*g^2 - a^4*b^2*c^2*h^2*k^2 - a^3*b^3*c^2*g^2*k^2 - a^3*b^2*c^3*g^2*j^2 - a^2*b^4*c^2*f^2*k^2 - a^2*b^3*c^3*f^2*j^2 - a^2*b^2*c^4*f^2*h^2 - 2*a^7*c*k^2*m^2 + 4*a^5*c^3*h^3*m - 2*a^6*b^2*h*m^3 + 4*a^6*c^2*g*l^3 + 4*a^4*c^4*g^3*l - 2*b^4*c^4*d^3*m + 2*a^5*b^3*f*m^3 - 4*a^6*c^2*d*m^3 + 4*a^5*c^3*f*k^3 + 4*a^3*c^5*f^3*k - 4*a^2*c^6*d^3*m + 2*b^3*c^5*d^3*k - 2*a^4*b^4*d*m^3 + 4*a^4*c^4*e*j^3 + 4*a^2*c^6*e^3*j - 2*b^2*c^6*d^3*h + 4*a^3*c^5*d*h^3 - 2*a*c^7*d^2*f^2 - a^6*b^2*k^2*m^2 - a^5*b^3*j^2*m^2 - a^4*b^4*h^2*m^2 - a^3*b^5*g^2*m^2 - a^2*b^6*f^2*m^2 - b^6*c^2*d^2*k^2 - b^5*c^3*d^2*j^2 - b^4*c^4*d^2*h^2 - b^3*c^5*d^2*g^2 - b^2*c^6*d^2*f^2 - a^7*b*l^2*m^2 - b^7*c*d^2*l^2 - a*b^7*e^2*m^2 - b*c^7*d^2*e^2 - b^8*d^2*m^2 - a^6*c^2*k^4 - a^5*c^3*j^4 - a^4*c^4*h^4 - a^3*c^5*g^4 - a^2*c^6*f^4 - a^7*c*l^4 - a*c^7*e^4 - a^8*m^4 - c^8*d^4, z, k1)*((16*a^3*c^6*m - 16*a^2*c^7*h - 4*b^2*c^7*d + 16*a*c^8*d - 20*a^2*b^2*c^5*m + 4*a*b^2*c^6*h - 4*a*b^3*c^5*k + 16*a^2*b*c^6*k + 4*a*b^4*c^4*m)/c^5 + (x*(4*b^2*c^7*e - 8*b^3*c^6*g + 16*a^2*c^7*j + 8*b^4*c^5*j - 8*b^5*c^4*l - 16*a*c^8*e + 32*a*b*c^7*g - 36*a*b^2*c^6*j + 44*a*b^3*c^5*l - 48*a^2*b*c^6*l))/c^5 + (root(128*a^2*b^2*c^8*z^4 - 16*a*b^4*c^7*z^4 - 256*a^3*c^9*z^4 + 384*a^3*b^2*c^6*l*z^3 - 144*a^2*b^4*c^5*l*z^3 + 128*a^2*b^3*c^6*j*z^3 - 128*a^2*b^2*c^7*g*z^3 + 16*a*b^6*c^4*l*z^3 - 256*a^3*b*c^7*j*z^3 - 16*a*b^5*c^5*j*z^3 + 16*a*b^4*c^6*g*z^3 - 256*a^4*c^7*l*z^3 + 256*a^3*c^8*g*z^3 - 96*a^4*b*c^5*j*l*z^2 + 8*a*b^7*c^2*j*l*z^2 + 160*a^4*b*c^5*h*m*z^2 - 8*a*b^7*c^2*h*m*z^2 + 8*a*b^6*c^3*h*k*z^2 - 8*a*b^6*c^3*g*l*z^2 + 8*a*b^6*c^3*f*m*z^2 + 160*a^3*b*c^6*g*j*z^2 - 96*a^3*b*c^6*f*k*z^2 - 96*a^3*b*c^6*e*l*z^2 - 96*a^3*b*c^6*d*m*z^2 + 8*a*b^5*c^4*g*j*z^2 - 8*a*b^5*c^4*f*k*z^2 - 8*a*b^5*c^4*e*l*z^2 - 8*a*b^5*c^4*d*m*z^2 + 8*a*b^4*c^5*e*j*z^2 + 8*a*b^4*c^5*d*k*z^2 + 8*a*b^4*c^5*f*h*z^2 + 32*a^2*b*c^7*e*g*z^2 + 32*a^2*b*c^7*d*h*z^2 - 8*a*b^3*c^6*e*g*z^2 - 8*a*b^3*c^6*d*h*z^2 + 16*a*b^2*c^7*d*f*z^2 + 8*a*b^8*c*k*m*z^2 - 304*a^4*b^2*c^4*k*m*z^2 + 264*a^3*b^4*c^3*k*m*z^2 - 80*a^2*b^6*c^2*k*m*z^2 + 184*a^3*b^3*c^4*j*l*z^2 - 72*a^2*b^5*c^3*j*l*z^2 - 200*a^3*b^3*c^4*h*m*z^2 + 72*a^2*b^5*c^3*h*m*z^2 - 240*a^3*b^2*c^5*g*l*z^2 + 144*a^3*b^2*c^5*h*k*z^2 + 144*a^3*b^2*c^5*f*m*z^2 + 80*a^2*b^4*c^4*g*l*z^2 - 64*a^2*b^4*c^4*h*k*z^2 - 64*a^2*b^4*c^4*f*m*z^2 - 72*a^2*b^3*c^5*g*j*z^2 + 56*a^2*b^3*c^5*f*k*z^2 + 56*a^2*b^3*c^5*e*l*z^2 + 56*a^2*b^3*c^5*d*m*z^2 - 48*a^2*b^2*c^6*e*j*z^2 - 48*a^2*b^2*c^6*d*k*z^2 - 48*a^2*b^2*c^6*f*h*z^2 - 112*a^5*b*c^4*m^2*z^2 + 44*a^2*b^7*c*m^2*z^2 + 80*a^4*b*c^5*k^2*z^2 - 4*a*b^7*c^2*k^2*z^2 - 4*a*b^6*c^3*j^2*z^2 - 48*a^3*b*c^6*h^2*z^2 - 4*a*b^5*c^4*h^2*z^2 - 4*a*b^4*c^5*g^2*z^2 + 16*a^2*b*c^7*f^2*z^2 - 4*a*b^3*c^6*f^2*z^2 + 8*a*b^2*c^7*e^2*z^2 + 64*a^5*c^5*k*m*z^2 + 192*a^4*c^6*g*l*z^2 - 64*a^4*c^6*h*k*z^2 - 64*a^4*c^6*f*m*z^2 + 64*a^3*c^7*e*j*z

^2 + 64*a^3*c^7*d*k*z^2 + 64*a^3*c^7*f*h*z^2 - 4*a*b^8*c*l^2*z^2 - 64*a^2*c
^8*d*f*z^2 + 16*a*b*c^8*d^2*z^2 + 252*a^4*b^3*c^3*m^2*z^2 - 168*a^3*b^5*c^2
*m^2*z^2 + 168*a^4*b^2*c^4*l^2*z^2 - 132*a^3*b^4*c^3*l^2*z^2 + 40*a^2*b^6*c
^2*l^2*z^2 - 100*a^3*b^3*c^4*k^2*z^2 + 36*a^2*b^5*c^3*k^2*z^2 - 56*a^3*b^2*
c^5*j^2*z^2 + 32*a^2*b^4*c^4*j^2*z^2 + 28*a^2*b^3*c^5*h^2*z^2 + 40*a^2*b^2*
c^6*g^2*z^2 - 96*a^5*c^5*l^2*z^2 - 32*a^4*c^6*j^2*z^2 - 96*a^3*c^7*g^2*z^2
- 32*a^2*c^8*e^2*z^2 - 4*b^3*c^7*d^2*z^2 - 4*a*b^9*m^2*z^2 + 32*a^5*b*c^3*h
*l*m*z + 8*a^2*b^6*c*g*k*m*z + 96*a^4*b*c^4*e*k*m*z + 32*a^4*b*c^4*h*j*k*z
+ 32*a^4*b*c^4*g*j*l*z + 32*a^4*b*c^4*f*j*m*z - 64*a^4*b*c^4*g*h*m*z - 8*a*
b^6*c^2*e*j*l*z + 8*a*b^6*c^2*e*h*m*z - 64*a^3*b*c^5*e*h*k*z + 64*a^3*b*c^5
*e*g*l*z - 64*a^3*b*c^5*e*f*m*z + 32*a^3*b*c^5*f*g*k*z - 32*a^3*b*c^5*d*h*l
*z + 32*a^3*b*c^5*d*g*m*z - 8*a*b^5*c^3*e*h*k*z + 8*a*b^5*c^3*e*g*l*z - 8*a
*b^5*c^3*e*f*m*z - 8*a*b^4*c^4*e*g*j*z + 8*a*b^4*c^4*e*f*k*z - 8*a*b^4*c^4*
d*f*l*z + 8*a*b^4*c^4*d*e*m*z - 32*a^2*b*c^6*d*f*j*z + 32*a^2*b*c^6*d*e*k*z
 + 8*a*b^3*c^5*d*f*j*z - 8*a*b^3*c^5*d*e*k*z + 32*a^2*b*c^6*e*f*h*z - 8*a*b
^3*c^5*e*f*h*z - 8*a*b^2*c^6*d*f*g*z + 8*a*b^2*c^6*d*e*h*z - 8*a*b^7*c*e*k*
m*z - 40*a^5*b^2*c^2*k*l*m*z + 48*a^4*b^3*c^2*j*k*m*z - 8*a^4*b^3*c^2*h*l*m
*z + 104*a^4*b^2*c^3*g*k*m*z - 56*a^3*b^4*c^2*g*k*m*z - 40*a^4*b^2*c^3*h*j*
m*z + 8*a^4*b^2*c^3*h*k*l*z + 8*a^4*b^2*c^3*f*l*m*z + 8*a^3*b^4*c^2*h*j*m*z
 - 152*a^3*b^3*c^3*e*k*m*z + 64*a^2*b^5*c^2*e*k*m*z - 40*a^3*b^3*c^3*g*j*l*
z - 8*a^3*b^3*c^3*h*j*k*z - 8*a^3*b^3*c^3*f*j*m*z + 8*a^2*b^5*c^2*g*j*l*z +
 48*a^3*b^3*c^3*g*h*m*z - 8*a^2*b^5*c^2*g*h*m*z - 104*a^3*b^2*c^4*e*j*l*z +
 56*a^2*b^4*c^3*e*j*l*z + 8*a^3*b^2*c^4*f*j*k*z - 8*a^3*b^2*c^4*d*k*l*z + 8
*a^3*b^2*c^4*d*j*m*z + 104*a^3*b^2*c^4*e*h*m*z - 56*a^2*b^4*c^3*e*h*m*z - 4
0*a^3*b^2*c^4*g*h*k*z - 40*a^3*b^2*c^4*f*g*m*z - 8*a^3*b^2*c^4*f*h*l*z + 8*
a^2*b^4*c^3*g*h*k*z + 8*a^2*b^4*c^3*f*g*m*z + 48*a^2*b^3*c^4*e*h*k*z - 48*a
^2*b^3*c^4*e*g*l*z + 48*a^2*b^3*c^4*e*f*m*z - 8*a^2*b^3*c^4*f*g*k*z + 8*a^2
*b^3*c^4*d*h*l*z - 8*a^2*b^3*c^4*d*g*m*z + 40*a^2*b^2*c^5*e*g*j*z - 40*a^2*
b^2*c^5*e*f*k*z + 40*a^2*b^2*c^5*d*f*l*z - 40*a^2*b^2*c^5*d*e*m*z - 8*a^2*b
^2*c^5*d*h*j*z + 8*a^2*b^2*c^5*d*g*k*z + 8*a^2*b^2*c^5*f*g*h*z + 8*a^4*b^4*
c*k*l*m*z - 64*a^5*b*c^3*j*k*m*z - 8*a^3*b^5*c*j*k*m*z - 32*a^6*b*c^2*l*m^2
*z + 24*a^5*b^3*c*l*m^2*z - 28*a^4*b^4*c*j*m^2*z + 16*a^5*b*c^3*k^2*l*z + 4
*a^3*b^5*c*j*l^2*z + 48*a^5*b*c^3*g*m^2*z + 32*a^3*b^5*c*g*m^2*z - 4*a^2*b^
6*c*g*l^2*z - 36*a^2*b^6*c*e*m^2*z - 32*a^4*b*c^4*g*k^2*z - 16*a^3*b*c^5*f^
2*l*z - 48*a^4*b*c^4*e*l^2*z - 32*a^3*b*c^5*g^2*j*z - 4*a*b^4*c^4*e^2*l*z +
 32*a^2*b*c^6*d^2*l*z - 24*a*b^3*c^5*d^2*l*z + 4*a*b^6*c^2*e*k^2*z + 32*a^3
*b*c^5*e*j^2*z + 16*a^3*b*c^5*g*h^2*z - 16*a^2*b*c^6*e^2*j*z + 4*a*b^5*c^3*
e*j^2*z + 4*a*b^3*c^5*e^2*j*z + 20*a*b^2*c^6*d^2*j*z + 4*a*b^4*c^4*e*h^2*z
- 16*a^2*b*c^6*e*g^2*z + 4*a*b^3*c^5*e*g^2*z - 4*a*b^2*c^6*e^2*g*z + 4*a*b^
2*c^6*e*f^2*z + 32*a^6*c^3*k*l*m*z - 32*a^5*c^4*h*k*l*z + 32*a^5*c^4*h*j*m*
z - 32*a^5*c^4*g*k*m*z - 32*a^5*c^4*f*l*m*z - 32*a^4*c^5*f*j*k*z + 32*a^4*c
^5*e*j*l*z + 32*a^4*c^5*d*k*l*z - 32*a^4*c^5*d*j*m*z + 32*a^4*c^5*g*h*k*z +
 32*a^4*c^5*f*h*l*z + 32*a^4*c^5*f*g*m*z - 32*a^4*c^5*e*h*m*z - 32*a^3*c^6*
e*g*j*z + 32*a^3*c^6*e*f*k*z + 32*a^3*c^6*d*h*j*z - 32*a^3*c^6*d*g*k*z - 32
*a^3*c^6*d*f*l*z + 32*a^3*c^6*d*e*m*z - 32*a^3*c^6*f*g*h*z + 4*a*b^7*c*e*l^

2*z + 32*a^2*c^7*d*f*g*z - 32*a^2*c^7*d*e*h*z - 16*a*b*c^7*d^2*g*z + 52*a^5
*b^2*c^2*j*m^2*z - 4*a^4*b^3*c^2*k^2*l*z + 36*a^4*b^2*c^3*j^2*l*z - 16*a^4*
b^3*c^2*j*l^2*z - 8*a^3*b^4*c^2*j^2*l*z - 20*a^4*b^2*c^3*j*k^2*z + 4*a^3*b^
4*c^2*j*k^2*z - 76*a^4*b^3*c^2*g*m^2*z - 60*a^4*b^2*c^3*g*l^2*z + 44*a^3*b^
2*c^4*g^2*l*z + 28*a^3*b^4*c^2*g*l^2*z - 8*a^2*b^4*c^3*g^2*l*z + 104*a^3*b^
4*c^2*e*m^2*z - 100*a^4*b^2*c^3*e*m^2*z + 24*a^3*b^3*c^3*g*k^2*z + 4*a^3*b^
2*c^4*h^2*j*z - 4*a^2*b^5*c^2*g*k^2*z + 4*a^2*b^3*c^4*f^2*l*z + 76*a^3*b^3*
c^3*e*l^2*z - 32*a^2*b^5*c^2*e*l^2*z + 20*a^2*b^2*c^5*e^2*l*z + 12*a^3*b^2*
c^4*g*j^2*z + 8*a^2*b^3*c^4*g^2*j*z - 4*a^2*b^4*c^3*g*j^2*z + 52*a^3*b^2*c^
4*e*k^2*z - 28*a^2*b^4*c^3*e*k^2*z - 4*a^2*b^2*c^5*f^2*j*z - 24*a^2*b^3*c^4
*e*j^2*z - 4*a^2*b^3*c^4*g*h^2*z - 20*a^2*b^2*c^5*e*h^2*z + 20*a^5*b^2*c^2*
l^3*z + 4*a^3*b^3*c^3*j^3*z - 4*a^2*b^2*c^5*g^3*z - 4*a^4*b^5*l*m^2*z - 16*
a^6*c^3*j*m^2*z - 16*a^5*c^4*j^2*l*z + 4*a^3*b^6*j*m^2*z + 16*a^5*c^4*j*k^2
*z + 48*a^5*c^4*g*l^2*z - 48*a^4*c^5*g^2*l*z - 4*a^2*b^7*g*m^2*z + 16*a^5*c
^4*e*m^2*z - 16*a^4*c^5*h^2*j*z + 16*a^4*c^5*g*j^2*z - 16*a^3*c^6*e^2*l*z +
4*b^5*c^4*d^2*l*z - 16*a^4*c^5*e*k^2*z + 16*a^3*c^6*f^2*j*z - 4*b^4*c^5*d^
2*j*z - 16*a^2*c^7*d^2*j*z - 4*a^4*b^4*c*l^3*z + 16*a^3*c^6*e*h^2*z - 16*a^
4*b*c^4*j^3*z + 16*a^2*c^7*e^2*g*z + 4*b^3*c^6*d^2*g*z - 16*a^2*c^7*e*f^2*z
- 4*b^2*c^7*d^2*e*z + 4*a*b^8*e*m^2*z + 16*a*c^8*d^2*e*z - 16*a^6*c^3*l^3*
z + 16*a^3*c^6*g^3*z + 4*a^5*b^2*c*g*k*l*m + 12*a^5*b*c^2*g*j*k*m + 12*a^5*
b*c^2*e*k*l*m - 4*a^5*b*c^2*h*j*k*l - 4*a^5*b*c^2*f*j*l*m - 4*a^4*b^3*c*g*j
*k*m - 4*a^4*b^3*c*e*k*l*m - 4*a^5*b*c^2*g*h*l*m + 4*a^3*b^4*c*e*j*k*m - 4*
a^3*b^4*c*f*h*k*m + 12*a^4*b*c^3*d*j*k*l - 20*a^4*b*c^3*e*g*k*m + 12*a^4*b*
c^3*f*h*j*l + 12*a^4*b*c^3*e*h*j*m + 12*a^4*b*c^3*d*h*k*m - 4*a^4*b*c^3*g*h
*j*k - 4*a^4*b*c^3*f*g*k*l - 4*a^4*b*c^3*f*g*j*m - 4*a^4*b*c^3*e*h*k*l - 4*
a^4*b*c^3*e*f*l*m - 4*a^4*b*c^3*d*g*l*m - 4*a^2*b^5*c*e*g*k*m + 4*a^2*b^5*c
*d*h*k*m - 20*a^3*b*c^4*d*f*j*l - 4*a^3*b*c^4*e*f*j*k - 4*a^3*b*c^4*d*g*j*k
- 4*a^3*b*c^4*d*e*k*l - 4*a^3*b*c^4*d*e*j*m - 4*a*b^5*c^2*d*f*j*l + 12*a^3
*b*c^4*e*g*h*k + 12*a^3*b*c^4*e*f*g*m + 12*a^3*b*c^4*d*g*h*l + 12*a^3*b*c^4
*d*f*h*m - 4*a^3*b*c^4*f*g*h*j - 4*a^3*b*c^4*e*f*h*l + 4*a*b^5*c^2*d*f*h*m
- 4*a*b^4*c^3*d*f*h*k + 4*a*b^4*c^3*d*f*g*l + 12*a^2*b*c^5*d*f*g*j + 12*a^2
*b*c^5*d*e*f*l - 4*a^2*b*c^5*d*e*h*j - 4*a^2*b*c^5*d*e*g*k - 4*a*b^3*c^4*d*
f*g*j - 4*a*b^3*c^4*d*e*f*l - 4*a^2*b*c^5*e*f*g*h + 4*a*b^2*c^5*d*e*f*j - 4
*a^6*b*c*j*k*l*m - 4*a*b^6*c*d*f*k*m - 4*a*b*c^6*d*e*f*g - 16*a^4*b^2*c^2*e
*j*k*m + 4*a^4*b^2*c^2*f*j*k*l + 4*a^4*b^2*c^2*d*j*l*m + 12*a^4*b^2*c^2*f*h
*k*m + 4*a^4*b^2*c^2*g*h*j*m + 4*a^4*b^2*c^2*e*h*l*m - 4*a^3*b^3*c^2*d*j*k*
l + 20*a^3*b^3*c^2*e*g*k*m - 16*a^3*b^3*c^2*d*h*k*m - 4*a^3*b^3*c^2*f*h*j*l
- 4*a^3*b^3*c^2*e*h*j*m - 40*a^3*b^2*c^3*d*f*k*m + 24*a^2*b^4*c^2*d*f*k*m
- 16*a^3*b^2*c^3*d*h*j*l + 12*a^3*b^2*c^3*e*g*j*l + 4*a^3*b^2*c^3*e*h*j*k +
4*a^3*b^2*c^3*e*f*j*m + 4*a^3*b^2*c^3*d*g*k*l - 4*a^2*b^4*c^2*e*g*j*l + 4*
a^2*b^4*c^2*d*h*j*l - 16*a^3*b^2*c^3*e*g*h*m + 4*a^3*b^2*c^3*f*g*h*l + 4*a^
2*b^4*c^2*e*g*h*m + 20*a^2*b^3*c^3*d*f*j*l - 16*a^2*b^3*c^3*d*f*h*m - 4*a^2
*b^3*c^3*e*g*h*k - 4*a^2*b^3*c^3*e*f*g*m - 4*a^2*b^3*c^3*d*g*h*l - 16*a^2*b
^2*c^4*d*f*g*l + 12*a^2*b^2*c^4*d*f*h*k + 4*a^2*b^2*c^4*e*f*g*k + 4*a^2*b^2
*c^4*d*g*h*j + 4*a^2*b^2*c^4*d*e*h*l + 4*a^2*b^2*c^4*d*e*g*m + 2*a^5*b^2*c*

j^2*k*m - 4*a^5*b^2*c*h*k^2*m - 2*a^5*b*c^2*h^2*k*m + 2*a^4*b^3*c*h^2*k*m + 2*a^5*b^2*c*h*k*l^2 + 2*a^5*b^2*c*f*l^2*m - 2*a^5*b*c^2*h*j^2*m + 2*a^3*b^4*c*g^2*k*m + 14*a^4*b*c^3*f^2*k*m - 10*a^5*b*c^2*f*k^2*m - 8*a^5*b^2*c*g*j*m^2 - 8*a^5*b^2*c*e*l*m^2 + 4*a^5*b^2*c*f*k*m^2 + 4*a^4*b^3*c*f*k^2*m - 2*a^5*b*c^2*g*k^2*l + 2*a^2*b^5*c*f^2*k*m + 6*a^5*b*c^2*f*k*l^2 + 6*a^5*b*c^2*d*l^2*m - 2*a^5*b*c^2*g*j*l^2 + 2*a^4*b^3*c*g*j*l^2 - 2*a^4*b^3*c*f*k*l^2 - 2*a^4*b^3*c*d*l^2*m - 2*a^4*b*c^3*g^2*j*l - 14*a*b^5*c^2*d^2*k*m - 10*a^5*b*c^2*e*j*m^2 + 10*a^4*b^3*c*e*j*m^2 - 10*a^3*b*c^4*d^2*k*m - 6*a^4*b^3*c*d*k*m^2 + 6*a^4*b*c^3*g^2*h*m - 4*a^3*b^4*c*d*k^2*m - 2*a^5*b*c^2*d*k*m^2 + 14*a^5*b*c^2*f*h*m^2 + 14*a^3*b*c^4*e^2*j*l - 10*a^4*b^3*c*f*h*m^2 - 10*a^4*b*c^3*f*h^2*m - 10*a^4*b*c^3*e*j^2*l - 2*a^4*b*c^3*g*h^2*l - 2*a^4*b*c^3*f*j^2*k - 2*a^4*b*c^3*d*j^2*m - 2*a^3*b^4*c*e*j*l^2 + 2*a^3*b^4*c*d*k*l^2 + 2*a*b^5*c^2*e^2*j*l - 12*a*b^4*c^3*d^2*j*l - 10*a^3*b*c^4*e^2*h*m + 6*a^4*b*c^3*e*j*k^2 + 2*a^3*b^4*c*f*h*l^2 - 2*a*b^5*c^2*e^2*h*m - 12*a^3*b^4*c*e*g*m^2 + 12*a^3*b^4*c*d*h*m^2 + 12*a*b^4*c^3*d^2*h*m + 6*a^3*b*c^4*f^2*g*l - 2*a^4*b*c^3*f*h*k^2 - 2*a^3*b*c^4*f^2*h*k + 14*a^4*b*c^3*e*g*l^2 - 10*a^4*b*c^3*d*h*l^2 - 10*a^3*b*c^4*e*g^2*l - 2*a^3*b*c^4*f*g^2*k - 2*a^3*b*c^4*d*g^2*m + 2*a^2*b^5*c*e*g*l^2 - 2*a^2*b^5*c*d*h*l^2 + 2*a*b^4*c^3*e^2*h*k - 2*a*b^4*c^3*e^2*g*l + 2*a*b^4*c^3*e^2*f*m - 14*a^2*b^5*c*d*f*m^2 + 14*a^2*b*c^5*d^2*h*k - 10*a^4*b*c^3*d*f*m^2 - 10*a^3*b*c^4*d*h^2*k - 10*a^2*b*c^5*d^2*g*l - 10*a*b^3*c^4*d^2*h*k + 10*a*b^3*c^4*d^2*g*l - 6*a*b^3*c^4*d^2*f*m - 4*a*b^4*c^3*d*f^2*m - 2*a^3*b*c^4*e*h^2*j - 2*a^2*b*c^5*d^2*f*m + 6*a^3*b*c^4*d*h*j^2 + 6*a^2*b*c^5*e^2*f*k + 6*a^2*b*c^5*d*e^2*m - 2*a^3*b*c^4*e*g*j^2 - 2*a^2*b*c^5*e^2*g*j + 2*a*b^3*c^4*e^2*g*j - 2*a*b^3*c^4*e^2*f*k - 2*a*b^3*c^4*d*e^2*m + 14*a^3*b*c^4*d*f*k^2 - 10*a^2*b*c^5*d*f^2*k - 8*a*b^2*c^5*d^2*g*j - 8*a*b^2*c^5*d^2*e*l + 4*a*b^3*c^4*d*f^2*k + 4*a*b^2*c^5*d^2*f*k - 2*a^2*b*c^5*e*f^2*j + 2*a*b^5*c^2*d*f*k^2 + 2*a*b^4*c^3*d*f*j^2 + 2*a*b^2*c^5*d*e^2*k - 2*a^2*b*c^5*d*g^2*h + 2*a*b^2*c^5*e^2*f*h - 4*a*b^2*c^5*d*f^2*h - 2*a^2*b*c^5*d*f*h^2 + 2*a*b^3*c^4*d*f*h^2 + 2*a*b^2*c^5*d*f*g^2 + 8*a^6*c^2*h*j*l*m - 8*a^6*c^2*g*k*l*m - 8*a^5*c^3*f*j*k*l + 8*a^5*c^3*e*j*k*m - 8*a^5*c^3*d*j*l*m + 8*a^5*c^3*g*h*k*l - 8*a^5*c^3*g*h*j*m - 8*a^5*c^3*f*h*k*m + 8*a^5*c^3*f*g*l*m - 8*a^5*c^3*e*h*l*m - 2*a^6*b*c*h*l^2*m + 8*a^4*c^4*f*g*j*k - 8*a^4*c^4*e*h*j*k - 8*a^4*c^4*e*g*j*l + 8*a^4*c^4*e*f*k*l - 8*a^4*c^4*e*f*j*m + 8*a^4*c^4*d*h*j*l - 8*a^4*c^4*d*g*k*l + 8*a^4*c^4*d*g*j*m + 8*a^4*c^4*d*f*k*m + 8*a^4*c^4*d*e*l*m + 6*a^6*b*c*g*l*m^2 - 2*a^6*b*c*h*k*m^2 - 8*a^4*c^4*f*g*h*l + 8*a^4*c^4*e*g*h*m + 2*a*b^6*c*e^2*k*m + 8*a^3*c^5*d*e*j*k + 8*a^3*c^5*e*f*h*j - 8*a^3*c^5*e*f*g*k - 8*a^3*c^5*d*g*h*j - 8*a^3*c^5*d*f*h*k + 8*a^3*c^5*d*f*g*l - 8*a^3*c^5*d*e*h*l - 8*a^3*c^5*d*e*g*m - 8*a^2*c^6*d*e*f*j + 8*a^2*c^6*d*e*g*h + 2*a*b^6*c*d*f*l^2 + 6*a*b*c^6*d^2*e*j - 2*a*b*c^6*d^2*f*h - 2*a*b*c^6*d*e^2*h - 8*a^4*b^2*c^2*g^2*k*m - 10*a^3*b^3*c^2*f^2*k*m + 2*a^4*b^2*c^2*h^2*j*l + 18*a^3*b^2*c^3*e^2*k*m - 12*a^2*b^4*c^2*e^2*k*m - 4*a^4*b^2*c^2*g*j^2*l + 2*a^3*b^3*c^2*g^2*j*l + 28*a^2*b^3*c^3*d^2*k*m + 14*a^4*b^2*c^2*d*k^2*m - 8*a^3*b^2*c^3*f^2*j*l + 2*a^4*b^2*c^2*g*j*k^2 + 2*a^4*b^2*c^2*e*k^2*l - 2*a^3*b^3*c^2*g^2*h*m + 2*a^2*b^4*c^2*f^2*j*l - 10*a^2*b^3*c^3*e^2*j*l - 8*a^4*b^2*c^2*d*k*l^2 + 4*a^4*b^2*c^2

```
*e*j*l^2 + 4*a^3*b^3*c^2*f*h^2*m + 4*a^3*b^3*c^2*e*j^2*l + 4*a^3*b^2*c^3*f^
2*h*m - 2*a^2*b^4*c^2*f^2*h*m + 18*a^2*b^2*c^4*d^2*j*l + 10*a^2*b^3*c^3*e^2
*h*m - 8*a^4*b^2*c^2*f*h*l^2 - 2*a^3*b^3*c^2*e*j*k^2 + 2*a^3*b^2*c^3*g^2*h*
k + 2*a^3*b^2*c^3*f*g^2*m - 22*a^4*b^2*c^2*d*h*m^2 - 22*a^2*b^2*c^4*d^2*h*m
 + 18*a^4*b^2*c^2*e*g*m^2 + 16*a^3*b^2*c^3*d*h^2*m - 4*a^3*b^2*c^3*f*h^2*k
- 4*a^2*b^4*c^2*d*h^2*m + 2*a^3*b^3*c^2*f*h*k^2 + 2*a^3*b^2*c^3*d*j^2*k + 2
*a^2*b^3*c^3*f^2*h*k - 2*a^2*b^3*c^3*f^2*g*l - 10*a^3*b^3*c^2*e*g*l^2 + 10*
a^3*b^3*c^2*d*h*l^2 - 8*a^2*b^2*c^4*e^2*h*k - 8*a^2*b^2*c^4*e^2*f*m + 4*a^2
*b^3*c^3*e*g^2*l + 4*a^2*b^2*c^4*e^2*g*l + 2*a^3*b^2*c^3*f*h*j^2 + 28*a^3*b
^3*c^2*d*f*m^2 + 14*a^2*b^2*c^4*d*f^2*m - 8*a^3*b^2*c^3*e*g*k^2 + 4*a^3*b^2
*c^3*d*h*k^2 + 4*a^2*b^3*c^3*d*h^2*k + 2*a^2*b^4*c^2*e*g*k^2 - 2*a^2*b^4*c^
2*d*h*k^2 + 2*a^2*b^2*c^4*f^2*g*j + 2*a^2*b^2*c^4*e*f^2*l + 18*a^3*b^2*c^3*
d*f*l^2 - 12*a^2*b^4*c^2*d*f*l^2 - 4*a^2*b^2*c^4*e*g^2*j + 2*a^2*b^3*c^3*e*
g*j^2 - 2*a^2*b^3*c^3*d*h*j^2 - 10*a^2*b^3*c^3*d*f*k^2 - 8*a^2*b^2*c^4*d*f*
j^2 + 2*a^2*b^2*c^4*e*g*h^2 + 4*a^5*b^2*c*h^2*m^2 - 2*a^4*b^2*c^2*h^3*m - 5
*a^5*b*c^2*g^2*m^2 + 5*a^4*b^3*c*g^2*m^2 + 3*a^5*b*c^2*h^2*l^2 + 6*a^3*b^4*
c*f^2*m^2 - 2*a^3*b^2*c^3*g^3*l + 2*a^2*b^3*c^3*f^3*m + 7*a^4*b*c^3*e^2*m^2
 + 7*a^2*b^5*c*e^2*m^2 - 5*a^4*b*c^3*f^2*l^2 + 3*a^4*b*c^3*g^2*k^2 - 2*a^4*
b^2*c^2*f*k^3 - 2*a^2*b^2*c^4*f^3*k + 7*a^3*b*c^4*d^2*l^2 + 7*a*b^5*c^2*d^2
*l^2 - 5*a^3*b*c^4*e^2*k^2 + 3*a^3*b*c^4*f^2*j^2 + 6*a*b^4*c^3*d^2*k^2 + 2*
a^3*b^3*c^2*d*k^3 - 2*a^3*b^2*c^3*e*j^3 - 5*a^2*b*c^5*d^2*j^2 + 5*a*b^3*c^4
*d^2*j^2 + 3*a^2*b*c^5*e^2*h^2 + 4*a*b^2*c^5*d^2*h^2 - 2*a^2*b^2*c^4*d*h^3
- 4*a^6*c^2*j^2*k*m + 2*a^6*b^2*j*l*m^2 + 4*a^6*c^2*j*k^2*l + 4*a^6*c^2*h*k
^2*m - 4*a^6*c^2*h*k*l^2 - 4*a^6*c^2*f*l^2*m + 4*a^5*c^3*g^2*k*m + 2*a^5*b^
3*h*k*m^2 - 2*a^5*b^3*g*l*m^2 + 4*a^6*c^2*g*j*m^2 + 4*a^6*c^2*f*k*m^2 + 4*a
^6*c^2*e*l*m^2 - 4*a^5*c^3*h^2*j*l + 4*a^5*c^3*h*j^2*k + 4*a^5*c^3*g*j^2*l
+ 4*a^5*c^3*f*j^2*m - 4*a^4*c^4*e^2*k*m + 2*a^4*b^4*g*j*m^2 - 2*a^4*b^4*f*k
*m^2 + 2*a^4*b^4*e*l*m^2 - 4*a^5*c^3*g*j*k^2 - 4*a^5*c^3*e*k^2*l - 4*a^5*c^
3*d*k^2*m + 4*a^4*c^4*f^2*j*l + 4*a^5*c^3*e*j*l^2 + 4*a^5*c^3*d*k*l^2 + 4*a
^4*c^4*f^2*h*m + 2*b^6*c^2*d^2*j*l - 2*a^3*b^5*e*j*m^2 + 2*a^3*b^5*d*k*m^2
+ 4*a^5*c^3*f*h*l^2 - 4*a^4*c^4*g^2*h*k - 4*a^4*c^4*f*g^2*m - 4*a^3*c^5*d^2
*j*l - 2*b^6*c^2*d^2*h*m + 2*a^3*b^5*f*h*m^2 + 12*a^5*c^3*d*h*m^2 - 12*a^4*
c^4*d*h^2*m + 12*a^3*c^5*d^2*h*m - 4*a^5*c^3*e*g*m^2 + 4*a^4*c^4*g*h^2*j +
4*a^4*c^4*f*h^2*k + 4*a^4*c^4*e*h^2*l - 4*a^4*c^4*d*j^2*k + 3*a^6*b*c*j^2*m
^2 - 4*a^4*c^4*f*h*j^2 + 4*a^3*c^5*e^2*h*k + 4*a^3*c^5*e^2*g*l + 4*a^3*c^5*
e^2*f*m + 2*b^5*c^3*d^2*h*k - 2*b^5*c^3*d^2*g*l + 2*b^5*c^3*d^2*f*m + 2*a^5
*b*c^2*j^3*l + 2*a^2*b^6*e*g*m^2 - 2*a^2*b^6*d*h*m^2 + 4*a^4*c^4*e*g*k^2 +
4*a^4*c^4*d*h*k^2 - 4*a^3*c^5*f^2*g*j - 4*a^3*c^5*e*f^2*l - 4*a^3*c^5*d*f^2
*m - 4*a^4*c^4*d*f*l^2 + 4*a^3*c^5*e*g^2*j + 4*a^3*c^5*d*g^2*k + 2*b^4*c^4*
d^2*g*j - 2*b^4*c^4*d^2*f*k + 2*b^4*c^4*d^2*e*l - 6*a^3*b*c^4*f^3*m + 4*a^3
*c^5*f*g^2*h + 4*a^2*c^6*d^2*g*j + 4*a^2*c^6*d^2*f*k + 4*a^2*c^6*d^2*e*l -
2*a^5*b^2*c*g*l^3 + 2*a^5*b*c^2*h*k^3 + 2*a^4*b*c^3*h^3*k - 4*a^3*c^5*e*g*h
^2 + 4*a^3*c^5*d*f*j^2 - 4*a^2*c^6*d*e^2*k - 2*b^3*c^5*d^2*e*j + 8*a^5*b^2*
c*d*m^3 + 8*a*b^6*c*d^2*m^2 + 8*a*b^2*c^5*d^3*m - 6*a^5*b*c^2*e*l^3 - 6*a^2
*b*c^5*e^3*l - 4*a^2*c^6*e^2*f*h + 2*b^3*c^5*d^2*f*h + 2*a^4*b^3*c*e*l^3 +
```

2*a^4*b*c^3*g*j^3 + 2*a^3*b*c^4*g^3*j + 2*a*b^3*c^4*e^3*l + 4*a^2*c^6*e*f^2
*g + 4*a^2*c^6*d*f^2*h - 6*a^4*b*c^3*d*k^3 - 4*a^2*c^6*d*f*g^2 + 2*b^2*c^6*
d^2*e*g - 2*a*b^2*c^5*e^3*j + 2*a^3*b*c^4*f*h^3 + 2*a^2*b*c^5*f^3*h + 2*a^2
*b*c^5*e*g^3 + 3*a*b*c^6*d^2*g^2 - 9*a^4*b^2*c^2*f^2*m^2 + 4*a^4*b^2*c^2*g^
2*l^2 - 14*a^3*b^3*c^2*e^2*m^2 + 5*a^3*b^3*c^2*f^2*l^2 - 20*a^2*b^4*c^2*d^2
*m^2 + 16*a^3*b^2*c^3*d^2*m^2 - 9*a^3*b^2*c^3*e^2*l^2 + 6*a^2*b^4*c^2*e^2*l
^2 + 4*a^3*b^2*c^3*f^2*k^2 - 14*a^2*b^3*c^3*d^2*l^2 + 5*a^2*b^3*c^3*e^2*k^2
 - 9*a^2*b^2*c^4*d^2*k^2 + 4*a^2*b^2*c^4*e^2*j^2 + 4*a^7*c*k*l^2*m - 4*a^7*
c*j*l*m^2 + 2*b^7*c*d^2*k*m + 2*a^6*b*c*k^3*m + 2*a^6*b*c*j*l^3 + 2*a*b^7*d
*f*m^2 - 6*a^6*b*c*f*m^3 - 6*a*b*c^6*d^3*k - 4*a*c^7*d^2*e*g + 4*a*c^7*d*e^
2*f + 2*a*b*c^6*e^3*g + 2*a*b*c^6*d*f^3 - a^5*b^2*c*j^2*l^2 - a^5*b*c^2*j^2
*k^2 - a^4*b^3*c*h^2*l^2 - a^3*b^4*c*g^2*l^2 - a^4*b*c^3*h^2*j^2 - a^2*b^5*
c*f^2*l^2 - a*b^5*c^2*e^2*k^2 - a^3*b*c^4*g^2*h^2 - a*b^4*c^3*e^2*j^2 - a^2
*b*c^5*f^2*g^2 - a*b^3*c^4*e^2*h^2 - a*b^2*c^5*e^2*g^2 + 2*a^7*b*k*m^3 + 4*
a^7*c*h*m^3 + 4*a*c^7*d^3*h + 2*b*c^7*d^3*f - a^6*b*c*k^2*l^2 - 2*a^6*c^2*j
^2*l^2 - 6*a^6*c^2*h^2*m^2 - a*b^6*c*e^2*l^2 - 6*a^5*c^3*g^2*l^2 - 2*a^5*c^
3*h^2*k^2 - 2*a^5*c^3*f^2*m^2 - 6*a^4*c^4*f^2*k^2 - 6*a^4*c^4*d^2*m^2 - 2*a
^4*c^4*g^2*j^2 - 2*a^4*c^4*e^2*l^2 - 6*a^3*c^5*e^2*j^2 - 2*a^3*c^5*d^2*k^2
 - 2*a^3*c^5*f^2*h^2 - a*b*c^6*e^2*f^2 - 6*a^2*c^6*d^2*h^2 - 2*a^2*c^6*e^2*g
^2 - a^4*b^2*c^2*h^2*k^2 - a^3*b^3*c^2*g^2*k^2 - a^3*b^2*c^3*g^2*j^2 - a^2*
b^4*c^2*f^2*k^2 - a^2*b^3*c^3*f^2*j^2 - a^2*b^2*c^4*f^2*h^2 - 2*a^7*c*k^2*m
^2 + 4*a^5*c^3*h^3*m - 2*a^6*b^2*h*m^3 + 4*a^6*c^2*g*l^3 + 4*a^4*c^4*g^3*l
 - 2*b^4*c^4*d^3*m + 2*a^5*b^3*f*m^3 - 4*a^6*c^2*d*m^3 + 4*a^5*c^3*f*k^3 + 4
*a^3*c^5*f^3*k - 4*a^2*c^6*d^3*m + 2*b^3*c^5*d^3*k - 2*a^4*b^4*d*m^3 + 4*a^
4*c^4*e*j^3 + 4*a^2*c^6*e^3*j - 2*b^2*c^6*d^3*h + 4*a^3*c^5*d*h^3 - 2*a*c^7
*d^2*f^2 - a^6*b^2*k^2*m^2 - a^5*b^3*j^2*m^2 - a^4*b^4*h^2*m^2 - a^3*b^5*g^
2*m^2 - a^2*b^6*f^2*m^2 - b^6*c^2*d^2*k^2 - b^5*c^3*d^2*j^2 - b^4*c^4*d^2*h
^2 - b^3*c^5*d^2*g^2 - b^2*c^6*d^2*f^2 - a^7*b*l^2*m^2 - b^7*c*d^2*l^2 - a*
b^7*e^2*m^2 - b*c^7*d^2*e^2 - b^8*d^2*m^2 - a^6*c^2*k^4 - a^5*c^3*j^4 - a^4
*c^4*h^4 - a^3*c^5*g^4 - a^2*c^6*f^4 - a^7*c*l^4 - a*c^7*e^4 - a^8*m^4 - c^
8*d^4, z, k1)*x*(8*b^3*c^7 - 32*a*b*c^8))/c^5) - (4*b*c^7*d*e + 8*a*c^7*d*g
 - 8*a*c^7*e*f - 4*b^2*c^6*d*g - 8*a^2*c^6*g*h + 4*b^3*c^5*d*j - 8*a^2*c^6*
d*l + 8*a^2*c^6*e*k + 8*a^2*c^6*f*j - 4*b^4*c^4*d*l + 8*a^3*c^5*g*m + 8*a^3
*c^5*h*l - 8*a^3*c^5*j*k - 8*a^4*c^4*l*m + 16*a*b^2*c^5*d*l - 4*a*b^2*c^5*e
*k - 4*a*b^2*c^5*f*j + 4*a*b^3*c^4*e*m + 4*a*b^3*c^4*f*l - 12*a^2*b*c^5*e*m
 - 12*a^2*b*c^5*f*l + 4*a^2*b*c^5*g*k + 4*a^2*b*c^5*h*j + 4*a^3*b*c^4*j*m +
 4*a^3*b*c^4*k*l - 4*a^2*b^2*c^4*g*m - 4*a^2*b^2*c^4*h*l + 4*a*b*c^6*e*h +
4*a*b*c^6*f*g - 12*a*b*c^6*d*j)/c^5 + (x*(4*c^8*d^2 + 2*b^8*m^2 - 4*a*c^7*f
^2 - 2*b*c^7*e^2 + 2*b^7*c*l^2 + 2*b^2*c^6*f^2 + 4*a^2*c^6*h^2 + 2*b^3*c^5*
g^2 + 2*b^4*c^4*h^2 - 4*a^3*c^5*k^2 + 2*b^5*c^3*j^2 + 2*b^6*c^2*k^2 + 4*a^4
*c^4*m^2 - 8*a*b^2*c^5*h^2 - 10*a*b^3*c^4*j^2 + 6*a^2*b*c^5*j^2 - 12*a*b^4*
c^3*k^2 - 14*a*b^5*c^2*l^2 - 18*a^3*b*c^4*l^2 - 4*b*c^7*d*f - 8*a*c^7*d*h +
 8*a*c^7*e*g - 4*b^7*c*k*m + 18*a^2*b^2*c^4*k^2 + 28*a^2*b^3*c^3*l^2 + 40*a
^2*b^4*c^2*m^2 - 32*a^3*b^2*c^3*m^2 - 10*a*b*c^6*g^2 + 4*b^2*c^6*d*h - 16*a
*b^6*c*m^2 - 4*b^3*c^5*f*h - 4*b^3*c^5*d*k + 8*a^2*c^6*d*m - 8*a^2*c^6*e*l

+ 8*a^2*c^6*f*k - 8*a^2*c^6*g*j + 4*b^4*c^4*d*m + 4*b^4*c^4*f*k - 4*b^4*c^4
*g*j - 4*b^5*c^3*f*m + 4*b^5*c^3*g*l - 4*b^5*c^3*h*k - 8*a^3*c^5*h*m + 8*a^
3*c^5*j*l + 4*b^6*c^2*h*m - 4*b^6*c^2*j*l - 16*a*b^2*c^5*d*m + 4*a*b^2*c^5*
e*l - 16*a*b^2*c^5*f*k + 20*a*b^2*c^5*g*j + 20*a*b^3*c^4*f*m - 24*a*b^3*c^4
*g*l + 20*a*b^3*c^4*h*k - 20*a^2*b*c^5*f*m + 28*a^2*b*c^5*g*l - 20*a^2*b*c^
5*h*k - 24*a*b^4*c^3*h*m + 24*a*b^4*c^3*j*l + 28*a*b^5*c^2*k*m + 28*a^3*b*c
^4*k*m + 36*a^2*b^2*c^4*h*m - 32*a^2*b^2*c^4*j*l - 56*a^2*b^3*c^3*k*m + 12*
a*b*c^6*f*h + 12*a*b*c^6*d*k - 4*a*b*c^6*e*j))/c^5) + (x*(c^7*e^3 + c^7*d^2
*g + b^7*e*m^2 - a^3*c^4*j^3 + b^2*c^5*e*g^2 - a^3*b^3*c*l^3 + 2*a^4*b*c^2*
l^3 + b^3*c^4*e*h^2 + 3*a^2*c^5*e*j^2 + a^2*c^5*g*h^2 + 2*b^2*c^5*e^2*j + b
^4*c^3*e*j^2 - a^2*c^5*g^2*j + a^3*c^4*e*l^2 + b^2*c^5*d^2*l + b^5*c^2*e*k^
2 + a^2*c^5*f^2*l - a^3*c^4*g*k^2 - 2*b^3*c^4*e^2*l - a^3*c^4*h^2*l + a^4*c
^3*g*m^2 + a^2*b^5*j*m^2 - a^4*c^3*j*l^2 + a^4*c^3*k^2*l - a^3*b^4*l*m^2 -
a^5*c^2*l*m^2 - 2*c^7*d*e*f + a^2*b^2*c^3*j^3 - a*b*c^5*g^3 + a*c^6*e*g^2 +
b*c^6*e*f^2 - a*c^6*f^2*g - 2*b*c^6*e^2*g - 3*a*c^6*e^2*j - b*c^6*d^2*j -
a*c^6*d^2*l + b^6*c*e*l^2 - a*b^6*g*m^2 - 2*a*b*c^5*e*h^2 + 5*a*b*c^5*e^2*l
- 6*a*b^5*c*e*m^2 - 2*b^2*c^5*e*f*h - a*b^5*c*g*l^2 - 2*b^2*c^5*d*e*k + 2*
b^3*c^4*d*e*m + 2*b^3*c^4*e*f*k - 2*b^3*c^4*e*g*j + 2*a^2*c^5*d*g*m + 2*a^2
*c^5*d*h*l - 2*a^2*c^5*e*f*m - 2*a^2*c^5*e*g*l - 2*a^2*c^5*e*h*k + 2*a^2*c^
5*f*g*k - 2*a^2*c^5*f*h*j - 2*a^2*c^5*d*j*k - 2*b^4*c^3*e*f*m + 2*b^4*c^3*e
*g*l - 2*b^4*c^3*e*h*k + 2*b^5*c^2*e*h*m - 2*a^3*c^4*g*h*m - 2*b^5*c^2*e*j*
l - 2*a^3*c^4*d*l*m + 2*a^3*c^4*e*k*m + 2*a^3*c^4*f*j*m - 2*a^3*c^4*f*k*l +
2*a^3*c^4*g*j*l + 2*a^3*c^4*h*j*k + 2*a^4*c^3*h*l*m - 2*a^4*c^3*j*k*m - 3*
a*b^2*c^4*e*j^2 - a*b^2*c^4*g*h^2 - 4*a*b^3*c^3*e*k^2 + 3*a^2*b*c^4*e*k^2 +
2*a*b^2*c^4*g^2*j - a*b^3*c^3*g*j^2 - 5*a*b^4*c^2*e*l^2 - a*b^4*c^2*g*k^2
+ a^2*b*c^4*h^2*j - 4*a^3*b*c^3*e*m^2 - 2*a*b^3*c^3*g^2*l + 4*a^2*b*c^4*g^2
*l - 5*a^3*b*c^3*g*l^2 + 5*a^2*b^4*c*g*m^2 - 2*a^3*b*c^3*j*k^2 + a^2*b^4*c*
j*l^2 + 3*a^3*b*c^3*j^2*l - 4*a^3*b^3*c*j*m^2 + 3*a^4*b*c^2*j*m^2 + 3*a^4*b
^2*c*l*m^2 + 2*b*c^6*d*e*h - 2*a*c^6*d*g*h + 2*a*c^6*e*f*h + 2*a*c^6*d*e*k
+ 2*a*c^6*d*f*j - 2*b^6*c*e*k*m + 6*a^2*b^2*c^3*e*l^2 + 3*a^2*b^2*c^3*g*k^2
+ 10*a^2*b^3*c^2*e*m^2 + 4*a^2*b^3*c^2*g*l^2 - 6*a^3*b^2*c^2*g*m^2 + a^2*b
^3*c^2*j*k^2 - 2*a^2*b^3*c^2*j^2*l - a^3*b^2*c^2*j*l^2 - a^3*b^2*c^2*k^2*l
+ 2*a*b*c^5*f*g*h - 4*a*b*c^5*d*e*m - 2*a*b*c^5*d*f*l + 2*a*b*c^5*d*g*k - 4
*a*b*c^5*e*f*k + 2*a*b*c^5*e*g*j + 2*a*b^5*c*g*k*m - 2*a*b^2*c^4*d*g*m + 6*
a*b^2*c^4*e*f*m - 4*a*b^2*c^4*e*g*l + 6*a*b^2*c^4*e*h*k - 2*a*b^2*c^4*f*g*k
- 8*a*b^3*c^3*e*h*m + 2*a*b^3*c^3*f*g*m + 2*a*b^3*c^3*g*h*k + 6*a^2*b*c^4*
e*h*m - 4*a^2*b*c^4*f*g*m - 4*a^2*b*c^4*g*h*k + 8*a*b^3*c^3*e*j*l + 2*a^2*b
*c^4*d*j*m - 8*a^2*b*c^4*e*j*l + 2*a^2*b*c^4*f*j*k - 2*a*b^4*c^2*g*h*m + 10
*a*b^4*c^2*e*k*m + 2*a*b^4*c^2*g*j*l + 2*a^3*b*c^3*f*l*m + 6*a^3*b*c^3*g*k*
m - 4*a^3*b*c^3*h*j*m + 2*a^3*b*c^3*h*k*l - 2*a^2*b^4*c*j*k*m + 2*a^3*b^3*c
*k*l*m - 4*a^4*b*c^2*k*l*m + 6*a^2*b^2*c^3*g*h*m - 12*a^2*b^2*c^3*e*k*m - 2
*a^2*b^2*c^3*f*j*m - 4*a^2*b^2*c^3*g*j*l - 2*a^2*b^2*c^3*h*j*k - 8*a^2*b^3*
c^2*g*k*m + 2*a^2*b^3*c^2*h*j*m - 2*a^3*b^2*c^2*h*l*m + 6*a^3*b^2*c^2*j*k*m
))/c^5)*root(128*a^2*b^2*c^8*z^4 - 16*a*b^4*c^7*z^4 - 256*a^3*c^9*z^4 + 384
*a^3*b^2*c^6*l*z^3 - 144*a^2*b^4*c^5*l*z^3 + 128*a^2*b^3*c^6*j*z^3 - 128*a^

2*b^2*c^7*g*z^3 + 16*a*b^6*c^4*l*z^3 - 256*a^3*b*c^7*j*z^3 - 16*a*b^5*c^5*j
*z^3 + 16*a*b^4*c^6*g*z^3 - 256*a^4*c^7*l*z^3 + 256*a^3*c^8*g*z^3 - 96*a^4*
b*c^5*j*l*z^2 + 8*a*b^7*c^2*j*l*z^2 + 160*a^4*b*c^5*h*m*z^2 - 8*a*b^7*c^2*h
*m*z^2 + 8*a*b^6*c^3*h*k*z^2 - 8*a*b^6*c^3*g*l*z^2 + 8*a*b^6*c^3*f*m*z^2 +
160*a^3*b*c^6*g*j*z^2 - 96*a^3*b*c^6*f*k*z^2 - 96*a^3*b*c^6*e*l*z^2 - 96*a^
3*b*c^6*d*m*z^2 + 8*a*b^5*c^4*g*j*z^2 - 8*a*b^5*c^4*f*k*z^2 - 8*a*b^5*c^4*e
*l*z^2 - 8*a*b^5*c^4*d*m*z^2 + 8*a*b^4*c^5*e*j*z^2 + 8*a*b^4*c^5*d*k*z^2 +
8*a*b^4*c^5*f*h*z^2 + 32*a^2*b*c^7*e*g*z^2 + 32*a^2*b*c^7*d*h*z^2 - 8*a*b^3
*c^6*e*g*z^2 - 8*a*b^3*c^6*d*h*z^2 + 16*a*b^2*c^7*d*f*z^2 + 8*a*b^8*c*k*m*z
^2 - 304*a^4*b^2*c^4*k*m*z^2 + 264*a^3*b^4*c^3*k*m*z^2 - 80*a^2*b^6*c^2*k*m
*z^2 + 184*a^3*b^3*c^4*j*l*z^2 - 72*a^2*b^5*c^3*j*l*z^2 - 200*a^3*b^3*c^4*h
*m*z^2 + 72*a^2*b^5*c^3*h*m*z^2 - 240*a^3*b^2*c^5*g*l*z^2 + 144*a^3*b^2*c^5
*h*k*z^2 + 144*a^3*b^2*c^5*f*m*z^2 + 80*a^2*b^4*c^4*g*l*z^2 - 64*a^2*b^4*c^
4*h*k*z^2 - 64*a^2*b^4*c^4*f*m*z^2 - 72*a^2*b^3*c^5*g*j*z^2 + 56*a^2*b^3*c^
5*f*k*z^2 + 56*a^2*b^3*c^5*e*l*z^2 + 56*a^2*b^3*c^5*d*m*z^2 - 48*a^2*b^2*c^
6*e*j*z^2 - 48*a^2*b^2*c^6*d*k*z^2 - 48*a^2*b^2*c^6*f*h*z^2 - 112*a^5*b*c^4
*m^2*z^2 + 44*a^2*b^7*c*m^2*z^2 + 80*a^4*b*c^5*k^2*z^2 - 4*a*b^7*c^2*k^2*z^
2 - 4*a*b^6*c^3*j^2*z^2 - 48*a^3*b*c^6*h^2*z^2 - 4*a*b^5*c^4*h^2*z^2 - 4*a*
b^4*c^5*g^2*z^2 + 16*a^2*b*c^7*f^2*z^2 - 4*a*b^3*c^6*f^2*z^2 + 8*a*b^2*c^7*
e^2*z^2 + 64*a^5*c^5*k*m*z^2 + 192*a^4*c^6*g*l*z^2 - 64*a^4*c^6*h*k*z^2 - 6
4*a^4*c^6*f*m*z^2 + 64*a^3*c^7*e*j*z^2 + 64*a^3*c^7*d*k*z^2 + 64*a^3*c^7*f*
h*z^2 - 4*a*b^8*c*l^2*z^2 - 64*a^2*c^8*d*f*z^2 + 16*a*b*c^8*d^2*z^2 + 252*a
^4*b^3*c^3*m^2*z^2 - 168*a^3*b^5*c^2*m^2*z^2 + 168*a^4*b^2*c^4*l^2*z^2 - 13
2*a^3*b^4*c^3*l^2*z^2 + 40*a^2*b^6*c^2*l^2*z^2 - 100*a^3*b^3*c^4*k^2*z^2 +
36*a^2*b^5*c^3*k^2*z^2 - 56*a^3*b^2*c^5*j^2*z^2 + 32*a^2*b^4*c^4*j^2*z^2 +
28*a^2*b^3*c^5*h^2*z^2 + 40*a^2*b^2*c^6*g^2*z^2 - 96*a^5*c^5*l^2*z^2 - 32*a
^4*c^6*j^2*z^2 - 96*a^3*c^7*g^2*z^2 - 32*a^2*c^8*e^2*z^2 - 4*b^3*c^7*d^2*z^
2 - 4*a*b^9*m^2*z^2 + 32*a^5*b*c^3*h*l*m*z + 8*a^2*b^6*c*g*k*m*z + 96*a^4*b
*c^4*e*k*m*z + 32*a^4*b*c^4*h*j*k*z + 32*a^4*b*c^4*g*j*l*z + 32*a^4*b*c^4*f
*j*m*z - 64*a^4*b*c^4*g*h*m*z - 8*a*b^6*c^2*e*j*l*z + 8*a*b^6*c^2*e*h*m*z -
64*a^3*b*c^5*e*h*k*z + 64*a^3*b*c^5*e*g*l*z - 64*a^3*b*c^5*e*f*m*z + 32*a^
3*b*c^5*f*g*k*z - 32*a^3*b*c^5*d*h*l*z + 32*a^3*b*c^5*d*g*m*z - 8*a*b^5*c^3
*e*h*k*z + 8*a*b^5*c^3*e*g*l*z - 8*a*b^5*c^3*e*f*m*z - 8*a*b^4*c^4*e*g*j*z
+ 8*a*b^4*c^4*e*f*k*z - 8*a*b^4*c^4*d*f*l*z + 8*a*b^4*c^4*d*e*m*z - 32*a^2*
b*c^6*d*f*j*z + 32*a^2*b*c^6*d*e*k*z + 8*a*b^3*c^5*d*f*j*z - 8*a*b^3*c^5*d*
e*k*z + 32*a^2*b*c^6*e*f*h*z - 8*a*b^3*c^5*e*f*h*z - 8*a*b^2*c^6*d*f*g*z +
8*a*b^2*c^6*d*e*h*z - 8*a*b^7*c*e*k*m*z - 40*a^5*b^2*c^2*k*l*m*z + 48*a^4*b
^3*c^2*j*k*m*z - 8*a^4*b^3*c^2*h*l*m*z + 104*a^4*b^2*c^3*g*k*m*z - 56*a^3*b
^4*c^2*g*k*m*z - 40*a^4*b^2*c^3*h*j*m*z + 8*a^4*b^2*c^3*h*k*l*z + 8*a^4*b^2
*c^3*f*l*m*z + 8*a^3*b^4*c^2*h*j*m*z - 152*a^3*b^3*c^3*e*k*m*z + 64*a^2*b^5
*c^2*e*k*m*z - 40*a^3*b^3*c^3*g*j*l*z - 8*a^3*b^3*c^3*h*j*k*z - 8*a^3*b^3*c
^3*f*j*m*z + 8*a^2*b^5*c^2*g*j*l*z + 48*a^3*b^3*c^3*g*h*m*z - 8*a^2*b^5*c^2
*g*h*m*z - 104*a^3*b^2*c^4*e*j*l*z + 56*a^2*b^4*c^3*e*j*l*z + 8*a^3*b^2*c^4
*f*j*k*z - 8*a^3*b^2*c^4*d*k*l*z + 8*a^3*b^2*c^4*d*j*m*z + 104*a^3*b^2*c^4*
e*h*m*z - 56*a^2*b^4*c^3*e*h*m*z - 40*a^3*b^2*c^4*g*h*k*z - 40*a^3*b^2*c^4*

f*g*m*z - 8*a^3*b^2*c^4*f*h*l*z + 8*a^2*b^4*c^3*g*h*k*z + 8*a^2*b^4*c^3*f*g
*m*z + 48*a^2*b^3*c^4*e*h*k*z - 48*a^2*b^3*c^4*e*g*l*z + 48*a^2*b^3*c^4*e*f
*m*z - 8*a^2*b^3*c^4*f*g*k*z + 8*a^2*b^3*c^4*d*h*l*z - 8*a^2*b^3*c^4*d*g*m*
z + 40*a^2*b^2*c^5*e*g*j*z - 40*a^2*b^2*c^5*e*f*k*z + 40*a^2*b^2*c^5*d*f*l*
z - 40*a^2*b^2*c^5*d*e*m*z - 8*a^2*b^2*c^5*d*h*j*z + 8*a^2*b^2*c^5*d*g*k*z
+ 8*a^2*b^2*c^5*f*g*h*z + 8*a^4*b^4*c*k*l*m*z - 64*a^5*b*c^3*j*k*m*z - 8*a^
3*b^5*c*j*k*m*z - 32*a^6*b*c^2*l*m^2*z + 24*a^5*b^3*c*l*m^2*z - 28*a^4*b^4*
c*j*m^2*z + 16*a^5*b*c^3*k^2*l*z + 4*a^3*b^5*c*j*l^2*z + 48*a^5*b*c^3*g*m^2
*z + 32*a^3*b^5*c*g*m^2*z - 4*a^2*b^6*c*g*l^2*z - 36*a^2*b^6*c*e*m^2*z - 32
*a^4*b*c^4*g*k^2*z - 16*a^3*b*c^5*f^2*l*z - 48*a^4*b*c^4*e*l^2*z - 32*a^3*b
*c^5*g^2*j*z - 4*a*b^4*c^4*e^2*l*z + 32*a^2*b*c^6*d^2*l*z - 24*a*b^3*c^5*d^
2*l*z + 4*a*b^6*c^2*e*k^2*z + 32*a^3*b*c^5*e*j^2*z + 16*a^3*b*c^5*g*h^2*z -
16*a^2*b*c^6*e^2*j*z + 4*a*b^5*c^3*e*j^2*z + 4*a*b^3*c^5*e^2*j*z + 20*a*b^
2*c^6*d^2*j*z + 4*a*b^4*c^4*e*h^2*z - 16*a^2*b*c^6*e*g^2*z + 4*a*b^3*c^5*e*
g^2*z - 4*a*b^2*c^6*e^2*g*z + 4*a*b^2*c^6*e*f^2*z + 32*a^6*c^3*k*l*m*z - 32
*a^5*c^4*h*k*l*z + 32*a^5*c^4*h*j*m*z - 32*a^5*c^4*g*k*m*z - 32*a^5*c^4*f*l
*m*z - 32*a^4*c^5*f*j*k*z + 32*a^4*c^5*e*j*l*z + 32*a^4*c^5*d*k*l*z - 32*a^
4*c^5*d*j*m*z + 32*a^4*c^5*g*h*k*z + 32*a^4*c^5*f*h*l*z + 32*a^4*c^5*f*g*m*
z - 32*a^4*c^5*e*h*m*z - 32*a^3*c^6*e*g*j*z + 32*a^3*c^6*e*f*k*z + 32*a^3*c
^6*d*h*j*z - 32*a^3*c^6*d*g*k*z - 32*a^3*c^6*d*f*l*z + 32*a^3*c^6*d*e*m*z -
32*a^3*c^6*f*g*h*z + 4*a*b^7*c*e*l^2*z + 32*a^2*c^7*d*f*g*z - 32*a^2*c^7*d
*e*h*z - 16*a*b*c^7*d^2*g*z + 52*a^5*b^2*c^2*j*m^2*z - 4*a^4*b^3*c^2*k^2*l*
z + 36*a^4*b^2*c^3*j^2*l*z - 16*a^4*b^3*c^2*j*l^2*z - 8*a^3*b^4*c^2*j^2*l*z
- 20*a^4*b^2*c^3*j*k^2*z + 4*a^3*b^4*c^2*j*k^2*z - 76*a^4*b^3*c^2*g*m^2*z
- 60*a^4*b^2*c^3*g*l^2*z + 44*a^3*b^2*c^4*g^2*l*z + 28*a^3*b^4*c^2*g*l^2*z
- 8*a^2*b^4*c^3*g^2*l*z + 104*a^3*b^4*c^2*e*m^2*z - 100*a^4*b^2*c^3*e*m^2*z
+ 24*a^3*b^3*c^3*g*k^2*z + 4*a^3*b^2*c^4*h^2*j*z - 4*a^2*b^5*c^2*g*k^2*z +
4*a^2*b^3*c^4*f^2*l*z + 76*a^3*b^3*c^3*e*l^2*z - 32*a^2*b^5*c^2*e*l^2*z +
20*a^2*b^2*c^5*e^2*l*z + 12*a^3*b^2*c^4*g*j^2*z + 8*a^2*b^3*c^4*g^2*j*z - 4
*a^2*b^4*c^3*g*j^2*z + 52*a^3*b^2*c^4*e*k^2*z - 28*a^2*b^4*c^3*e*k^2*z - 4*
a^2*b^2*c^5*f^2*j*z - 24*a^2*b^3*c^4*e*j^2*z - 4*a^2*b^3*c^4*g*h^2*z - 20*a
^2*b^2*c^5*e*h^2*z + 20*a^5*b^2*c^2*l^3*z + 4*a^3*b^3*c^3*j^3*z - 4*a^2*b^2
*c^5*g^3*z - 4*a^4*b^5*l*m^2*z - 16*a^6*c^3*j*m^2*z - 16*a^5*c^4*j^2*l*z +
4*a^3*b^6*j*m^2*z + 16*a^5*c^4*j*k^2*z + 48*a^5*c^4*g*l^2*z - 48*a^4*c^5*g^
2*l*z - 4*a^2*b^7*g*m^2*z + 16*a^5*c^4*e*m^2*z - 16*a^4*c^5*h^2*j*z + 16*a^
4*c^5*g*j^2*z - 16*a^3*c^6*e^2*l*z + 4*b^5*c^4*d^2*l*z - 16*a^4*c^5*e*k^2*z
+ 16*a^3*c^6*f^2*j*z - 4*b^4*c^5*d^2*j*z - 16*a^2*c^7*d^2*j*z - 4*a^4*b^4*
c*l^3*z + 16*a^3*c^6*e*h^2*z - 16*a^4*b*c^4*j^3*z + 16*a^2*c^7*e^2*g*z + 4*
b^3*c^6*d^2*g*z - 16*a^2*c^7*e*f^2*z - 4*b^2*c^7*d^2*e*z + 4*a*b^8*e*m^2*z
+ 16*a*c^8*d^2*e*z - 16*a^6*c^3*l^3*z + 16*a^3*c^6*g^3*z + 4*a^5*b^2*c*g*k*
l*m + 12*a^5*b*c^2*g*j*k*m + 12*a^5*b*c^2*e*k*l*m - 4*a^5*b*c^2*h*j*k*l - 4
*a^5*b*c^2*f*j*l*m - 4*a^4*b^3*c*g*j*k*m - 4*a^4*b^3*c*e*k*l*m - 4*a^5*b*c^
2*g*h*l*m + 4*a^3*b^4*c*e*j*k*m - 4*a^3*b^4*c*f*h*k*m + 12*a^4*b*c^3*d*j*k*
l - 20*a^4*b*c^3*e*g*k*m + 12*a^4*b*c^3*f*h*j*l + 12*a^4*b*c^3*e*h*j*m + 12
*a^4*b*c^3*d*h*k*m - 4*a^4*b*c^3*g*h*j*k - 4*a^4*b*c^3*f*g*k*l - 4*a^4*b*c^

3*f*g*j*m - 4*a^4*b*c^3*e*h*k*l - 4*a^4*b*c^3*e*f*l*m - 4*a^4*b*c^3*d*g*l*m
 - 4*a^2*b^5*c*e*g*k*m + 4*a^2*b^5*c*d*h*k*m - 20*a^3*b*c^4*d*f*j*l - 4*a^3
*b*c^4*e*f*j*k - 4*a^3*b*c^4*d*g*j*k - 4*a^3*b*c^4*d*e*k*l - 4*a^3*b*c^4*d*
e*j*m - 4*a*b^5*c^2*d*f*j*l + 12*a^3*b*c^4*e*g*h*k + 12*a^3*b*c^4*e*f*g*m +
 12*a^3*b*c^4*d*g*h*l + 12*a^3*b*c^4*d*f*h*m - 4*a^3*b*c^4*f*g*h*j - 4*a^3*
b*c^4*e*f*h*l + 4*a*b^5*c^2*d*f*h*m - 4*a*b^4*c^3*d*f*h*k + 4*a*b^4*c^3*d*f
*g*l + 12*a^2*b*c^5*d*f*g*j + 12*a^2*b*c^5*d*e*f*l - 4*a^2*b*c^5*d*e*h*j -
4*a^2*b*c^5*d*e*g*k - 4*a*b^3*c^4*d*f*g*j - 4*a*b^3*c^4*d*e*f*l - 4*a^2*b*c
^5*e*f*g*h + 4*a*b^2*c^5*d*e*f*j - 4*a^6*b*c*j*k*l*m - 4*a*b^6*c*d*f*k*m -
4*a*b*c^6*d*e*f*g - 16*a^4*b^2*c^2*e*j*k*m + 4*a^4*b^2*c^2*f*j*k*l + 4*a^4*
b^2*c^2*d*j*l*m + 12*a^4*b^2*c^2*f*h*k*m + 4*a^4*b^2*c^2*g*h*j*m + 4*a^4*b^
2*c^2*e*h*l*m - 4*a^3*b^3*c^2*d*j*k*l + 20*a^3*b^3*c^2*e*g*k*m - 16*a^3*b^3
*c^2*d*h*k*m - 4*a^3*b^3*c^2*f*h*j*l - 4*a^3*b^3*c^2*e*h*j*m - 40*a^3*b^2*c
^3*d*f*k*m + 24*a^2*b^4*c^2*d*f*k*m - 16*a^3*b^2*c^3*d*h*j*l + 12*a^3*b^2*c
^3*e*g*j*l + 4*a^3*b^2*c^3*e*h*j*k + 4*a^3*b^2*c^3*e*f*j*m + 4*a^3*b^2*c^3*
d*g*k*l - 4*a^2*b^4*c^2*e*g*j*l + 4*a^2*b^4*c^2*d*h*j*l - 16*a^3*b^2*c^3*e*
g*h*m + 4*a^3*b^2*c^3*f*g*h*l + 4*a^2*b^4*c^2*e*g*h*m + 20*a^2*b^3*c^3*d*f*
j*l - 16*a^2*b^3*c^3*d*f*h*m - 4*a^2*b^3*c^3*e*g*h*k - 4*a^2*b^3*c^3*e*f*g*
m - 4*a^2*b^3*c^3*d*g*h*l - 16*a^2*b^2*c^4*d*f*g*l + 12*a^2*b^2*c^4*d*f*h*k
 + 4*a^2*b^2*c^4*e*f*g*k + 4*a^2*b^2*c^4*d*g*h*j + 4*a^2*b^2*c^4*d*e*h*l +
4*a^2*b^2*c^4*d*e*g*m + 2*a^5*b^2*c*j^2*k*m - 4*a^5*b^2*c*h*k^2*m - 2*a^5*b
*c^2*h^2*k*m + 2*a^4*b^3*c*h^2*k*m + 2*a^5*b^2*c*h*k*l^2 + 2*a^5*b^2*c*f*l^
2*m - 2*a^5*b*c^2*h*j^2*m + 2*a^3*b^4*c*g^2*k*m + 14*a^4*b*c^3*f^2*k*m - 10
*a^5*b*c^2*f*k^2*m - 8*a^5*b^2*c*g*j*m^2 - 8*a^5*b^2*c*e*l*m^2 + 4*a^5*b^2*
c*f*k*m^2 + 4*a^4*b^3*c*f*k^2*m - 2*a^5*b*c^2*g*k^2*l + 2*a^2*b^5*c*f^2*k*m
 + 6*a^5*b*c^2*f*k*l^2 + 6*a^5*b*c^2*d*l^2*m - 2*a^5*b*c^2*g*j*l^2 + 2*a^4*
b^3*c*g*j*l^2 - 2*a^4*b^3*c*f*k*l^2 - 2*a^4*b^3*c*d*l^2*m - 2*a^4*b*c^3*g^2
*j*l - 14*a*b^5*c^2*d^2*k*m - 10*a^5*b*c^2*e*j*m^2 + 10*a^4*b^3*c*e*j*m^2 -
 10*a^3*b*c^4*d^2*k*m - 6*a^4*b^3*c*d*k*m^2 + 6*a^4*b*c^3*g^2*h*m - 4*a^3*b
^4*c*d*k^2*m - 2*a^5*b*c^2*d*k*m^2 + 14*a^5*b*c^2*f*h*m^2 + 14*a^3*b*c^4*e^
2*j*l - 10*a^4*b^3*c*f*h*m^2 - 10*a^4*b*c^3*f*h^2*m - 10*a^4*b*c^3*e*j^2*l
- 2*a^4*b*c^3*g*h^2*l - 2*a^4*b*c^3*f*j^2*k - 2*a^4*b*c^3*d*j^2*m - 2*a^3*b
^4*c*e*j*l^2 + 2*a^3*b^4*c*d*k*l^2 + 2*a*b^5*c^2*e^2*j*l - 12*a*b^4*c^3*d^2
*j*l - 10*a^3*b*c^4*e^2*h*m + 6*a^4*b*c^3*e*j*k^2 + 2*a^3*b^4*c*f*h*l^2 - 2
*a*b^5*c^2*e^2*h*m - 12*a^3*b^4*c*e*g*m^2 + 12*a^3*b^4*c*d*h*m^2 + 12*a*b^4
*c^3*d^2*h*m + 6*a^3*b*c^4*f^2*g*l - 2*a^4*b*c^3*f*h*k^2 - 2*a^3*b*c^4*f^2*
h*k + 14*a^4*b*c^3*e*g*l^2 - 10*a^4*b*c^3*d*h*l^2 - 10*a^3*b*c^4*e*g^2*l -
2*a^3*b*c^4*f*g^2*k - 2*a^3*b*c^4*d*g^2*m + 2*a^2*b^5*c*e*g*l^2 - 2*a^2*b^5
*c*d*h*l^2 + 2*a*b^4*c^3*e^2*h*k - 2*a*b^4*c^3*e^2*g*l + 2*a*b^4*c^3*e^2*f*
m - 14*a^2*b^5*c*d*f*m^2 + 14*a^2*b*c^5*d^2*h*k - 10*a^4*b*c^3*d*f*m^2 - 10
*a^3*b*c^4*d*h^2*k - 10*a^2*b*c^5*d^2*g*l - 10*a*b^3*c^4*d^2*h*k + 10*a*b^3
*c^4*d^2*g*l - 6*a*b^3*c^4*d^2*f*m - 4*a*b^4*c^3*d*f^2*m - 2*a^3*b*c^4*e*h^
2*j - 2*a^2*b*c^5*d^2*f*m + 6*a^3*b*c^4*d*h*j^2 + 6*a^2*b*c^5*e^2*f*k + 6*a
^2*b*c^5*d*e^2*m - 2*a^3*b*c^4*e*g*j^2 - 2*a^2*b*c^5*e^2*g*j + 2*a*b^3*c^4*
e^2*g*j - 2*a*b^3*c^4*e^2*f*k - 2*a*b^3*c^4*d*e^2*m + 14*a^3*b*c^4*d*f*k^2

- 10*a^2*b*c^5*d*f^2*k - 8*a*b^2*c^5*d^2*g*j - 8*a*b^2*c^5*d^2*e*l + 4*a*b^
3*c^4*d*f^2*k + 4*a*b^2*c^5*d^2*f*k - 2*a^2*b*c^5*e*f^2*j + 2*a*b^5*c^2*d*f
*k^2 + 2*a*b^4*c^3*d*f*j^2 + 2*a*b^2*c^5*d*e^2*k - 2*a^2*b*c^5*d*g^2*h + 2*
a*b^2*c^5*e^2*f*h - 4*a*b^2*c^5*d*f^2*h - 2*a^2*b*c^5*d*f*h^2 + 2*a*b^3*c^4
*d*f*h^2 + 2*a*b^2*c^5*d*f*g^2 + 8*a^6*c^2*h*j*l*m - 8*a^6*c^2*g*k*l*m - 8*
a^5*c^3*f*j*k*l + 8*a^5*c^3*e*j*k*m - 8*a^5*c^3*d*j*l*m + 8*a^5*c^3*g*h*k*l
 - 8*a^5*c^3*g*h*j*m - 8*a^5*c^3*f*h*k*m + 8*a^5*c^3*f*g*l*m - 8*a^5*c^3*e*
h*l*m - 2*a^6*b*c*h*l^2*m + 8*a^4*c^4*f*g*j*k - 8*a^4*c^4*e*h*j*k - 8*a^4*c
^4*e*g*j*l + 8*a^4*c^4*e*f*k*l - 8*a^4*c^4*e*f*j*m + 8*a^4*c^4*d*h*j*l - 8*
a^4*c^4*d*g*k*l + 8*a^4*c^4*d*g*j*m + 8*a^4*c^4*d*f*k*m + 8*a^4*c^4*d*e*l*m
 + 6*a^6*b*c*g*l*m^2 - 2*a^6*b*c*h*k*m^2 - 8*a^4*c^4*f*g*h*l + 8*a^4*c^4*e*
g*h*m + 2*a*b^6*c*e^2*k*m + 8*a^3*c^5*d*e*j*k + 8*a^3*c^5*e*f*h*j - 8*a^3*c
^5*e*f*g*k - 8*a^3*c^5*d*g*h*j - 8*a^3*c^5*d*f*h*k + 8*a^3*c^5*d*f*g*l - 8*
a^3*c^5*d*e*h*l - 8*a^3*c^5*d*e*g*m - 8*a^2*c^6*d*e*f*j + 8*a^2*c^6*d*e*g*h
 + 2*a*b^6*c*d*f*l^2 + 6*a*b*c^6*d^2*e*j - 2*a*b*c^6*d^2*f*h - 2*a*b*c^6*d*
e^2*h - 8*a^4*b^2*c^2*g^2*k*m - 10*a^3*b^3*c^2*f^2*k*m + 2*a^4*b^2*c^2*h^2*
j*l + 18*a^3*b^2*c^3*e^2*k*m - 12*a^2*b^4*c^2*e^2*k*m - 4*a^4*b^2*c^2*g*j^2
*l + 2*a^3*b^3*c^2*g^2*j*l + 28*a^2*b^3*c^3*d^2*k*m + 14*a^4*b^2*c^2*d*k^2*
m - 8*a^3*b^2*c^3*f^2*j*l + 2*a^4*b^2*c^2*g*j*k^2 + 2*a^4*b^2*c^2*e*k^2*l -
 2*a^3*b^3*c^2*g^2*h*m + 2*a^2*b^4*c^2*f^2*j*l - 10*a^2*b^3*c^3*e^2*j*l - 8
*a^4*b^2*c^2*d*k*l^2 + 4*a^4*b^2*c^2*e*j*l^2 + 4*a^3*b^3*c^2*f*h^2*m + 4*a^
3*b^3*c^2*e*j^2*l + 4*a^3*b^2*c^3*f^2*h*m - 2*a^2*b^4*c^2*f^2*h*m + 18*a^2*
b^2*c^4*d^2*j*l + 10*a^2*b^3*c^3*e^2*h*m - 8*a^4*b^2*c^2*f*h*l^2 - 2*a^3*b^
3*c^2*e*j*k^2 + 2*a^3*b^2*c^3*g^2*h*k + 2*a^3*b^2*c^3*f*g^2*m - 22*a^4*b^2*
c^2*d*h*m^2 - 22*a^2*b^2*c^4*d^2*h*m + 18*a^4*b^2*c^2*e*g*m^2 + 16*a^3*b^2*
c^3*d*h^2*m - 4*a^3*b^2*c^3*f*h^2*k - 4*a^2*b^4*c^2*d*h^2*m + 2*a^3*b^3*c^2
*f*h*k^2 + 2*a^3*b^2*c^3*d*j^2*k + 2*a^2*b^3*c^3*f^2*h*k - 2*a^2*b^3*c^3*f^
2*g*l - 10*a^3*b^3*c^2*e*g*l^2 + 10*a^3*b^3*c^2*d*h*l^2 - 8*a^2*b^2*c^4*e^2
*h*k - 8*a^2*b^2*c^4*e^2*f*m + 4*a^2*b^3*c^3*e*g^2*l + 4*a^2*b^2*c^4*e^2*g*
l + 2*a^3*b^2*c^3*f*h*j^2 + 28*a^3*b^3*c^2*d*f*m^2 + 14*a^2*b^2*c^4*d*f^2*m
 - 8*a^3*b^2*c^3*e*g*k^2 + 4*a^3*b^2*c^3*d*h*k^2 + 4*a^2*b^3*c^3*d*h^2*k +
2*a^2*b^4*c^2*e*g*k^2 - 2*a^2*b^4*c^2*d*h*k^2 + 2*a^2*b^2*c^4*f^2*g*j + 2*a
^2*b^2*c^4*e*f^2*l + 18*a^3*b^2*c^3*d*f*l^2 - 12*a^2*b^4*c^2*d*f*l^2 - 4*a^
2*b^2*c^4*e*g^2*j + 2*a^2*b^3*c^3*e*g*j^2 - 2*a^2*b^3*c^3*d*h*j^2 - 10*a^2*
b^3*c^3*d*f*k^2 - 8*a^2*b^2*c^4*d*f*j^2 + 2*a^2*b^2*c^4*e*g*h^2 + 4*a^5*b^2
*c*h^2*m^2 - 2*a^4*b^2*c^2*h^3*m - 5*a^5*b*c^2*g^2*m^2 + 5*a^4*b^3*c*g^2*m^
2 + 3*a^5*b*c^2*h^2*l^2 + 6*a^3*b^4*c*f^2*m^2 - 2*a^3*b^2*c^3*g^3*l + 2*a^2
*b^3*c^3*f^3*m + 7*a^4*b*c^3*e^2*m^2 + 7*a^2*b^5*c*e^2*m^2 - 5*a^4*b*c^3*f^
2*l^2 + 3*a^4*b*c^3*g^2*k^2 - 2*a^4*b^2*c^2*f*k^3 - 2*a^2*b^2*c^4*f^3*k + 7
*a^3*b*c^4*d^2*l^2 + 7*a*b^5*c^2*d^2*l^2 - 5*a^3*b*c^4*e^2*k^2 + 3*a^3*b*c^
4*f^2*j^2 + 6*a*b^4*c^3*d^2*k^2 + 2*a^3*b^3*c^2*d*k^3 - 2*a^3*b^2*c^3*e*j^3
 - 5*a^2*b*c^5*d^2*j^2 + 5*a*b^3*c^4*d^2*j^2 + 3*a^2*b*c^5*e^2*h^2 + 4*a*b^
2*c^5*d^2*h^2 - 2*a^2*b^2*c^4*d*h^3 - 4*a^6*c^2*j^2*k*m + 2*a^6*b^2*j*l*m^2
 + 4*a^6*c^2*j*k^2*l + 4*a^6*c^2*h*k^2*m - 4*a^6*c^2*h*k*l^2 - 4*a^6*c^2*f*
l^2*m + 4*a^5*c^3*g^2*k*m + 2*a^5*b^3*h*k*m^2 - 2*a^5*b^3*g*l*m^2 + 4*a^6*c

^2*g*j*m^2 + 4*a^6*c^2*f*k*m^2 + 4*a^6*c^2*e*l*m^2 - 4*a^5*c^3*h^2*j*l + 4*
a^5*c^3*h*j^2*k + 4*a^5*c^3*g*j^2*l + 4*a^5*c^3*f*j^2*m - 4*a^4*c^4*e^2*k*m
 + 2*a^4*b^4*g*j*m^2 - 2*a^4*b^4*f*k*m^2 + 2*a^4*b^4*e*l*m^2 - 4*a^5*c^3*g*
j*k^2 - 4*a^5*c^3*e*k^2*l - 4*a^5*c^3*d*k^2*m + 4*a^4*c^4*f^2*j*l + 4*a^5*c
^3*e*j*l^2 + 4*a^5*c^3*d*k*l^2 + 4*a^4*c^4*f^2*h*m + 2*b^6*c^2*d^2*j*l - 2*
a^3*b^5*e*j*m^2 + 2*a^3*b^5*d*k*m^2 + 4*a^5*c^3*f*h*l^2 - 4*a^4*c^4*g^2*h*k
 - 4*a^4*c^4*f*g^2*m - 4*a^3*c^5*d^2*j*l - 2*b^6*c^2*d^2*h*m + 2*a^3*b^5*f*
h*m^2 + 12*a^5*c^3*d*h*m^2 - 12*a^4*c^4*d*h^2*m + 12*a^3*c^5*d^2*h*m - 4*a^
5*c^3*e*g*m^2 + 4*a^4*c^4*g*h^2*j + 4*a^4*c^4*f*h^2*k + 4*a^4*c^4*e*h^2*l -
 4*a^4*c^4*d*j^2*k + 3*a^6*b*c*j^2*m^2 - 4*a^4*c^4*f*h*j^2 + 4*a^3*c^5*e^2*
h*k + 4*a^3*c^5*e^2*g*l + 4*a^3*c^5*e^2*f*m + 2*b^5*c^3*d^2*h*k - 2*b^5*c^3
*d^2*g*l + 2*b^5*c^3*d^2*f*m + 2*a^5*b*c^2*j^3*l + 2*a^2*b^6*e*g*m^2 - 2*a^
2*b^6*d*h*m^2 + 4*a^4*c^4*e*g*k^2 + 4*a^4*c^4*d*h*k^2 - 4*a^3*c^5*f^2*g*j -
 4*a^3*c^5*e*f^2*l - 4*a^3*c^5*d*f^2*m - 4*a^4*c^4*d*f*l^2 + 4*a^3*c^5*e*g^
2*j + 4*a^3*c^5*d*g^2*k + 2*b^4*c^4*d^2*g*j - 2*b^4*c^4*d^2*f*k + 2*b^4*c^4
*d^2*e*l - 6*a^3*b*c^4*f^3*m + 4*a^3*c^5*f*g^2*h + 4*a^2*c^6*d^2*g*j + 4*a^
2*c^6*d^2*f*k + 4*a^2*c^6*d^2*e*l - 2*a^5*b^2*c*g*l^3 + 2*a^5*b*c^2*h*k^3 +
 2*a^4*b*c^3*h^3*k - 4*a^3*c^5*e*g*h^2 + 4*a^3*c^5*d*f*j^2 - 4*a^2*c^6*d*e^
2*k - 2*b^3*c^5*d^2*e*j + 8*a^5*b^2*c*d*m^3 + 8*a*b^6*c*d^2*m^2 + 8*a*b^2*c
^5*d^3*m - 6*a^5*b*c^2*e*l^3 - 6*a^2*b*c^5*e^3*l - 4*a^2*c^6*e^2*f*h + 2*b^
3*c^5*d^2*f*h + 2*a^4*b^3*c*e*l^3 + 2*a^4*b*c^3*g*j^3 + 2*a^3*b*c^4*g^3*j +
 2*a*b^3*c^4*e^3*l + 4*a^2*c^6*e*f^2*g + 4*a^2*c^6*d*f^2*h - 6*a^4*b*c^3*d*
k^3 - 4*a^2*c^6*d*f*g^2 + 2*b^2*c^6*d^2*e*g - 2*a*b^2*c^5*e^3*j + 2*a^3*b*c
^4*f*h^3 + 2*a^2*b*c^5*f^3*h + 2*a^2*b*c^5*e*g^3 + 3*a*b*c^6*d^2*g^2 - 9*a^
4*b^2*c^2*f^2*m^2 + 4*a^4*b^2*c^2*g^2*l^2 - 14*a^3*b^3*c^2*e^2*m^2 + 5*a^3*
b^3*c^2*f^2*l^2 - 20*a^2*b^4*c^2*d^2*m^2 + 16*a^3*b^2*c^3*d^2*m^2 - 9*a^3*b
^2*c^3*e^2*l^2 + 6*a^2*b^4*c^2*e^2*l^2 + 4*a^3*b^2*c^3*f^2*k^2 - 14*a^2*b^3
*c^3*d^2*l^2 + 5*a^2*b^3*c^3*e^2*k^2 - 9*a^2*b^2*c^4*d^2*k^2 + 4*a^2*b^2*c^
4*e^2*j^2 + 4*a^7*c*k*l^2*m - 4*a^7*c*j*l*m^2 + 2*b^7*c*d^2*k*m + 2*a^6*b*c
*k^3*m + 2*a^6*b*c*j*l^3 + 2*a*b^7*d*f*m^2 - 6*a^6*b*c*f*m^3 - 6*a*b*c^6*d^
3*k - 4*a*c^7*d^2*e*g + 4*a*c^7*d*e^2*f + 2*a*b*c^6*e^3*g + 2*a*b*c^6*d*f^3
 - a^5*b^2*c*j^2*l^2 - a^5*b*c^2*j^2*k^2 - a^4*b^3*c*h^2*l^2 - a^3*b^4*c*g^
2*l^2 - a^4*b*c^3*h^2*j^2 - a^2*b^5*c*f^2*l^2 - a*b^5*c^2*e^2*k^2 - a^3*b*c
^4*g^2*h^2 - a*b^4*c^3*e^2*j^2 - a^2*b*c^5*f^2*g^2 - a*b^3*c^4*e^2*h^2 - a*
b^2*c^5*e^2*g^2 + 2*a^7*b*k*m^3 + 4*a^7*c*h*m^3 + 4*a*c^7*d^3*h + 2*b*c^7*d
^3*f - a^6*b*c*k^2*l^2 - 2*a^6*c^2*j^2*l^2 - 6*a^6*c^2*h^2*m^2 - a*b^6*c*e^
2*l^2 - 6*a^5*c^3*g^2*l^2 - 2*a^5*c^3*h^2*k^2 - 2*a^5*c^3*f^2*m^2 - 6*a^4*c
^4*f^2*k^2 - 6*a^4*c^4*d^2*m^2 - 2*a^4*c^4*g^2*j^2 - 2*a^4*c^4*e^2*l^2 - 6*
a^3*c^5*e^2*j^2 - 2*a^3*c^5*d^2*k^2 - 2*a^3*c^5*f^2*h^2 - a*b*c^6*e^2*f^2 -
 6*a^2*c^6*d^2*h^2 - 2*a^2*c^6*e^2*g^2 - a^4*b^2*c^2*h^2*k^2 - a^3*b^3*c^2*
g^2*k^2 - a^3*b^2*c^3*g^2*j^2 - a^2*b^4*c^2*f^2*k^2 - a^2*b^3*c^3*f^2*j^2 -
 a^2*b^2*c^4*f^2*h^2 - 2*a^7*c*k^2*m^2 + 4*a^5*c^3*h^3*m - 2*a^6*b^2*h*m^3
 + 4*a^6*c^2*g*l^3 + 4*a^4*c^4*g^3*l - 2*b^4*c^4*d^3*m + 2*a^5*b^3*f*m^3 - 4
*a^6*c^2*d*m^3 + 4*a^5*c^3*f*k^3 + 4*a^3*c^5*f^3*k - 4*a^2*c^6*d^3*m + 2*b^
3*c^5*d^3*k - 2*a^4*b^4*d*m^3 + 4*a^4*c^4*e*j^3 + 4*a^2*c^6*e^3*j - 2*b^2*c

```
^6*d^3*h + 4*a^3*c^5*d*h^3 - 2*a*c^7*d^2*f^2 - a^6*b^2*k^2*m^2 - a^5*b^3*j^
2*m^2 - a^4*b^4*h^2*m^2 - a^3*b^5*g^2*m^2 - a^2*b^6*f^2*m^2 - b^6*c^2*d^2*k
^2 - b^5*c^3*d^2*j^2 - b^4*c^4*d^2*h^2 - b^3*c^5*d^2*g^2 - b^2*c^6*d^2*f^2
- a^7*b*l^2*m^2 - b^7*c*d^2*l^2 - a*b^7*e^2*m^2 - b*c^7*d^2*e^2 - b^8*d^2*m
^2 - a^6*c^2*k^4 - a^5*c^3*j^4 - a^4*c^4*h^4 - a^3*c^5*g^4 - a^2*c^6*f^4 -
a^7*c*l^4 - a*c^7*e^4 - a^8*m^4 - c^8*d^4, z, k1), k1, 1, 4) + (l*x^4)/(4*c
) + (m*x^5)/(5*c)
```

3.26 $\int \frac{d+ex}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 18, antiderivative size = 94

$$\int \frac{d+ex}{(4-5x^2+x^4)^2} dx = \frac{dx(17-5x^2)}{72(4-5x^2+x^4)} + \frac{e(5-2x^2)}{18(4-5x^2+x^4)} + \frac{19}{432} d\,\text{arctanh}\left(\frac{x}{2}\right) - \frac{1}{54} d\,\text{arctanh}(x) + \frac{1}{27} e \log(1-x^2) - \frac{1}{27} e \log(4-x^2)$$

```
[Out] 1/72*d*x*(-5*x^2+17)/(x^4-5*x^2+4)+1/18*e*(-2*x^2+5)/(x^4-5*x^2+4)+19/432*d
*arctanh(1/2*x)-1/54*d*arctanh(x)+1/27*e*ln(-x^2+1)-1/27*e*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 94, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.500, Rules used = {1687, 12, 1106, 1180, 213, 1121, 628, 630, 31}

$$\int \frac{d+ex}{(4-5x^2+x^4)^2} dx = \frac{19}{432} d\,\text{arctanh}\left(\frac{x}{2}\right) - \frac{1}{54} d\,\text{arctanh}(x) + \frac{dx(17-5x^2)}{72(x^4-5x^2+4)} + \frac{1}{27} e \log(1-x^2) - \frac{1}{27} e \log(4-x^2) + \frac{e(5-2x^2)}{18(x^4-5x^2+4)}$$

```
[In] Int[(d + e*x)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d*x*(17 - 5*x^2))/(72*(4 - 5*x^2 + x^4)) + (e*(5 - 2*x^2))/(18*(4 - 5*x^2
+ x^4)) + (19*d*ArcTanh[x/2])/432 - (d*ArcTanh[x])/54 + (e*Log[1 - x^2])/27
 - (e*Log[4 - x^2])/27
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
```

```
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d}{(4-5x^2+x^4)^2}\,dx+\int \frac{ex}{(4-5x^2+x^4)^2}\,dx\\
&= d\int \frac{1}{(4-5x^2+x^4)^2}\,dx+e\int \frac{x}{(4-5x^2+x^4)^2}\,dx\\
&= \frac{dx(17-5x^2)}{72\,(4-5x^2+x^4)}-\frac{1}{72}d\int \frac{-1+5x^2}{4-5x^2+x^4}\,dx+\frac{1}{2}e\text{Subst}\left(\int \frac{1}{(4-5x+x^2)^2}\,dx,x,x^2\right)\\
&= \frac{dx(17-5x^2)}{72\,(4-5x^2+x^4)}+\frac{e(5-2x^2)}{18\,(4-5x^2+x^4)}+\frac{1}{54}d\int \frac{1}{-1+x^2}\,dx\\
&\quad -\frac{1}{216}(19d)\int \frac{1}{-4+x^2}\,dx-\frac{1}{9}e\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx,x,x^2\right)\\
&= \frac{dx(17-5x^2)}{72\,(4-5x^2+x^4)}+\frac{e(5-2x^2)}{18\,(4-5x^2+x^4)}+\frac{19}{432}d\tanh^{-1}\left(\frac{x}{2}\right)-\frac{1}{54}d\tanh^{-1}(x)\\
&\quad -\frac{1}{27}e\text{Subst}\left(\int \frac{1}{-4+x}\,dx,x,x^2\right)+\frac{1}{27}e\text{Subst}\left(\int \frac{1}{-1+x}\,dx,x,x^2\right)\\
&= \frac{dx(17-5x^2)}{72\,(4-5x^2+x^4)}+\frac{e(5-2x^2)}{18\,(4-5x^2+x^4)}+\frac{19}{432}d\tanh^{-1}\left(\frac{x}{2}\right)\\
&\quad -\frac{1}{54}d\tanh^{-1}(x)+\frac{1}{27}e\log\left(1-x^2\right)-\frac{1}{27}e\log\left(4-x^2\right)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.96

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = \frac{1}{864}\left(\frac{12(e(20-8x^2)+dx(17-5x^2))}{4-5x^2+x^4}+8(d+4e)\log(1-x)\right.$$
$$-(19d+32e)\log(2-x)-8(d-4e)\log(1+x)$$
$$\left.+(19d-32e)\log(2+x)\right)$$

```
[In] Integrate[(d + e*x)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(e*(20 - 8*x^2) + d*x*(17 - 5*x^2)))/(4 - 5*x^2 + x^4) + 8*(d + 4*e)*L
og[1 - x] - (19*d + 32*e)*Log[2 - x] - 8*(d - 4*e)*Log[1 + x] + (19*d - 32*
e)*Log[2 + x])/864
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.88

method	result
norman	$\frac{-\frac{1}{9}e\,x^2+\frac{17}{72}dx-\frac{5}{72}x^3d+\frac{5}{18}e}{x^4-5x^2+4}+\left(-\frac{19d}{864}-\frac{e}{27}\right)\ln(x-2)+\left(-\frac{d}{108}+\frac{e}{27}\right)\ln(x+1)+\left(\frac{d}{108}+\frac{e}{27}\right)\ln(x-$
risch	$\frac{-\frac{1}{9}e\,x^2+\frac{17}{72}dx-\frac{5}{72}x^3d+\frac{5}{18}e}{x^4-5x^2+4}-\frac{\ln(x+1)d}{108}+\frac{\ln(x+1)e}{27}+\frac{\ln(1-x)d}{108}+\frac{\ln(1-x)e}{27}-\frac{19\ln(2-x)d}{864}-\frac{\ln(2-x)e}{27}+\frac{19\ln(x+}{864}$
default	$-\frac{\frac{d}{144}-\frac{e}{72}}{x+2}+\left(\frac{19d}{864}-\frac{e}{27}\right)\ln(x+2)+\left(-\frac{d}{108}+\frac{e}{27}\right)\ln(x+1)-\frac{\frac{d}{36}-\frac{e}{36}}{x+1}-\frac{\frac{d}{36}+\frac{e}{36}}{x-1}+\left(\frac{d}{108}+\frac{e}{27}\right)\ln(x$
parallelrisch	$-\frac{-240e-204dx+76\ln(x-2)d+128\ln(x-2)e-32\ln(x-1)d-128\ln(x-1)e+32\ln(x-2)x^4e+96e\,x^2-160\ln(x-2)x^2e+40\ln(x-}{}$

```
[In] int((e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (-1/9*e*x^2+17/72*d*x-5/72*x^3*d+5/18*e)/(x^4-5*x^2+4)+(-19/864*d-1/27*e)*l
n(x-2)+(-1/108*d+1/27*e)*ln(x+1)+(1/108*d+1/27*e)*ln(x-1)+(19/864*d-1/27*e)
*ln(x+2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 169 vs. $2(80) = 160$.

Time = 0.28 (sec) , antiderivative size = 169, normalized size of antiderivative = 1.80

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = -\frac{60\,dx^3 + 96\,ex^2 - 204\,dx - ((19\,d-32\,e)x^4 - 5\,(19\,d-32\,e)x^2 + 76\,d - 128\,e)\log(x+2) + 8\,((d-4\,e}{}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/864*(60*d*x^3 + 96*e*x^2 - 204*d*x - ((19*d - 32*e)*x^4 - 5*(19*d - 32*e
)*x^2 + 76*d - 128*e)*log(x + 2) + 8*((d - 4*e)*x^4 - 5*(d - 4*e)*x^2 + 4*d
 - 16*e)*log(x + 1) - 8*((d + 4*e)*x^4 - 5*(d + 4*e)*x^2 + 4*d + 16*e)*log(
x - 1) + ((19*d + 32*e)*x^4 - 5*(19*d + 32*e)*x^2 + 76*d + 128*e)*log(x - 2
) - 240*e)/(x^4 - 5*x^2 + 4)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 604 vs. $2(78) = 156$.

Time = 2.04 (sec) , antiderivative size = 604, normalized size of antiderivative = 6.43

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = -\frac{(d-4e)\log\left(x+\frac{-6006260d^4e+2341251d^4(d-4e)-18247680d^2e^3+24099840d^2e^2(d-4e)+7387904d^2e(d-4e)^2-665280d^2(d-4e)^3+5}{1675971d^5-66150400d^3e^2+318767104de^4}\right.}{108}$$
$$+\frac{(d+4e)\log\left(x+\frac{-6006260d^4e-2341251d^4(d+4e)-18247680d^2e^3-24099840d^2e^2(d+4e)+7387904d^2e(d+4e)^2+665280d^2(d+4e)^3+58}{1675971d^5-66150400d^3e^2+318767104de^4}\right.}{108}$$
$$+\frac{(19d-32e)\log\left(x+\frac{-6006260d^4e-\frac{2341251d^4\cdot(19d-32e)}{8}-18247680d^2e^3-3012480d^2e^2\cdot(19d-32e)+115436d^2e(19d-32e)^2+\frac{10395d^2(}{}}{1675971d^5-66150400d^3e^2+31}\right.}{864}$$
$$-\frac{(19d+32e)\log\left(x+\frac{-6006260d^4e+\frac{2341251d^4\cdot(19d+32e)}{8}-18247680d^2e^3+3012480d^2e^2\cdot(19d+32e)+115436d^2e(19d+32e)^2-\frac{10395d^2(}{}}{1675971d^5-66150400d^3e^2+31}\right.}{864}$$
$$+\frac{-5dx^3+17dx-8ex^2+20e}{72x^4-360x^2+288}$$

```
[In] integrate((e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] -(d - 4*e)*log(x + (-6006260*d**4*e + 2341251*d**4*(d - 4*e) - 18247680*d**
2*e**3 + 24099840*d**2*e**2*(d - 4*e) + 7387904*d**2*e*(d - 4*e)**2 - 66528
0*d**2*(d - 4*e)**3 + 587202560*e**5 - 12582912*e**4*(d - 4*e) - 36700160*e
```

```
**3*(d - 4*e)**2 + 786432*e**2*(d - 4*e)**3)/(1675971*d**5 - 66150400*d**3*
e**2 + 318767104*d*e**4))/108 + (d + 4*e)*log(x + (-6006260*d**4*e - 234125
1*d**4*(d + 4*e) - 18247680*d**2*e**3 - 24099840*d**2*e**2*(d + 4*e) + 7387
904*d**2*e*(d + 4*e)**2 + 665280*d**2*(d + 4*e)**3 + 587202560*e**5 + 12582
912*e**4*(d + 4*e) - 36700160*e**3*(d + 4*e)**2 - 786432*e**2*(d + 4*e)**3)
/(1675971*d**5 - 66150400*d**3*e**2 + 318767104*d*e**4))/108 + (19*d - 32*e
)*log(x + (-6006260*d**4*e - 2341251*d**4*(19*d - 32*e)/8 - 18247680*d**2*e
**3 - 3012480*d**2*e**2*(19*d - 32*e) + 115436*d**2*e*(19*d - 32*e)**2 + 10
395*d**2*(19*d - 32*e)**3/8 + 587202560*e**5 + 1572864*e**4*(19*d - 32*e) -
 573440*e**3*(19*d - 32*e)**2 - 1536*e**2*(19*d - 32*e)**3)/(1675971*d**5 -
 66150400*d**3*e**2 + 318767104*d*e**4))/864 - (19*d + 32*e)*log(x + (-6006
260*d**4*e + 2341251*d**4*(19*d + 32*e)/8 - 18247680*d**2*e**3 + 3012480*d*
*2*e**2*(19*d + 32*e) + 115436*d**2*e*(19*d + 32*e)**2 - 10395*d**2*(19*d +
 32*e)**3/8 + 587202560*e**5 - 1572864*e**4*(19*d + 32*e) - 573440*e**3*(19
*d + 32*e)**2 + 1536*e**2*(19*d + 32*e)**3)/(1675971*d**5 - 66150400*d**3*e
**2 + 318767104*d*e**4))/864 + (-5*d*x**3 + 17*d*x - 8*e*x**2 + 20*e)/(72*x
**4 - 360*x**2 + 288)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 83, normalized size of antiderivative = 0.88

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = \frac{1}{864}\,(19\,d-32\,e)\log{(x+2)}-\frac{1}{108}\,(d-4\,e)\log{(x+1)} + \frac{1}{108}\,(d+4\,e)\log{(x-1)}-\frac{1}{864}\,(19\,d+32\,e)\log{(x-2)} - \frac{5\,dx^3+8\,ex^2-17\,dx-20\,e}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/864*(19*d - 32*e)*log(x + 2) - 1/108*(d - 4*e)*log(x + 1) + 1/108*(d + 4*
e)*log(x - 1) - 1/864*(19*d + 32*e)*log(x - 2) - 1/72*(5*d*x^3 + 8*e*x^2 -
17*d*x - 20*e)/(x^4 - 5*x^2 + 4)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 87, normalized size of antiderivative = 0.93

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = \frac{1}{864}\,(19\,d-32\,e)\log\left(|x+2|\right)-\frac{1}{108}\,(d-4\,e)\log\left(|x+1|\right) + \frac{1}{108}\,(d+4\,e)\log\left(|x-1|\right)-\frac{1}{864}\,(19\,d+32\,e)\log\left(|x-2|\right) - \frac{5\,dx^3+8\,ex^2-17\,dx-20\,e}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/864*(19*d - 32*e)*log(abs(x + 2)) - 1/108*(d - 4*e)*log(abs(x + 1)) + 1/1
08*(d + 4*e)*log(abs(x - 1)) - 1/864*(19*d + 32*e)*log(abs(x - 2)) - 1/72*(
5*d*x^3 + 8*e*x^2 - 17*d*x - 20*e)/(x^4 - 5*x^2 + 4)
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.89

$$\int \frac{d+ex}{(4-5x^2+x^4)^2}\,dx = \ln\left(x-1\right)\,\left(\frac{d}{108}+\frac{e}{27}\right)-\ln\left(x+1\right)\,\left(\frac{d}{108}-\frac{e}{27}\right) - \ln\left(x-2\right)\,\left(\frac{19\,d}{864}+\frac{e}{27}\right)+\ln\left(x+2\right)\,\left(\frac{19\,d}{864}-\frac{e}{27}\right) + \frac{-\frac{5\,d\,x^3}{72}-\frac{e\,x^2}{9}+\frac{17\,d\,x}{72}+\frac{5\,e}{18}}{x^4-5\,x^2+4}$$

```
[In] int((d + e*x)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/108 + e/27) - log(x + 1)*(d/108 - e/27) - log(x - 2)*((19*d)/
864 + e/27) + log(x + 2)*((19*d)/864 - e/27) + ((5*e)/18 + (17*d*x)/72 - (5
*d*x^3)/72 - (e*x^2)/9)/(x^4 - 5*x^2 + 4)
```

3.27 $\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2} \, dx$

Optimal result

Integrand size = 23, antiderivative size = 115

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2} \, dx = \frac{e(5-2x^2)}{18(4-5x^2+x^4)} + \frac{x(17d+20f-(5d+8f)x^2)}{72(4-5x^2+x^4)} + \frac{1}{432}(19d+52f)\text{arctanh}\left(\frac{x}{2}\right) - \frac{1}{54}(d+7f)\text{arctanh}(x) + \frac{1}{27}e\log(1-x^2) - \frac{1}{27}e\log(4-x^2)$$

```
[Out] 1/18*e*(-2*x^2+5)/(x^4-5*x^2+4)+1/72*x*(17*d+20*f-(5*d+8*f)*x^2)/(x^4-5*x^2
+4)+1/432*(19*d+52*f)*arctanh(1/2*x)-1/54*(d+7*f)*arctanh(x)+1/27*e*ln(-x^2
+1)-1/27*e*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 115, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.391, Rules used = {1687, 1192, 1180, 213, 12, 1121, 628, 630, 31}

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2} \, dx = \frac{1}{432}\text{arctanh}\left(\frac{x}{2}\right)(19d+52f) - \frac{1}{54}\text{arctanh}(x)(d+7f) + \frac{x(-(x^2(5d+8f))+17d+20f)}{72(x^4-5x^2+4)} + \frac{1}{27}e\log(1-x^2) - \frac{1}{27}e\log(4-x^2) + \frac{e(5-2x^2)}{18(x^4-5x^2+4)}$$

```
[In] Int[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (e*(5 - 2*x^2))/(18*(4 - 5*x^2 + x^4)) + (x*(17*d + 20*f - (5*d + 8*f)*x^2)
)/(72*(4 - 5*x^2 + x^4)) + ((19*d + 52*f)*ArcTanh[x/2])/432 - ((d + 7*f)*Ar
cTanh[x])/54 + (e*Log[1 - x^2])/27 - (e*Log[4 - x^2])/27
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{ex}{(4-5x^2+x^4)^2}\,dx + \int \frac{d+fx^2}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} - \frac{1}{72}\int \frac{-d+20f+(5d+8f)x^2}{4-5x^2+x^4}\,dx + e\int \frac{x}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(4-5x+x^2)^2}\,dx, x, x^2\right) \\
&\quad - \frac{1}{54}(-d-7f)\int \frac{1}{-1+x^2}\,dx - \frac{1}{216}(19d+52f)\int \frac{1}{-4+x^2}\,dx \\
&= \frac{e(5-2x^2)}{18\,(4-5x^2+x^4)} + \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f)\tanh^{-1}\left(\frac{x}{2}\right) \\
&\quad - \frac{1}{54}(d+7f)\tanh^{-1}(x) - \frac{1}{9}e\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right) \\
&= \frac{e(5-2x^2)}{18\,(4-5x^2+x^4)} + \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} \\
&\quad + \frac{1}{432}(19d+52f)\tanh^{-1}\left(\frac{x}{2}\right) - \frac{1}{54}(d+7f)\tanh^{-1}(x) \\
&\quad - \frac{1}{27}e\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) + \frac{1}{27}e\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) \\
&= \frac{e(5-2x^2)}{18\,(4-5x^2+x^4)} + \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f)\tanh^{-1}\left(\frac{x}{2}\right) \\
&\quad - \frac{1}{54}(d+7f)\tanh^{-1}(x) + \frac{1}{27}e\log\left(1-x^2\right) - \frac{1}{27}e\log\left(4-x^2\right)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 112, normalized size of antiderivative = 0.97

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx = \frac{1}{864}\left(\frac{12(17dx+20fx-5dx^3-8fx^3+e(20-8x^2))}{4-5x^2+x^4} + 8(d+4e+7f)\log(1-x)-(19d+32e+52f)\log(2-x) - 8(d-4e+7f)\log(1+x)+(19d-32e+52f)\log(2+x)\right)$$

```
[In] Integrate[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(17*d*x + 20*f*x - 5*d*x^3 - 8*f*x^3 + e*(20 - 8*x^2)))/(4 - 5*x^2 + x
^4) + 8*(d + 4*e + 7*f)*Log[1 - x] - (19*d + 32*e + 52*f)*Log[2 - x] - 8*(d
 - 4*e + 7*f)*Log[1 + x] + (19*d - 32*e + 52*f)*Log[2 + x])/864
```

Maple [A] (verified)

Time = 0.09 (sec) , antiderivative size = 105, normalized size of antiderivative = 0.91

method	result
norman	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}\right)x-\frac{e\,x^2}{9}+\frac{5e}{18}}{x^4-5x^2+4}+\left(-\frac{19d}{864}-\frac{e}{27}-\frac{13f}{216}\right)\ln\left(x-2\right)+\left(-\frac{d}{108}+\frac{e}{27}-\frac{7f}{108}\right)\ln\left(x+1\right)$
default	$-\frac{\frac{d}{144}-\frac{e}{72}+\frac{f}{36}}{x+2}+\left(\frac{19d}{864}-\frac{e}{27}+\frac{13f}{216}\right)\ln\left(x+2\right)+\left(-\frac{d}{108}+\frac{e}{27}-\frac{7f}{108}\right)\ln\left(x+1\right)-\frac{\frac{d}{36}-\frac{e}{36}+\frac{f}{36}}{x+1}-$
risch	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}\right)x-\frac{e\,x^2}{9}+\frac{5e}{18}}{x^4-5x^2+4}+\frac{19\ln(x+2)d}{864}-\frac{\ln(x+2)e}{27}+\frac{13\ln(x+2)f}{216}-\frac{\ln(x+1)d}{108}+\frac{\ln(x+1)e}{27}-$
parallelrisch	$-\frac{-240e+96f\,x^3-204dx+76\ln(x-2)d+128\ln(x-2)e-32\ln(x-1)d-128\ln(x-1)e+32\ln(x-2)x^4e-208\ln(x+2)f+224\ln(x}{}$

```
[In] int((f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] ((-5/72*d-1/9*f)*x^3+(17/72*d+5/18*f)*x-1/9*e*x^2+5/18*e)/(x^4-5*x^2+4)+(-1
9/864*d-1/27*e-13/216*f)*ln(x-2)+(-1/108*d+1/27*e-7/108*f)*ln(x+1)+(1/108*d
+1/27*e+7/108*f)*ln(x-1)+(19/864*d-1/27*e+13/216*f)*ln(x+2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 217 vs. 2(100) = 200.

Time = 0.31 (sec) , antiderivative size = 217, normalized size of antiderivative = 1.89

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(5\,d+8\,f)x^3+96\,ex^2-12\,(17\,d+20\,f)x-((19\,d-32\,e+52\,f)x^4-5\,(19\,d-32\,e+52\,f)x^2+76\,d}{}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/864*(12*(5*d + 8*f)*x^3 + 96*e*x^2 - 12*(17*d + 20*f)*x - ((19*d - 32*e
+ 52*f)*x^4 - 5*(19*d - 32*e + 52*f)*x^2 + 76*d - 128*e + 208*f)*log(x + 2)
 + 8*((d - 4*e + 7*f)*x^4 - 5*(d - 4*e + 7*f)*x^2 + 4*d - 16*e + 28*f)*log(
x + 1) - 8*((d + 4*e + 7*f)*x^4 - 5*(d + 4*e + 7*f)*x^2 + 4*d + 16*e + 28*f
)*log(x - 1) + ((19*d + 32*e + 52*f)*x^4 - 5*(19*d + 32*e + 52*f)*x^2 + 76*
d + 128*e + 208*f)*log(x - 2) - 240*e)/(x^4 - 5*x^2 + 4)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 106, normalized size of antiderivative = 0.92

$$\begin{aligned}\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{864}\,(19\,d-32\,e+52\,f)\log(x+2) \\ &\quad -\frac{1}{108}\,(d-4\,e+7\,f)\log(x+1)+\frac{1}{108}\,(d+4\,e+7\,f)\log(x-1) \\ &\quad -\frac{1}{864}\,(19\,d+32\,e+52\,f)\log(x-2) \\ &\quad -\frac{(5\,d+8\,f)x^3+8\,ex^2-(17\,d+20\,f)x-20\,e}{72\,(x^4-5\,x^2+4)}\end{aligned}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/864*(19*d - 32*e + 52*f)*log(x + 2) - 1/108*(d - 4*e + 7*f)*log(x + 1) +
1/108*(d + 4*e + 7*f)*log(x - 1) - 1/864*(19*d + 32*e + 52*f)*log(x - 2) -
1/72*((5*d + 8*f)*x^3 + 8*e*x^2 - (17*d + 20*f)*x - 20*e)/(x^4 - 5*x^2 + 4)
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 109, normalized size of antiderivative = 0.95

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx = \frac{1}{864}\,(19\,d-32\,e+52\,f)\log\left(|x+2|\right) - \frac{1}{108}\,(d-4\,e+7\,f)\log\left(|x+1|\right) + \frac{1}{108}\,(d+4\,e+7\,f)\log\left(|x-1|\right) - \frac{1}{864}\,(19\,d+32\,e+52\,f)\log\left(|x-2|\right) - \frac{5\,dx^3+8\,fx^3+8\,ex^2-17\,dx-20\,fx-20\,e}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/864*(19*d - 32*e + 52*f)*log(abs(x + 2)) - 1/108*(d - 4*e + 7*f)*log(abs(
x + 1)) + 1/108*(d + 4*e + 7*f)*log(abs(x - 1)) - 1/864*(19*d + 32*e + 52*f
)*log(abs(x - 2)) - 1/72*(5*d*x^3 + 8*f*x^3 + 8*e*x^2 - 17*d*x - 20*f*x - 2
0*e)/(x^4 - 5*x^2 + 4)
```

Mupad [B] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 107, normalized size of antiderivative = 0.93

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^2}\,dx = \ln(x-1)\left(\frac{d}{108}+\frac{e}{27}+\frac{7\,f}{108}\right) - \ln(x+1)\left(\frac{d}{108}-\frac{e}{27}+\frac{7\,f}{108}\right) - \ln(x-2)\left(\frac{19\,d}{864}+\frac{e}{27}+\frac{13\,f}{216}\right) + \ln(x+2)\left(\frac{19\,d}{864}-\frac{e}{27}+\frac{13\,f}{216}\right) + \frac{\left(-\frac{5\,d}{72}-\frac{f}{9}\right)x^3-\frac{e\,x^2}{9}+\left(\frac{17\,d}{72}+\frac{5\,f}{18}\right)x+\frac{5\,e}{18}}{x^4-5\,x^2+4}$$

```
[In] int((d + e*x + f*x^2)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/108 + e/27 + (7*f)/108) - log(x + 1)*(d/108 - e/27 + (7*f)/10
8) - log(x - 2)*((19*d)/864 + e/27 + (13*f)/216) + log(x + 2)*((19*d)/864 -
 e/27 + (13*f)/216) + ((5*e)/18 - x^3*((5*d)/72 + f/9) - (e*x^2)/9 + x*((17
*d)/72 + (5*f)/18))/(x^4 - 5*x^2 + 4)
```

3.28 $\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 28, antiderivative size = 138

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx &= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)}+\frac{5e+8g-(2e+5g)x^2}{18\,(4-5x^2+x^4)}\\ &\quad+\frac{1}{432}(19d+52f)\operatorname{arctanh}\left(\frac{x}{2}\right)-\frac{1}{54}(d+7f)\operatorname{arctanh}(x)\\ &\quad+\frac{1}{54}(2e+5g)\log\left(1-x^2\right)-\frac{1}{54}(2e+5g)\log\left(4-x^2\right)\end{aligned}$$

```
[Out] 1/72*x*(17*d+20*f-(5*d+8*f)*x^2)/(x^4-5*x^2+4)+1/18*(5*e+8*g-(2*e+5*g)*x^2)
/(x^4-5*x^2+4)+1/432*(19*d+52*f)*arctanh(1/2*x)-1/54*(d+7*f)*arctanh(x)+1/5
4*(2*e+5*g)*ln(-x^2+1)-1/54*(2*e+5*g)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 138, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.286, Rules used = {1687, 1192, 1180, 213, 1261, 652, 630, 31}

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{432}\operatorname{arctanh}\left(\frac{x}{2}\right)(19d+52f)-\frac{1}{54}\operatorname{arctanh}(x)(d+7f)\\ &\quad+\frac{x(-(x^2(5d+8f))+17d+20f)}{72\,(x^4-5x^2+4)}+\frac{1}{54}(2e+5g)\log\left(1-x^2\right)\\ &\quad-\frac{1}{54}(2e+5g)\log\left(4-x^2\right)+\frac{-(x^2(2e+5g))+5e+8g}{18\,(x^4-5x^2+4)}\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (x*(17*d + 20*f - (5*d + 8*f)*x^2))/(72*(4 - 5*x^2 + x^4)) + (5*e + 8*g - (
2*e + 5*g)*x^2)/(18*(4 - 5*x^2 + x^4)) + ((19*d + 52*f)*ArcTanh[x/2])/432 -
 ((d + 7*f)*ArcTanh[x])/54 + ((2*e + 5*g)*Log[1 - x^2])/54 - ((2*e + 5*g)*L
og[4 - x^2])/54
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
```

```
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{(4-5x^2+x^4)^2}\,dx + \int \frac{x(e+gx^2)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} - \frac{1}{72}\int \frac{-d+20f+(5d+8f)x^2}{4-5x^2+x^4}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(4-5x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g-(2e+5g)x^2}{18\,(4-5x^2+x^4)} - \frac{1}{54}(-d-7f)\int \frac{1}{-1+x^2}\,dx \\
&\quad - \frac{1}{216}(19d+52f)\int \frac{1}{-4+x^2}\,dx + \frac{1}{18}(-2e-5g)\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right) \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g-(2e+5g)x^2}{18\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f)\tanh^{-1}\left(\frac{x}{2}\right) \\
&\quad - \frac{1}{54}(d+7f)\tanh^{-1}(x) + \frac{1}{54}(-2e-5g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) \\
&\quad + \frac{1}{54}(2e+5g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) \\
&= \frac{x(17d+20f-(5d+8f)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g-(2e+5g)x^2}{18\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f)\tanh^{-1}\left(\frac{x}{2}\right) \\
&\quad - \frac{1}{54}(d+7f)\tanh^{-1}(x) + \frac{1}{54}(2e+5g)\log\left(1-x^2\right) - \frac{1}{54}(2e+5g)\log\left(4-x^2\right)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 134, normalized size of antiderivative = 0.97

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx \\
&=\frac{1}{864}\left(\frac{12(17dx+20fx-5dx^3-8fx^3+e(20-8x^2)-4g(-8+5x^2))}{4-5x^2+x^4}\right. \\
&\quad +8(d+4e+7f+10g)\log(1-x)-(19d+32e+52f+80g)\log(2-x) \\
&\quad \left.-8(d-4e+7f-10g)\log(1+x)+(19d-32e+52f-80g)\log(2+x)\right)
\end{aligned}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(17*d*x + 20*f*x - 5*d*x^3 - 8*f*x^3 + e*(20 - 8*x^2) - 4*g*(-8 + 5*x^
2)))/(4 - 5*x^2 + x^4) + 8*(d + 4*e + 7*f + 10*g)*Log[1 - x] - (19*d + 32*e
 + 52*f + 80*g)*Log[2 - x] - 8*(d - 4*e + 7*f - 10*g)*Log[1 + x] + (19*d -
32*e + 52*f - 80*g)*Log[2 + x])/864
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 125, normalized size of antiderivative = 0.91

method	result
norman	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}\right)x+\left(-\frac{e}{9}-\frac{5g}{18}\right)x^2+\frac{5e}{18}+\frac{4g}{9}}{x^4-5x^2+4}+\left(-\frac{19d}{864}-\frac{e}{27}-\frac{13f}{216}-\frac{5g}{54}\right)\ln(x-2)+\left(-\frac{d}{108}+\frac{e}{27}-\right.$
default	$-\frac{\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}}{x+2}+\left(\frac{19d}{864}-\frac{e}{27}+\frac{13f}{216}-\frac{5g}{54}\right)\ln(x+2)+\left(-\frac{d}{108}+\frac{e}{27}-\frac{7f}{108}+\frac{5g}{54}\right)\ln(x+1)-$
risch	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}\right)x+\left(-\frac{e}{9}-\frac{5g}{18}\right)x^2+\frac{5e}{18}+\frac{4g}{9}}{x^4-5x^2+4}+\frac{19\ln(x+2)d}{864}-\frac{\ln(x+2)e}{27}+\frac{13\ln(x+2)f}{216}-\frac{5\ln(x+2)g}{54}+$
parallelrisch	$-\frac{-384g-240e+96f x^3+240g x^2-204dx+76\ln(x-2)d+128\ln(x-2)e-32\ln(x-1)d-128\ln(x-1)e+32\ln(x-2)x^4e-208\ln(x}{}$

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] ((-5/72*d-1/9*f)*x^3+(17/72*d+5/18*f)*x+(-1/9*e-5/18*g)*x^2+5/18*e+4/9*g)/(
x^4-5*x^2+4)+(-19/864*d-1/27*e-13/216*f-5/54*g)*ln(x-2)+(-1/108*d+1/27*e-7/
108*f+5/54*g)*ln(x+1)+(1/108*d+1/27*e+7/108*f+5/54*g)*ln(x-1)+(19/864*d-1/2
7*e+13/216*f-5/54*g)*ln(x+2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 262 vs. 2(122) = 244.

Time = 0.49 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.90

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(5\,d+8\,f)x^3+48\,(2\,e+5\,g)x^2-12\,(17\,d+20\,f)x-((19\,d-32\,e+52\,f-80\,g)x^4-5\,(19\,d-32\,e}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/864*(12*(5*d + 8*f)*x^3 + 48*(2*e + 5*g)*x^2 - 12*(17*d + 20*f)*x - ((19
*d - 32*e + 52*f - 80*g)*x^4 - 5*(19*d - 32*e + 52*f - 80*g)*x^2 + 76*d - 1
28*e + 208*f - 320*g)*log(x + 2) + 8*((d - 4*e + 7*f - 10*g)*x^4 - 5*(d - 4
*e + 7*f - 10*g)*x^2 + 4*d - 16*e + 28*f - 40*g)*log(x + 1) - 8*((d + 4*e +
 7*f + 10*g)*x^4 - 5*(d + 4*e + 7*f + 10*g)*x^2 + 4*d + 16*e + 28*f + 40*g)
*log(x - 1) + ((19*d + 32*e + 52*f + 80*g)*x^4 - 5*(19*d + 32*e + 52*f + 80
*g)*x^2 + 76*d + 128*e + 208*f + 320*g)*log(x - 2) - 240*e - 384*g)/(x^4 -
5*x^2 + 4)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 127, normalized size of antiderivative = 0.92

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g)\log(x+2)\\ &\quad-\frac{1}{108}\,(d-4\,e+7\,f-10\,g)\log(x+1)\\ &\quad+\frac{1}{108}\,(d+4\,e+7\,f+10\,g)\log(x-1)\\ &\quad-\frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g)\log(x-2)\\ &\quad-\frac{(5\,d+8\,f)x^3+4\,(2\,e+5\,g)x^2-(17\,d+20\,f)x-20\,e-32\,g}{72\,(x^4-5\,x^2+4)}\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/864*(19*d - 32*e + 52*f - 80*g)*log(x + 2) - 1/108*(d - 4*e + 7*f - 10*g)
*log(x + 1) + 1/108*(d + 4*e + 7*f + 10*g)*log(x - 1) - 1/864*(19*d + 32*e
+ 52*f + 80*g)*log(x - 2) - 1/72*((5*d + 8*f)*x^3 + 4*(2*e + 5*g)*x^2 - (17
*d + 20*f)*x - 20*e - 32*g)/(x^4 - 5*x^2 + 4)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 130, normalized size of antiderivative = 0.94

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g)\log\left(|x+2|\right)\\ &\quad-\frac{1}{108}\,(d-4\,e+7\,f-10\,g)\log\left(|x+1|\right)\\ &\quad+\frac{1}{108}\,(d+4\,e+7\,f+10\,g)\log\left(|x-1|\right)\\ &\quad-\frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g)\log\left(|x-2|\right)\\ &\quad-\frac{5\,dx^3+8\,fx^3+8\,ex^2+20\,gx^2-17\,dx-20\,fx-20\,e-32\,g}{72\,(x^4-5\,x^2+4)}\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/864*(19*d - 32*e + 52*f - 80*g)*log(abs(x + 2)) - 1/108*(d - 4*e + 7*f -
10*g)*log(abs(x + 1)) + 1/108*(d + 4*e + 7*f + 10*g)*log(abs(x - 1)) - 1/86
4*(19*d + 32*e + 52*f + 80*g)*log(abs(x - 2)) - 1/72*(5*d*x^3 + 8*f*x^3 + 8
*e*x^2 + 20*g*x^2 - 17*d*x - 20*f*x - 20*e - 32*g)/(x^4 - 5*x^2 + 4)
```

Mupad [B] (verification not implemented)

Time = 7.88 (sec) , antiderivative size = 128, normalized size of antiderivative = 0.93

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^2}\,dx &= \ln(x-1)\,\left(\frac{d}{108}+\frac{e}{27}+\frac{7\,f}{108}+\frac{5\,g}{54}\right)\\ &\quad-\ln(x+1)\,\left(\frac{d}{108}-\frac{e}{27}+\frac{7\,f}{108}-\frac{5\,g}{54}\right)\\ &\quad-\ln(x-2)\,\left(\frac{19\,d}{864}+\frac{e}{27}+\frac{13\,f}{216}+\frac{5\,g}{54}\right)\\ &\quad+\ln(x+2)\,\left(\frac{19\,d}{864}-\frac{e}{27}+\frac{13\,f}{216}-\frac{5\,g}{54}\right)\\ &\quad+\frac{\left(-\frac{5\,d}{72}-\frac{f}{9}\right)\,x^3+\left(-\frac{e}{9}-\frac{5\,g}{18}\right)\,x^2+\left(\frac{17\,d}{72}+\frac{5\,f}{18}\right)\,x+\frac{5\,e}{18}+\frac{4\,g}{9}}{x^4-5\,x^2+4}\end{aligned}$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/108 + e/27 + (7*f)/108 + (5*g)/54) - log(x + 1)*(d/108 - e/27
 + (7*f)/108 - (5*g)/54) - log(x - 2)*((19*d)/864 + e/27 + (13*f)/216 + (5*
g)/54) + log(x + 2)*((19*d)/864 - e/27 + (13*f)/216 - (5*g)/54) + ((5*e)/18
 + (4*g)/9 - x^3*((5*d)/72 + f/9) - x^2*(e/9 + (5*g)/18) + x*((17*d)/72 + (
5*f)/18))/(x^4 - 5*x^2 + 4)
```

3.29 $\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 33, antiderivative size = 150

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx &= \frac{5e+8g-(2e+5g)x^2}{18(4-5x^2+x^4)} \\ &\quad+\frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72(4-5x^2+x^4)} \\ &\quad+\frac{1}{432}(19d+52f+112h)\text{arctanh}\left(\frac{x}{2}\right) \\ &\quad-\frac{1}{54}(d+7f+13h)\text{arctanh}(x) \\ &\quad+\frac{1}{54}(2e+5g)\log\left(1-x^2\right)-\frac{1}{54}(2e+5g)\log\left(4-x^2\right)\end{aligned}$$

```
[Out] 1/18*(5*e+8*g-(2*e+5*g)*x^2)/(x^4-5*x^2+4)+1/72*x*(17*d+20*f+32*h-(5*d+8*f+
20*h)*x^2)/(x^4-5*x^2+4)+1/432*(19*d+52*f+112*h)*arctanh(1/2*x)-1/54*(d+7*f
+13*h)*arctanh(x)+1/54*(2*e+5*g)*ln(-x^2+1)-1/54*(2*e+5*g)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.13 (sec) , antiderivative size = 150, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.242, Rules used

$= \{1687, 1692, 1180, 213, 1261, 652, 630, 31\}$

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx = \frac{1}{432}\text{arctanh}\left(\frac{x}{2}\right)(19d+52f+112h) - \frac{1}{54}\text{arctanh}(x)(d+7f+13h) + \frac{x(-(x^2(5d+8f+20h))+17d+20f+32h)}{72\,(x^4-5x^2+4)} + \frac{1}{54}(2e+5g)\log\left(1-x^2\right) - \frac{1}{54}(2e+5g)\log\left(4-x^2\right) + \frac{-(x^2(2e+5g))+5e+8g}{18\,(x^4-5x^2+4)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (5*e + 8*g - (2*e + 5*g)*x^2)/(18*(4 - 5*x^2 + x^4)) + (x*(17*d + 20*f + 32
*h - (5*d + 8*f + 20*h)*x^2))/(72*(4 - 5*x^2 + x^4)) + ((19*d + 52*f + 112*
h)*ArcTanh[x/2])/432 - ((d + 7*f + 13*h)*ArcTanh[x])/54 + ((2*e + 5*g)*Log[
1 - x^2])/54 - ((2*e + 5*g)*Log[4 - x^2])/54
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{(4-5x^2+x^4)^2}\,dx + \int \frac{d+fx^2+hx^4}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} \\
&\quad - \frac{1}{72}\int \frac{-d+20f+32h+(5d+8f+20h)x^2}{4-5x^2+x^4}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(4-5x+x^2)^2}\,dx, x, x^2\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{5e+8g-(2e+5g)x^2}{18\left(4-5x^2+x^4\right)}+\frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\left(4-5x^2+x^4\right)} \\
&\quad + \frac{1}{18}(-2e-5g)\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx,x,x^2\right) \\
&\quad - \frac{1}{54}(-d-7f-13h)\int \frac{1}{-1+x^2}\,dx-\frac{1}{216}(19d+52f+112h)\int \frac{1}{-4+x^2}\,dx \\
&= \frac{5e+8g-(2e+5g)x^2}{18\left(4-5x^2+x^4\right)}+\frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\left(4-5x^2+x^4\right)} \\
&\quad + \frac{1}{432}(19d+52f+112h)\tanh^{-1}\left(\frac{x}{2}\right)-\frac{1}{54}(d+7f+13h)\tanh^{-1}(x) \\
&\quad + \frac{1}{54}(-2e-5g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx,x,x^2\right) \\
&\quad + \frac{1}{54}(2e+5g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx,x,x^2\right) \\
&= \frac{5e+8g-(2e+5g)x^2}{18\left(4-5x^2+x^4\right)}+\frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\left(4-5x^2+x^4\right)} \\
&\quad + \frac{1}{432}(19d+52f+112h)\tanh^{-1}\left(\frac{x}{2}\right)-\frac{1}{54}(d+7f+13h)\tanh^{-1}(x) \\
&\quad + \frac{1}{54}(2e+5g)\log\left(1-x^2\right)-\frac{1}{54}(2e+5g)\log\left(4-x^2\right)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 159, normalized size of antiderivative = 1.06

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4}{\left(4-5x^2+x^4\right)^2}\,dx \\
&= \frac{1}{864}\left(-\frac{12(4e(-5+2x^2)+4g(-8+5x^2)+x(4f(-5+2x^2)+d(-17+5x^2)+4h(-8+5x^2)))}{4-5x^2+x^4}\right. \\
&\quad + 8(d+4e+7f+10g+13h)\log(1-x)-(19d+32e+52f+80g+112h)\log(2-x) \\
&\quad \left. - 8(d-4e+7f-10g+13h)\log(1+x)+(19d-32e+52f-80g+112h)\log(2+x)\right)
\end{aligned}
$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((-12*(4*e*(-5 + 2*x^2) + 4*g*(-8 + 5*x^2) + x*(4*f*(-5 + 2*x^2) + d*(-17 +
 5*x^2) + 4*h*(-8 + 5*x^2))))/(4 - 5*x^2 + x^4) + 8*(d + 4*e + 7*f + 10*g +
 13*h)*Log[1 - x] - (19*d + 32*e + 52*f + 80*g + 112*h)*Log[2 - x] - 8*(d -
 4*e + 7*f - 10*g + 13*h)*Log[1 + x] + (19*d - 32*e + 52*f - 80*g + 112*h)*
Log[2 + x])/864
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 143, normalized size of antiderivative = 0.95

method	result
norman	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}-\frac{5h}{18}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}+\frac{4h}{9}\right)x+\left(-\frac{e}{9}-\frac{5g}{18}\right)x^2+\frac{5e}{18}+\frac{4g}{9}}{x^4-5x^2+4}+\left(-\frac{19d}{864}-\frac{e}{27}-\frac{13f}{216}-\frac{5g}{54}-\frac{7h}{54}\right)\ln\left(x-2\right)+($
default	$-\frac{\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}}{x+2}+\left(\frac{19d}{864}-\frac{e}{27}+\frac{13f}{216}-\frac{5g}{54}+\frac{7h}{54}\right)\ln\left(x+2\right)+\left(-\frac{d}{108}+\frac{e}{27}-\frac{7f}{108}+\frac{5g}{54}-\frac{13h}{108}\right)\ln$
risch	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}-\frac{5h}{18}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}+\frac{4h}{9}\right)x+\left(-\frac{e}{9}-\frac{5g}{18}\right)x^2+\frac{5e}{18}+\frac{4g}{9}}{x^4-5x^2+4}-\frac{\ln(x+1)d}{108}+\frac{\ln(x+1)e}{27}-\frac{7\ln(x+1)f}{108}+\frac{5\ln(x+1)g}{54}-$
parallelrisch	$-\frac{-384g-240e+96f\,x^3+240g\,x^2+240h\,x^3-204dx+76\ln(x-2)d+128\ln(x-2)e-32\ln(x-1)d-128\ln(x-1)e+32\ln(x-2)x^4e}{}$

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] ((-5/72*d-1/9*f-5/18*h)*x^3+(17/72*d+5/18*f+4/9*h)*x+(-1/9*e-5/18*g)*x^2+5/
18*e+4/9*g)/(x^4-5*x^2+4)+(-19/864*d-1/27*e-13/216*f-5/54*g-7/54*h)*ln(x-2)
+(-1/108*d+1/27*e-7/108*f+5/54*g-13/108*h)*ln(x+1)+(1/108*d+1/27*e+7/108*f+
5/54*g+13/108*h)*ln(x-1)+(19/864*d-1/27*e+13/216*f-5/54*g+7/54*h)*ln(x+2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 304 vs. 2(134) = 268.

Time = 1.40 (sec) , antiderivative size = 304, normalized size of antiderivative = 2.03

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx =$$
$$-\frac{12\,(5\,d+8\,f+20\,h)x^3+48\,(2\,e+5\,g)x^2-12\,(17\,d+20\,f+32\,h)x-((19\,d-32\,e+52\,f-80\,g+11}{}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/864*(12*(5*d + 8*f + 20*h)*x^3 + 48*(2*e + 5*g)*x^2 - 12*(17*d + 20*f +
32*h)*x - ((19*d - 32*e + 52*f - 80*g + 112*h)*x^4 - 5*(19*d - 32*e + 52*f
- 80*g + 112*h)*x^2 + 76*d - 128*e + 208*f - 320*g + 448*h)*log(x + 2) + 8*
((d - 4*e + 7*f - 10*g + 13*h)*x^4 - 5*(d - 4*e + 7*f - 10*g + 13*h)*x^2 +
4*d - 16*e + 28*f - 40*g + 52*h)*log(x + 1) - 8*((d + 4*e + 7*f + 10*g + 13
*h)*x^4 - 5*(d + 4*e + 7*f + 10*g + 13*h)*x^2 + 4*d + 16*e + 28*f + 40*g +
52*h)*log(x - 1) + ((19*d + 32*e + 52*f + 80*g + 112*h)*x^4 - 5*(19*d + 32*
e + 52*f + 80*g + 112*h)*x^2 + 76*d + 128*e + 208*f + 320*g + 448*h)*log(x
- 2) - 240*e - 384*g)/(x^4 - 5*x^2 + 4)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 145, normalized size of antiderivative = 0.97

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g+112\,h)\log(x+2) \\
&\quad - \frac{1}{108}\,(d-4\,e+7\,f-10\,g+13\,h)\log(x+1) \\
&\quad + \frac{1}{108}\,(d+4\,e+7\,f+10\,g+13\,h)\log(x-1) \\
&\quad - \frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g+112\,h)\log(x-2) \\
&\quad - \frac{(5\,d+8\,f+20\,h)x^3+4\,(2\,e+5\,g)x^2-(17\,d+20\,f+32\,h)x-20\,e-32\,g}{72\,(x^4-5\,x^2+4)}
\end{aligned}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/864*(19*d - 32*e + 52*f - 80*g + 112*h)*log(x + 2) - 1/108*(d - 4*e + 7*f
 - 10*g + 13*h)*log(x + 1) + 1/108*(d + 4*e + 7*f + 10*g + 13*h)*log(x - 1)
 - 1/864*(19*d + 32*e + 52*f + 80*g + 112*h)*log(x - 2) - 1/72*((5*d + 8*f
+ 20*h)*x^3 + 4*(2*e + 5*g)*x^2 - (17*d + 20*f + 32*h)*x - 20*e - 32*g)/(x^
4 - 5*x^2 + 4)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 152, normalized size of antiderivative = 1.01

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g+112\,h)\log\left(|x+2|\right) - \frac{1}{108}\,(d-4\,e+7\,f-10\,g+13\,h)\log\left(|x+1|\right) + \frac{1}{108}\,(d+4\,e+7\,f+10\,g+13\,h)\log\left(|x-1|\right) - \frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g+112\,h)\log\left(|x-2|\right) - \frac{5\,dx^3+8\,fx^3+20\,hx^3+8\,ex^2+20\,gx^2-17\,dx-20\,fx-32\,hx-20\,e-32\,g}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
[Out] 1/864*(19*d - 32*e + 52*f - 80*g + 112*h)*log(abs(x + 2)) - 1/108*(d - 4*e
+ 7*f - 10*g + 13*h)*log(abs(x + 1)) + 1/108*(d + 4*e + 7*f + 10*g + 13*h)*
log(abs(x - 1)) - 1/864*(19*d + 32*e + 52*f + 80*g + 112*h)*log(abs(x - 2))
 - 1/72*(5*d*x^3 + 8*f*x^3 + 20*h*x^3 + 8*e*x^2 + 20*g*x^2 - 17*d*x - 20*f*
x - 32*h*x - 20*e - 32*g)/(x^4 - 5*x^2 + 4)
```

Mupad [B] (verification not implemented)

Time = 7.98 (sec) , antiderivative size = 146, normalized size of antiderivative = 0.97

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{\left(-\frac{5\,d}{72}-\frac{f}{9}-\frac{5\,h}{18}\right)\,x^3+\left(-\frac{e}{9}-\frac{5\,g}{18}\right)\,x^2+\left(\frac{17\,d}{72}+\frac{5\,f}{18}+\frac{4\,h}{9}\right)\,x+\frac{5\,e}{18}+\frac{4\,g}{9}}{x^4-5\,x^2+4} + \ln(x-1)\,\left(\frac{d}{108}+\frac{e}{27}+\frac{7\,f}{108}+\frac{5\,g}{54}+\frac{13\,h}{108}\right) - \ln(x+1)\,\left(\frac{d}{108}-\frac{e}{27}+\frac{7\,f}{108}-\frac{5\,g}{54}+\frac{13\,h}{108}\right) - \ln(x-2)\,\left(\frac{19\,d}{864}+\frac{e}{27}+\frac{13\,f}{216}+\frac{5\,g}{54}+\frac{7\,h}{54}\right) + \ln(x+2)\,\left(\frac{19\,d}{864}-\frac{e}{27}+\frac{13\,f}{216}-\frac{5\,g}{54}+\frac{7\,h}{54}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^4 - 5*x^2 + 4)^2,x)
[Out] ((5*e)/18 + (4*g)/9 - x^2*(e/9 + (5*g)/18) + x*((17*d)/72 + (5*f)/18 + (4*h
)/9) - x^3*((5*d)/72 + f/9 + (5*h)/18))/(x^4 - 5*x^2 + 4) + log(x - 1)*(d/1
08 + e/27 + (7*f)/108 + (5*g)/54 + (13*h)/108) - log(x + 1)*(d/108 - e/27 +
 (7*f)/108 - (5*g)/54 + (13*h)/108) - log(x - 2)*((19*d)/864 + e/27 + (13*f
)/216 + (5*g)/54 + (7*h)/54) + log(x + 2)*((19*d)/864 - e/27 + (13*f)/216 -
 (5*g)/54 + (7*h)/54)
```

3.30 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 38, antiderivative size = 162

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx &= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} \\ &+ \frac{5e+8g+20i-(2e+5g+17i)x^2}{18\,(4-5x^2+x^4)} \\ &+ \frac{1}{432}(19d+52f+112h)\text{arctanh}\left(\frac{x}{2}\right) \\ &- \frac{1}{54}(d+7f+13h)\text{arctanh}(x) \\ &+ \frac{1}{54}(2e+5g+8i)\log\left(1-x^2\right) \\ &- \frac{1}{54}(2e+5g+8i)\log\left(4-x^2\right)\end{aligned}$$

```
[Out] 1/72*x*(17*d+20*f+32*h-(5*d+8*f+20*h)*x^2)/(x^4-5*x^2+4)+1/18*(5*e+8*g+20*i
-(2*e+5*g+17*i)*x^2)/(x^4-5*x^2+4)+1/432*(19*d+52*f+112*h)*arctanh(1/2*x)-1
/54*(d+7*f+13*h)*arctanh(x)+1/54*(2*e+5*g+8*i)*ln(-x^2+1)-1/54*(2*e+5*g+8*i
)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.15 (sec) , antiderivative size = 162, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.237, Rules used = {1687, 1692, 1180, 213, 1677, 1674, 12, 630, 31}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx = \frac{1}{432}\text{arctanh}\left(\frac{x}{2}\right)(19d+52f+112h) - \frac{1}{54}\text{arctanh}(x)(d+7f+13h) + \frac{x(-(x^2(5d+8f+20h))+17d+20f+32h)}{72(x^4-5x^2+4)} + \frac{1}{54}\log\left(1-x^2\right)(2e+5g+8i) - \frac{1}{54}\log\left(4-x^2\right)(2e+5g+8i) + \frac{-(x^2(2e+5g+17i))+5e+8g+20i}{18(x^4-5x^2+4)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (x*(17*d + 20*f + 32*h - (5*d + 8*f + 20*h)*x^2))/(72*(4 - 5*x^2 + x^4)) +
(5*e + 8*g + 20*i - (2*e + 5*g + 17*i)*x^2)/(18*(4 - 5*x^2 + x^4)) + ((19*d
 + 52*f + 112*h)*ArcTanh[x/2])/432 - ((d + 7*f + 13*h)*ArcTanh[x])/54 + ((2
*e + 5*g + 8*i)*Log[1 - x^2])/54 - ((2*e + 5*g + 8*i)*Log[4 - x^2])/54
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
```

```
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
```

```
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2+hx^4}{(4-5x^2+x^4)^2}\,dx + \int \frac{x(e+gx^2+ix^4)}{(4-5x^2+x^4)^2}\,dx$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} - \frac{1}{72}\int \frac{-d+20f+32h+(5d+8f+20h)x^2}{4-5x^2+x^4}\,dx + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{(4-5x+x^2)^2}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{18\,(4-5x^2+x^4)} - \frac{1}{18}\text{Subst}\left(\int \frac{2e+5g+8i}{4-5x+x^2}\,dx, x, x^2\right) - \frac{1}{54}(-d-7f-13h)\int \frac{1}{-1+x^2}\,dx - \frac{1}{216}(19d+52f+112h)\int \frac{1}{-4+x^2}\,dx$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{18\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f+112h)\tanh^{-1}\left(\frac{x}{2}\right) - \frac{1}{54}(d+7f+13h)\tanh^{-1}(x) - \frac{1}{18}(2e+5g+8i)\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{18\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f+112h)\tanh^{-1}\left(\frac{x}{2}\right) - \frac{1}{54}(d+7f+13h)\tanh^{-1}(x) - \frac{1}{54}(-2e-5g-8i)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) - \frac{1}{54}(2e+5g+8i)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{72\,(4-5x^2+x^4)} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{18\,(4-5x^2+x^4)} + \frac{1}{432}(19d+52f+112h)\tanh^{-1}\left(\frac{x}{2}\right) - \frac{1}{54}(d+7f+13h)\tanh^{-1}(x) + \frac{1}{54}(2e+5g+8i)\log\left(1-x^2\right) - \frac{1}{54}(2e+5g+8i)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 185, normalized size of antiderivative = 1.14

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx$$
$$=\frac{20e+32g+80i+17dx+20fx+32hx-8ex^2-20gx^2-68ix^2-5dx^3-8fx^3-20hx^3}{72(4-5x^2+x^4)}$$
$$+\frac{1}{108}(d+4e+7f+10g+13h+16i)\log(1-x)$$
$$+\frac{1}{864}(-19d-32e-52f-80g-112h-128i)\log(2-x)$$
$$+\frac{1}{108}(-d+4e-7f+10g-13h+16i)\log(1+x)$$
$$+\frac{1}{864}(19d-32e+52f-80g+112h-128i)\log(2+x)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (20*e + 32*g + 80*i + 17*d*x + 20*f*x + 32*h*x - 8*e*x^2 - 20*g*x^2 - 68*i*
x^2 - 5*d*x^3 - 8*f*x^3 - 20*h*x^3)/(72*(4 - 5*x^2 + x^4)) + ((d + 4*e + 7*
f + 10*g + 13*h + 16*i)*Log[1 - x])/108 + ((-19*d - 32*e - 52*f - 80*g - 11
2*h - 128*i)*Log[2 - x])/864 + ((-d + 4*e - 7*f + 10*g - 13*h + 16*i)*Log[1
 + x])/108 + ((19*d - 32*e + 52*f - 80*g + 112*h - 128*i)*Log[2 + x])/864
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 161, normalized size of antiderivative = 0.99

method	result
norman	$\frac{\left(-\frac{5d}{72}-\frac{f}{9}-\frac{5h}{18}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}+\frac{4h}{9}\right)x+\left(-\frac{5g}{18}-\frac{e}{9}-\frac{17i}{18}\right)x^2+\frac{4g}{9}+\frac{5e}{18}+\frac{10i}{9}}{x^4-5x^2+4}+\left(-\frac{19d}{864}-\frac{e}{27}-\frac{13f}{216}-\frac{5g}{54}-\frac{7h}{54}-\frac{4i}{27}\right)\ln$
default	$-\frac{\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}-\frac{2i}{9}}{x+2}+\left(\frac{19d}{864}-\frac{e}{27}+\frac{13f}{216}-\frac{5g}{54}+\frac{7h}{54}-\frac{4i}{27}\right)\ln(x+2)+\left(-\frac{d}{108}+\frac{e}{27}-\frac{7f}{108}+\frac{5g}{54}-\right.$
risch	$\frac{4\ln(x+1)i}{27}+\frac{\left(-\frac{5d}{72}-\frac{f}{9}-\frac{5h}{18}\right)x^3+\left(\frac{17d}{72}+\frac{5f}{18}+\frac{4h}{9}\right)x+\left(-\frac{5g}{18}-\frac{e}{9}-\frac{17i}{18}\right)x^2+\frac{4g}{9}+\frac{5e}{18}+\frac{10i}{9}}{x^4-5x^2+4}+\frac{13\ln(x+2)f}{216}+\frac{7\ln(1-x)f}{108}-\frac{7\ln}{}$
parallelrisch	$-\frac{-960i-384g-240e+96fx^3+240gx^2+240hx^3-204dx+816ix^2+76\ln(x-2)d+128\ln(x-2)e-32\ln(x-1)d-128\ln(x-1)e-51}{}$

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE
)
```

```
[Out] ((-5/72*d-1/9*f-5/18*h)*x^3+(17/72*d+5/18*f+4/9*h)*x+(-5/18*g-1/9*e-17/18*i
)*x^2+4/9*g+5/18*e+10/9*i)/(x^4-5*x^2+4)+(-19/864*d-1/27*e-13/216*f-5/54*g-
7/54*h-4/27*i)*ln(x-2)+(-1/108*d+1/27*e-7/108*f+5/54*g-13/108*h+4/27*i)*ln(
```

```
x+1)+(1/108*d+1/27*e+7/108*f+5/54*g+13/108*h+4/27*i)*ln(x-1)+(19/864*d-1/27
*e+13/216*f-5/54*g+7/54*h-4/27*i)*ln(x+2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 346 vs. $2(146) = 292$.

Time = 6.24 (sec) , antiderivative size = 346, normalized size of antiderivative = 2.14

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + ix^5}{(4 - 5x^2 + x^4)^2} dx = -\frac{12 (5 d + 8 f + 20 h)x^3 + 48 (2 e + 5 g + 17 i)x^2 - 12 (17 d + 20 f + 32 h)x - ((19 d - 32 e + 52 f - 80}{}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fri
cas")
```

```
[Out] -1/864*(12*(5*d + 8*f + 20*h)*x^3 + 48*(2*e + 5*g + 17*i)*x^2 - 12*(17*d +
20*f + 32*h)*x - ((19*d - 32*e + 52*f - 80*g + 112*h - 128*i)*x^4 - 5*(19*d
 - 32*e + 52*f - 80*g + 112*h - 128*i)*x^2 + 76*d - 128*e + 208*f - 320*g +
 448*h - 512*i)*log(x + 2) + 8*((d - 4*e + 7*f - 10*g + 13*h - 16*i)*x^4 -
5*(d - 4*e + 7*f - 10*g + 13*h - 16*i)*x^2 + 4*d - 16*e + 28*f - 40*g + 52*
h - 64*i)*log(x + 1) - 8*((d + 4*e + 7*f + 10*g + 13*h + 16*i)*x^4 - 5*(d +
 4*e + 7*f + 10*g + 13*h + 16*i)*x^2 + 4*d + 16*e + 28*f + 40*g + 52*h + 64
*i)*log(x - 1) + ((19*d + 32*e + 52*f + 80*g + 112*h + 128*i)*x^4 - 5*(19*d
 + 32*e + 52*f + 80*g + 112*h + 128*i)*x^2 + 76*d + 128*e + 208*f + 320*g +
 448*h + 512*i)*log(x - 2) - 240*e - 384*g - 960*i)/(x^4 - 5*x^2 + 4)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + ix^5}{(4 - 5x^2 + x^4)^2} dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 163, normalized size of antiderivative = 1.01

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g+112\,h-128\,i)\log{(x+2)} - \frac{1}{108}\,(d-4\,e+7\,f-10\,g+13\,h-16\,i)\log{(x+1)} + \frac{1}{108}\,(d+4\,e+7\,f+10\,g+13\,h+16\,i)\log{(x-1)} - \frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g+112\,h+128\,i)\log{(x-2)} - \frac{(5\,d+8\,f+20\,h)x^3+4\,(2\,e+5\,g+17\,i)x^2-(17\,d+20\,f+32\,h)x-20\,e-32\,g-80\,i}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="max
ima")
```

```
[Out] 1/864*(19*d - 32*e + 52*f - 80*g + 112*h - 128*i)*log(x + 2) - 1/108*(d - 4
*e + 7*f - 10*g + 13*h - 16*i)*log(x + 1) + 1/108*(d + 4*e + 7*f + 10*g + 1
3*h + 16*i)*log(x - 1) - 1/864*(19*d + 32*e + 52*f + 80*g + 112*h + 128*i)*
log(x - 2) - 1/72*((5*d + 8*f + 20*h)*x^3 + 4*(2*e + 5*g + 17*i)*x^2 - (17*
d + 20*f + 32*h)*x - 20*e - 32*g - 80*i)/(x^4 - 5*x^2 + 4)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 173, normalized size of antiderivative = 1.07

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{1}{864}\,(19\,d-32\,e+52\,f-80\,g+112\,h-128\,i)\log{(|x+2|)} - \frac{1}{108}\,(d-4\,e+7\,f-10\,g+13\,h-16\,i)\log{(|x+1|)} + \frac{1}{108}\,(d+4\,e+7\,f+10\,g+13\,h+16\,i)\log{(|x-1|)} - \frac{1}{864}\,(19\,d+32\,e+52\,f+80\,g+112\,h+128\,i)\log{(|x-2|)} - \frac{5\,dx^3+8\,fx^3+20\,hx^3+8\,ex^2+20\,gx^2+68\,ix^2-17\,dx-20\,fx-32\,hx-20\,e-32\,g-80\,i}{72\,(x^4-5\,x^2+4)}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="gia
c")
```

```
[Out] 1/864*(19*d - 32*e + 52*f - 80*g + 112*h - 128*i)*log(abs(x + 2)) - 1/108*(
d - 4*e + 7*f - 10*g + 13*h - 16*i)*log(abs(x + 1)) + 1/108*(d + 4*e + 7*f
+ 10*g + 13*h + 16*i)*log(abs(x - 1)) - 1/864*(19*d + 32*e + 52*f + 80*g +
112*h + 128*i)*log(abs(x - 2)) - 1/72*(5*d*x^3 + 8*f*x^3 + 20*h*x^3 + 8*e*x
^2 + 20*g*x^2 + 68*i*x^2 - 17*d*x - 20*f*x - 32*h*x - 20*e - 32*g - 80*i)/(
x^4 - 5*x^2 + 4)
```

Mupad [B] (verification not implemented)

Time = 0.35 (sec) , antiderivative size = 164, normalized size of antiderivative = 1.01

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{\left(-\frac{5\,d}{72}-\frac{f}{9}-\frac{5\,h}{18}\right)\,x^3+\left(-\frac{e}{9}-\frac{5\,g}{18}-\frac{17\,i}{18}\right)\,x^2+\left(\frac{17\,d}{72}+\frac{5\,f}{18}+\frac{4\,h}{9}\right)\,x+\frac{5\,e}{18}+\frac{4\,g}{9}+\frac{10\,i}{9}}{x^4-5\,x^2+4} \\
&\quad+\ln\left(x-1\right)\,\left(\frac{d}{108}+\frac{e}{27}+\frac{7\,f}{108}+\frac{5\,g}{54}+\frac{13\,h}{108}+\frac{4\,i}{27}\right) \\
&\quad-\ln\left(x+1\right)\,\left(\frac{d}{108}-\frac{e}{27}+\frac{7\,f}{108}-\frac{5\,g}{54}+\frac{13\,h}{108}-\frac{4\,i}{27}\right) \\
&\quad-\ln\left(x-2\right)\,\left(\frac{19\,d}{864}+\frac{e}{27}+\frac{13\,f}{216}+\frac{5\,g}{54}+\frac{7\,h}{54}+\frac{4\,i}{27}\right) \\
&\quad+\ln\left(x+2\right)\,\left(\frac{19\,d}{864}-\frac{e}{27}+\frac{13\,f}{216}-\frac{5\,g}{54}+\frac{7\,h}{54}-\frac{4\,i}{27}\right)
\end{aligned}
$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] ((5*e)/18 + (4*g)/9 + (10*i)/9 + x*((17*d)/72 + (5*f)/18 + (4*h)/9) - x^3*(
(5*d)/72 + f/9 + (5*h)/18) - x^2*(e/9 + (5*g)/18 + (17*i)/18))/(x^4 - 5*x^2
 + 4) + log(x - 1)*(d/108 + e/27 + (7*f)/108 + (5*g)/54 + (13*h)/108 + (4*i
)/27) - log(x + 1)*(d/108 - e/27 + (7*f)/108 - (5*g)/54 + (13*h)/108 - (4*i
)/27) - log(x - 2)*((19*d)/864 + e/27 + (13*f)/216 + (5*g)/54 + (7*h)/54 +
(4*i)/27) + log(x + 2)*((19*d)/864 - e/27 + (13*f)/216 - (5*g)/54 + (7*h)/5
4 - (4*i)/27)
```

3.31 $\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 16, antiderivative size = 140

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \frac{dx(1-x^2)}{6(1+x^2+x^4)}+\frac{e(1+2x^2)}{6(1+x^2+x^4)}-\frac{d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{3\sqrt{3}}+\frac{d\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{3\sqrt{3}}+\frac{2e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{4}d\log\left(1-x+x^2\right)+\frac{1}{4}d\log\left(1+x+x^2\right)$$

```
[Out] 1/6*d*x*(-x^2+1)/(x^4+x^2+1)+1/6*e*(2*x^2+1)/(x^4+x^2+1)-1/4*d*ln(x^2-x+1)+
1/4*d*ln(x^2+x+1)-1/9*d*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/9*d*arctan(1/
3*(1+2*x)*3^(1/2))*3^(1/2)+2/9*e*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 140, normalized size of antiderivative = 1.00, number of steps used = 17, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.625, Rules used = {1687, 12, 1106, 1183, 648, 632, 210, 642, 1121, 628}

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = -\frac{d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{3\sqrt{3}}+\frac{d\arctan\left(\frac{2x+1}{\sqrt{3}}\right)}{3\sqrt{3}}+\frac{2e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{4}d\log\left(x^2-x+1\right)+\frac{1}{4}d\log\left(x^2+x+1\right)+\frac{dx(1-x^2)}{6(x^4+x^2+1)}+\frac{e(2x^2+1)}{6(x^4+x^2+1)}$$

```
[In] Int[(d + e*x)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (d*x*(1 - x^2))/(6*(1 + x^2 + x^4)) + (e*(1 + 2*x^2))/(6*(1 + x^2 + x^4)) -
 (d*ArcTan[(1 - 2*x)/Sqrt[3]])/(3*Sqrt[3]) + (d*ArcTan[(1 + 2*x)/Sqrt[3]])/
(3*Sqrt[3]) + (2*e*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - (d*Log[1 - x
+ x^2])/4 + (d*Log[1 + x + x^2])/4
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
```

```
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d}{(1+x^2+x^4)^2}\,dx + \int \frac{ex}{(1+x^2+x^4)^2}\,dx \\
&= d\int \frac{1}{(1+x^2+x^4)^2}\,dx + e\int \frac{x}{(1+x^2+x^4)^2}\,dx \\
&= \frac{dx(1-x^2)}{6\,(1+x^2+x^4)} + \frac{1}{6}d\int \frac{5-x^2}{1+x^2+x^4}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{dx(1-x^2)}{6\,(1+x^2+x^4)} + \frac{e(1+2x^2)}{6\,(1+x^2+x^4)} + \frac{1}{12}d\int \frac{5-6x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{12}d\int \frac{5+6x}{1+x+x^2}\,dx + \frac{1}{3}e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&= \frac{dx(1-x^2)}{6\,(1+x^2+x^4)} + \frac{e(1+2x^2)}{6\,(1+x^2+x^4)} + \frac{1}{6}d\int \frac{1}{1-x+x^2}\,dx + \frac{1}{6}d\int \frac{1}{1+x+x^2}\,dx \\
&\quad - \frac{1}{4}d\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{4}d\int \frac{1+2x}{1+x+x^2}\,dx - \frac{1}{3}(2e)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1\right. \\
&\quad \left. + 2x^2\right)
\end{aligned}
$$

$$= \frac{dx(1-x^2)}{6(1+x^2+x^4)} + \frac{e(1+2x^2)}{6(1+x^2+x^4)} + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{4}d\log\left(1-x+x^2\right) + \frac{1}{4}d\log\left(1+x+x^2\right) - \frac{1}{3}d\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) - \frac{1}{3}d\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)$$

$$= \frac{dx(1-x^2)}{6(1+x^2+x^4)} + \frac{e(1+2x^2)}{6(1+x^2+x^4)} - \frac{d\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{3\sqrt{3}} + \frac{d\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{3\sqrt{3}} + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{4}d\log\left(1-x+x^2\right) + \frac{1}{4}d\log\left(1+x+x^2\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.35 (sec) , antiderivative size = 146, normalized size of antiderivative = 1.04

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \frac{e+dx+2ex^2-dx^3}{6(1+x^2+x^4)} - \frac{\left(-11i+\sqrt{3}\right)d\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{6\sqrt{6+6i\sqrt{3}}} - \frac{\left(11i+\sqrt{3}\right)d\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{6\sqrt{6-6i\sqrt{3}}} - \frac{2e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)}{3\sqrt{3}}$$

```
[In] Integrate[(d + e*x)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (e + d*x + 2*e*x^2 - d*x^3)/(6*(1 + x^2 + x^4)) - ((-11*I + Sqrt[3])*d*ArcT
an[((-I + Sqrt[3])*x)/2])/(6*Sqrt[6 + (6*I)*Sqrt[3]]) - ((11*I + Sqrt[3])*d
*ArcTan[((I + Sqrt[3])*x)/2])/(6*Sqrt[6 - (6*I)*Sqrt[3]]) - (2*e*ArcTan[Sqr
t[3]/(1 + 2*x^2)])/(3*Sqrt[3])
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 124, normalized size of antiderivative = 0.89

method	result
default	$-\frac{\left(\frac{d}{3}-\frac{e}{3}\right)x-\frac{2d}{3}-\frac{e}{3}}{4(x^2-x+1)} - \frac{d\ln(x^2-x+1)}{4} - \frac{(-2d-4e)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{18} + \frac{\left(-\frac{d}{3}-\frac{e}{3}\right)x-\frac{2d}{3}+\frac{e}{3}}{4x^2+4x+4} + \frac{d\ln(x^2+x+1)}{4} + \frac{(2d-4e}{}$
risch	$\frac{d\ln(10044d^2x^2+5952e^2x^2+10044d^2x+5952e^2x+10044d^2+5952e^2)}{4} + \frac{\sqrt{3}\,d\arctan\left(\frac{18d^2x\sqrt{3}}{27d^2+16e^2}+\frac{32e^2x\sqrt{3}}{3(27d^2+16e^2)}+\frac{9\sqrt{3}\,d^2}{27d^2+16e^2}+\frac{1}{3(27}\right.}{9}$

```
[In] int((e*x+d)/(x^4+x^2+1)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*((1/3*d-1/3*e)*x-2/3*d-1/3*e)/(x^2-x+1)-1/4*d*ln(x^2-x+1)-1/18*(-2*d-4
*e)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*((-1/3*d-1/3*e)*x-2/3*d+1/3*e)/
(x^2+x+1)+1/4*d*ln(x^2+x+1)+1/18*(2*d-4*e)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1
/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.10

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = -\frac{6\,dx^3 - 12\,ex^2 - 4\,\sqrt{3}((d-2\,e)x^4 + (d-2\,e)x^2 + d - 2\,e)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right) - 4\,\sqrt{3}((d+2\,e)x^4 +$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^2,x, algorithm="fricas")
```

```
[Out] -1/36*(6*d*x^3 - 12*e*x^2 - 4*sqrt(3)*((d - 2*e)*x^4 + (d - 2*e)*x^2 + d -
2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) - 4*sqrt(3)*((d + 2*e)*x^4 + (d + 2*e)*x
^2 + d + 2*e)*arctan(1/3*sqrt(3)*(2*x - 1)) - 6*d*x - 9*(d*x^4 + d*x^2 + d)
*log(x^2 + x + 1) + 9*(d*x^4 + d*x^2 + d)*log(x^2 - x + 1) - 6*e)/(x^4 + x^
2 + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.01 (sec) , antiderivative size = 952, normalized size of antiderivative = 6.80

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(x**4+x**2+1)**2,x)
```

```
[Out] (-d/4 - sqrt(3)*I*(d + 2*e)/18)*log(x + (-10309*d**4*e + 1026*d**4*(-d/4 -
sqrt(3)*I*(d + 2*e)/18) - 7200*d**2*e**3 - 31536*d**2*e**2*(-d/4 - sqrt(3)*
I*(d + 2*e)/18) + 108432*d**2*e*(-d/4 - sqrt(3)*I*(d + 2*e)/18)**2 + 163296
*d**2*(-d/4 - sqrt(3)*I*(d + 2*e)/18)**3 + 1792*e**5 + 11520*e**4*(-d/4 - s
qrt(3)*I*(d + 2*e)/18) + 48384*e**3*(-d/4 - sqrt(3)*I*(d + 2*e)/18)**2 + 31
1040*e**2*(-d/4 - sqrt(3)*I*(d + 2*e)/18)**3)/(3348*d**5 - 11408*d**3*e**2
- 7936*d*e**4)) + (-d/4 + sqrt(3)*I*(d + 2*e)/18)*log(x + (-10309*d**4*e +
1026*d**4*(-d/4 + sqrt(3)*I*(d + 2*e)/18) - 7200*d**2*e**3 - 31536*d**2*e**
2*(-d/4 + sqrt(3)*I*(d + 2*e)/18) + 108432*d**2*e*(-d/4 + sqrt(3)*I*(d + 2*
e)/18)**2 + 163296*d**2*(-d/4 + sqrt(3)*I*(d + 2*e)/18)**3 + 1792*e**5 + 11
520*e**4*(-d/4 + sqrt(3)*I*(d + 2*e)/18) + 48384*e**3*(-d/4 + sqrt(3)*I*(d
```

```
+ 2*e)/18)**2 + 311040*e**2*(-d/4 + sqrt(3)*I*(d + 2*e)/18)**3)/(3348*d**5
- 11408*d**3*e**2 - 7936*d*e**4)) + (d/4 - sqrt(3)*I*(d - 2*e)/18)*log(x +
(-10309*d**4*e + 1026*d**4*(d/4 - sqrt(3)*I*(d - 2*e)/18) - 7200*d**2*e**3
- 31536*d**2*e**2*(d/4 - sqrt(3)*I*(d - 2*e)/18) + 108432*d**2*e*(d/4 - sqr
t(3)*I*(d - 2*e)/18)**2 + 163296*d**2*(d/4 - sqrt(3)*I*(d - 2*e)/18)**3 + 1
792*e**5 + 11520*e**4*(d/4 - sqrt(3)*I*(d - 2*e)/18) + 48384*e**3*(d/4 - sq
rt(3)*I*(d - 2*e)/18)**2 + 311040*e**2*(d/4 - sqrt(3)*I*(d - 2*e)/18)**3)/(
3348*d**5 - 11408*d**3*e**2 - 7936*d*e**4)) + (d/4 + sqrt(3)*I*(d - 2*e)/18
)*log(x + (-10309*d**4*e + 1026*d**4*(d/4 + sqrt(3)*I*(d - 2*e)/18) - 7200*
d**2*e**3 - 31536*d**2*e**2*(d/4 + sqrt(3)*I*(d - 2*e)/18) + 108432*d**2*e*
(d/4 + sqrt(3)*I*(d - 2*e)/18)**2 + 163296*d**2*(d/4 + sqrt(3)*I*(d - 2*e)/
18)**3 + 1792*e**5 + 11520*e**4*(d/4 + sqrt(3)*I*(d - 2*e)/18) + 48384*e**3
*(d/4 + sqrt(3)*I*(d - 2*e)/18)**2 + 311040*e**2*(d/4 + sqrt(3)*I*(d - 2*e)
/18)**3)/(3348*d**5 - 11408*d**3*e**2 - 7936*d*e**4)) + (-d*x**3 + d*x + 2*
e*x**2 + e)/(6*x**4 + 6*x**2 + 6)
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 96, normalized size of antiderivative = 0.69

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \frac{1}{9}\sqrt{3}(d-2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{9}\sqrt{3}(d+2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{4}d\log\left(x^2+x+1\right) - \frac{1}{4}d\log\left(x^2-x+1\right) - \frac{dx^3-2ex^2-dx-e}{6(x^4+x^2+1)}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^2,x, algorithm="maxima")
[Out] 1/9*sqrt(3)*(d - 2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/9*sqrt(3)*(d + 2*e)
*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*d*log(x^2 + x + 1) - 1/4*d*log(x^2 - x
 + 1) - 1/6*(d*x^3 - 2*e*x^2 - d*x - e)/(x^4 + x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 96, normalized size of antiderivative = 0.69

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \frac{1}{9}\sqrt{3}(d-2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{9}\sqrt{3}(d+2e)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{4}d\log\left(x^2+x+1\right) - \frac{1}{4}d\log\left(x^2-x+1\right) - \frac{dx^3-2ex^2-dx-e}{6(x^4+x^2+1)}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^2,x, algorithm="giac")
```

```
[Out] 1/9*sqrt(3)*(d - 2*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/9*sqrt(3)*(d + 2*e)
*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/4*d*log(x^2 + x + 1) - 1/4*d*log(x^2 - x
 + 1) - 1/6*(d*x^3 - 2*e*x^2 - d*x - e)/(x^4 + x^2 + 1)
```

Mupad [B] (verification not implemented)

Time = 0.14 (sec) , antiderivative size = 149, normalized size of antiderivative = 1.06

$$\int \frac{d+ex}{(1+x^2+x^4)^2}\,dx = \frac{-\frac{d\,x^3}{6}+\frac{e\,x^2}{3}+\frac{d\,x}{6}+\frac{e}{6}}{x^4+x^2+1}-\ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right)$$
$$+\ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{d}{4}-\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right)$$
$$+\ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right)$$
$$+\ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right)$$

```
[In] int((d + e*x)/(x^2 + x^4 + 1)^2,x)
```

```
[Out] (e/6 + (d*x)/6 - (d*x^3)/6 + (e*x^2)/3)/(x^2 + x^4 + 1) - log(x - (3^(1/2)*
1i)/2 - 1/2)*(d/4 + (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9) + log(x - (3^(1/2
)*1i)/2 + 1/2)*(d/4 - (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9) + log(x + (3^(1
/2)*1i)/2 - 1/2)*((3^(1/2)*d*1i)/18 - d/4 + (3^(1/2)*e*1i)/9) + log(x + (3^
(1/2)*1i)/2 + 1/2)*(d/4 + (3^(1/2)*d*1i)/18 - (3^(1/2)*e*1i)/9)
```

3.32 $\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 21, antiderivative size = 165

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \frac{e(1+2x^2)}{6(1+x^2+x^4)} + \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} - \frac{(4d+f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{2e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right)$$

```
[Out] 1/6*e*(2*x^2+1)/(x^4+x^2+1)+1/6*x*(d+f-(d-2*f)*x^2)/(x^4+x^2+1)-1/8*(2*d-f)
*ln(x^2-x+1)+1/8*(2*d-f)*ln(x^2+x+1)-1/36*(4*d+f)*arctan(1/3*(1-2*x)*3^(1/2
))*3^(1/2)+1/36*(4*d+f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+2/9*e*arctan(1/
3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.476, Rules used = {1687, 1192, 1183, 648, 632, 210, 642, 12, 1121, 628}

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(4d+f)}{12\sqrt{3}}+\frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(4d+f)}{12\sqrt{3}} + \frac{2e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{8}(2d-f)\log\left(x^2-x+1\right) + \frac{1}{8}(2d-f)\log\left(x^2+x+1\right) + \frac{x(-(x^2(d-2f))+d+f)}{6\left(x^4+x^2+1\right)}+\frac{e(2x^2+1)}{6\left(x^4+x^2+1\right)}$$

```
[In] Int[(d + e*x + f*x^2)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (e*(1 + 2*x^2))/(6*(1 + x^2 + x^4)) + (x*(d + f - (d - 2*f)*x^2))/(6*(1 + x
^2 + x^4)) - ((4*d + f)*ArcTan[(1 - 2*x)/Sqrt[3]])/(12*Sqrt[3]) + ((4*d + f
)*ArcTan[(1 + 2*x)/Sqrt[3]])/(12*Sqrt[3]) + (2*e*ArcTan[(1 + 2*x^2)/Sqrt[3]
])/(3*Sqrt[3]) - ((2*d - f)*Log[1 - x + x^2])/8 + ((2*d - f)*Log[1 + x + x^
2])/8
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
```

```
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{ex}{(1+x^2+x^4)^2}\,dx + \int \frac{d+fx^2}{(1+x^2+x^4)^2}\,dx \\
&= \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} + \frac{1}{6}\int \frac{5d-f+(-d+2f)x^2}{1+x^2+x^4}\,dx + e\int \frac{x}{(1+x^2+x^4)^2}\,dx \\
&= \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} + \frac{1}{12}\int \frac{5d-f-(6d-3f)x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{12}\int \frac{5d-f+(6d-3f)x}{1+x+x^2}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{e(1+2x^2)}{6(1+x^2+x^4)} + \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} + \frac{1}{3}e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&\quad + \frac{1}{8}(2d-f)\int \frac{1+2x}{1+x+x^2}\,dx + \frac{1}{8}(-2d+f)\int \frac{-1+2x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{24}(4d+f)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{24}(4d+f)\int \frac{1}{1+x+x^2}\,dx \\
&= \frac{e(1+2x^2)}{6(1+x^2+x^4)} + \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) \\
&\quad + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right) - \frac{1}{3}(2e)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) \\
&\quad + \frac{1}{12}(-4d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) \\
&\quad + \frac{1}{12}(-4d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) \\
&= \frac{e(1+2x^2)}{6(1+x^2+x^4)} + \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} - \frac{(4d+f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} \\
&\quad + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.28 (sec) , antiderivative size = 186, normalized size of antiderivative = 1.13

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\left(\frac{6(e+2ex^2+x(d+f-dx^2+2fx^2))}{1+x^2+x^4} - \frac{\left(\left(-11i+\sqrt{3}\right)d-2\left(-2i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(\left(11i+\sqrt{3}\right)d-2\left(2i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 8\sqrt{3}e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2)/(1 + x^2 + x^4)^2,x]
```

```
[Out] ((6*(e + 2*e*x^2 + x*(d + f - d*x^2 + 2*f*x^2)))/(1 + x^2 + x^4) - (((-11*I
 + Sqrt[3])*d - 2*(-2*I + Sqrt[3])*f)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqrt[(1
 + I*Sqrt[3])/6] - (((11*I + Sqrt[3])*d - 2*(2*I + Sqrt[3])*f)*ArcTan[((I +
 Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 8*Sqrt[3]*e*ArcTan[Sqrt[3]/(1 +
2*x^2)])/36
```

Maple [A] (verified)

Time = 0.27 (sec) , antiderivative size = 154, normalized size of antiderivative = 0.93

method	result
default	$-\frac{\left(\frac{d}{3}-\frac{e}{3}-\frac{2f}{3}\right)x-\frac{2d}{3}-\frac{e}{3}+\frac{f}{3}}{4(x^2-x+1)}-\frac{(6d-3f)\ln\left(x^2-x+1\right)}{24}-\frac{\left(-2d-4e-\frac{f}{2}\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{18}+\frac{\left(-\frac{d}{3}-\frac{e}{3}+\frac{2f}{3}\right)x-\frac{2d}{3}+\frac{e}{3}+\frac{f}{3}}{4x^2+4x+4}$
risch	Expression too large to display

```
[In] int((f*x^2+e*x+d)/(x^4+x^2+1)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*((1/3*d-1/3*e-2/3*f)*x-2/3*d-1/3*e+1/3*f)/(x^2-x+1)-1/24*(6*d-3*f)*ln(
x^2-x+1)-1/18*(-2*d-4*e-1/2*f)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*((-1
/3*d-1/3*e+2/3*f)*x-2/3*d+1/3*e+1/3*f)/(x^2+x+1)+1/24*(6*d-3*f)*ln(x^2+x+1)
+1/18*(2*d-4*e+1/2*f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 212, normalized size of antiderivative = 1.28

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = -\frac{12\,(d-2\,f)x^3-24\,ex^2-2\,\sqrt{3}((4\,d-8\,e+f)x^4+(4\,d-8\,e+f)x^2+4\,d-8\,e+f)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+\right.}{}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="fricas")
```

```
[Out] -1/72*(12*(d - 2*f)*x^3 - 24*e*x^2 - 2*sqrt(3)*((4*d - 8*e + f)*x^4 + (4*d
- 8*e + f)*x^2 + 4*d - 8*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*(
(4*d + 8*e + f)*x^4 + (4*d + 8*e + f)*x^2 + 4*d + 8*e + f)*arctan(1/3*sqrt(
3)*(2*x - 1)) - 12*(d + f)*x - 9*((2*d - f)*x^4 + (2*d - f)*x^2 + 2*d - f)*
log(x^2 + x + 1) + 9*((2*d - f)*x^4 + (2*d - f)*x^2 + 2*d - f)*log(x^2 - x
+ 1) - 12*e)/(x^4 + x^2 + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 81.03 (sec) , antiderivative size = 4106, normalized size of antiderivative = 24.88

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4+x**2+1)**2,x)
```

```
[Out] (-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)*log(x + (-164944*d**5*e + 16416
*d**5*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 336520*d**4*e*f + 20066
4*d**4*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) - 115200*d**3*e**3 - 5
04576*d**3*e**2*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) - 272380*d**3*e
*f**2 + 1734912*d**3*e*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 229
500*d**3*f**2*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 2612736*d**3*(-
d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 51840*d**2*e**3*f + 881280*d
**2*e**2*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 119420*d**2*e*f**3
 - 2477952*d**2*e*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**2 + 50436*
d**2*f**3*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) - 2519424*d**2*f*(-d/
4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 28672*d*e**5 + 184320*d*e**4*(
-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 8640*d*e**3*f**2 + 774144*d*e*
*3*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 409536*d*e**2*f**2*(-d/
4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 4976640*d*e**2*(-d/4 + f/8 - sqrt
(3)*I*(4*d + 8*e + f)/72)**3 - 31040*d*e*f**4 + 1270080*d*e*f**2*(-d/4 + f/
```

```
8 - sqrt(3)*I*(4*d + 8*e + f)/72)**2 + 14040*d*f**4*(-d/4 + f/8 - sqrt(3)*I
*(4*d + 8*e + f)/72) + 139968*d*f**2*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f
)/72)**3 - 20480*e**5*f - 36864*e**4*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e +
 f)/72) - 2880*e**3*f**3 - 552960*e**3*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e
 + f)/72)**2 + 70848*e**2*f**3*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)
- 995328*e**2*f*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 3956*e*f**
5 - 209088*e*f**3*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 3996*f**
5*(-d/4 + f/8 - sqrt(3)*I*(4*d + 8*e + f)/72) + 233280*f**3*(-d/4 + f/8 - s
qrt(3)*I*(4*d + 8*e + f)/72)**3)/(53568*d**6 - 69984*d**5*f - 182528*d**4*e
**2 + 23652*d**4*f**2 + 377344*d**3*e**2*f + 5400*d**3*f**3 - 126976*d**2*e
**4 - 278400*d**2*e**2*f**2 - 4131*d**2*f**4 + 102400*d*e**4*f + 93568*d*e*
*2*f**3 + 81*d*f**5 - 28672*e**4*f**2 - 11648*e**2*f**4 + 189*f**6)) + (-d/
4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)*log(x + (-164944*d**5*e + 16416*d**
5*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 336520*d**4*e*f + 200664*d*
*4*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) - 115200*d**3*e**3 - 50457
6*d**3*e**2*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) - 272380*d**3*e*f**
2 + 1734912*d**3*e*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 229500*
d**3*f**2*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 2612736*d**3*(-d/4
+ f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 51840*d**2*e**3*f + 881280*d**2*
e**2*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 119420*d**2*e*f**3 - 2
477952*d**2*e*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**2 + 50436*d**2
*f**3*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) - 2519424*d**2*f*(-d/4 +
f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 28672*d*e**5 + 184320*d*e**4*(-d/4
 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 8640*d*e**3*f**2 + 774144*d*e**3*(
-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 409536*d*e**2*f**2*(-d/4 +
f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 4976640*d*e**2*(-d/4 + f/8 + sqrt(3)*
I*(4*d + 8*e + f)/72)**3 - 31040*d*e*f**4 + 1270080*d*e*f**2*(-d/4 + f/8 +
sqrt(3)*I*(4*d + 8*e + f)/72)**2 + 14040*d*f**4*(-d/4 + f/8 + sqrt(3)*I*(4*
d + 8*e + f)/72) + 139968*d*f**2*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72
)**3 - 20480*e**5*f - 36864*e**4*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/
72) - 2880*e**3*f**3 - 552960*e**3*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f
)/72)**2 + 70848*e**2*f**3*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) - 99
5328*e**2*f*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**3 + 3956*e*f**5 -
209088*e*f**3*(-d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72)**2 - 3996*f**5*(-
d/4 + f/8 + sqrt(3)*I*(4*d + 8*e + f)/72) + 233280*f**3*(-d/4 + f/8 + sqrt(
3)*I*(4*d + 8*e + f)/72)**3)/(53568*d**6 - 69984*d**5*f - 182528*d**4*e**2
+ 23652*d**4*f**2 + 377344*d**3*e**2*f + 5400*d**3*f**3 - 126976*d**2*e**4
- 278400*d**2*e**2*f**2 - 4131*d**2*f**4 + 102400*d*e**4*f + 93568*d*e**2*f
**3 + 81*d*f**5 - 28672*e**4*f**2 - 11648*e**2*f**4 + 189*f**6)) + (d/4 - f
/8 - sqrt(3)*I*(4*d - 8*e + f)/72)*log(x + (-164944*d**5*e + 16416*d**5*(d/
4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) + 336520*d**4*e*f + 200664*d**4*f*(
d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) - 115200*d**3*e**3 - 504576*d**3*
e**2*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) - 272380*d**3*e*f**2 + 1734
912*d**3*e*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72)**2 - 229500*d**3*f**2
*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) + 2612736*d**3*(d/4 - f/8 - sqr
```

t(3)*I*(4*d - 8*e + f)/72)**3 + 51840*d**2*e**3*f + 881280*d**2*e**2*f*(d/4
 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) + 119420*d**2*e*f**3 - 2477952*d**2*
e*f*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72)**2 + 50436*d**2*f**3*(d/4 -
f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) - 2519424*d**2*f*(d/4 - f/8 - sqrt(3)*I
*(4*d - 8*e + f)/72)**3 + 28672*d*e**5 + 184320*d*e**4*(d/4 - f/8 - sqrt(3)
I(4*d - 8*e + f)/72) + 8640*d*e**3*f**2 + 774144*d*e**3*(d/4 - f/8 - sqrt
(3)*I*(4*d - 8*e + f)/72)**2 - 409536*d*e**2*f**2*(d/4 - f/8 - sqrt(3)*I*(4
*d - 8*e + f)/72) + 4976640*d*e**2*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/7
2)**3 - 31040*d*e*f**4 + 1270080*d*e*f**2*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e
 + f)/72)**2 + 14040*d*f**4*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) + 13
9968*d*f**2*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72)**3 - 20480*e**5*f -
36864*e**4*f*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) - 2880*e**3*f**3 -
552960*e**3*f*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72)**2 + 70848*e**2*f*
3(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72) - 995328*e**2*f*(d/4 - f/8 -
sqrt(3)*I*(4*d - 8*e + f)/72)**3 + 3956*e*f**5 - 209088*e*f**3*(d/4 - f/8 -
 sqrt(3)*I*(4*d - 8*e + f)/72)**2 - 3996*f**5*(d/4 - f/8 - sqrt(3)*I*(4*d -
 8*e + f)/72) + 233280*f**3*(d/4 - f/8 - sqrt(3)*I*(4*d - 8*e + f)/72)**3)/
(53568*d**6 - 69984*d**5*f - 182528*d**4*e**2 + 23652*d**4*f**2 + 377344*d*
*3*e**2*f + 5400*d**3*f**3 - 126976*d**2*e**4 - 278400*d**2*e**2*f**2 - 413
1*d**2*f**4 + 102400*d*e**4*f + 93568*d*e**2*f**3 + 81*d*f**5 - 28672*e**4*
f**2 - 11648*e**2*f**4 + 189*f**6)) + (d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f
)/72)*log(x + (-164944*d**5*e + 16416*d**5*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*
e + f)/72) + 336520*d**4*e*f + 200664*d**4*f*(d/4 - f/8 + sqrt(3)*I*(4*d -
8*e + f)/72) - 115200*d**3*e**3 - 504576*d**3*e**2*(d/4 - f/8 + sqrt(3)*I*(
4*d - 8*e + f)/72) - 272380*d**3*e*f**2 + 1734912*d**3*e*(d/4 - f/8 + sqrt(
3)*I*(4*d - 8*e + f)/72)**2 - 229500*d**3*f**2*(d/4 - f/8 + sqrt(3)*I*(4*d
- 8*e + f)/72) + 2612736*d**3*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**3
 + 51840*d**2*e**3*f + 881280*d**2*e**2*f*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e
 + f)/72) + 119420*d**2*e*f**3 - 2477952*d**2*e*f*(d/4 - f/8 + sqrt(3)*I*(4
*d - 8*e + f)/72)**2 + 50436*d**2*f**3*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e +
f)/72) - 2519424*d**2*f*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**3 + 286
72*d*e**5 + 184320*d*e**4*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72) + 8640
*d*e**3*f**2 + 774144*d*e**3*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**2
- 409536*d*e**2*f**2*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72) + 4976640*d
*e**2*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**3 - 31040*d*e*f**4 + 1270
080*d*e*f**2*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**2 + 14040*d*f**4*(
d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72) + 139968*d*f**2*(d/4 - f/8 + sqrt
(3)*I*(4*d - 8*e + f)/72)**3 - 20480*e**5*f - 36864*e**4*f*(d/4 - f/8 + sqr
t(3)*I*(4*d - 8*e + f)/72) - 2880*e**3*f**3 - 552960*e**3*f*(d/4 - f/8 + sq
rt(3)*I*(4*d - 8*e + f)/72)**2 + 70848*e**2*f**3*(d/4 - f/8 + sqrt(3)*I*(4*
d - 8*e + f)/72) - 995328*e**2*f*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)
3 + 3956*e*f5 - 209088*e*f**3*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72
)**2 - 3996*f**5*(d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72) + 233280*f**3*(
d/4 - f/8 + sqrt(3)*I*(4*d - 8*e + f)/72)**3)/(53568*d**6 - 69984*d**5*f -
182528*d**4*e**2 + 23652*d**4*f**2 + 377344*d**3*e**2*f + 5400*d**3*f**3 -

```
126976*d**2*e**4 - 278400*d**2*e**2*f**2 - 4131*d**2*f**4 + 102400*d*e**4*f
 + 93568*d*e**2*f**3 + 81*d*f**5 - 28672*e**4*f**2 - 11648*e**2*f**4 + 189*
f**6)) + (2*e*x**2 + e + x**3*(-d + 2*f) + x*(d + f))/(6*x**4 + 6*x**2 + 6)
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 120, normalized size of antiderivative = 0.73

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\sqrt{3}(4d-8e+f)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{36}\sqrt{3}(4d+8e+f)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{8}(2d-f)\log\left(x^2+x+1\right) - \frac{1}{8}(2d-f)\log\left(x^2-x+1\right) - \frac{(d-2f)x^3-2ex^2-(d+f)x-e}{6(x^4+x^2+1)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="maxima")
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36*sqrt(3)*(
4*d + 8*e + f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d - f)*log(x^2 + x +
1) - 1/8*(2*d - f)*log(x^2 - x + 1) - 1/6*((d - 2*f)*x^3 - 2*e*x^2 - (d + f
)*x - e)/(x^4 + x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 124, normalized size of antiderivative = 0.75

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\sqrt{3}(4d-8e+f)\arctan\left(\frac{1}{3}\sqrt{3}(2x+1)\right) + \frac{1}{36}\sqrt{3}(4d+8e+f)\arctan\left(\frac{1}{3}\sqrt{3}(2x-1)\right) + \frac{1}{8}(2d-f)\log\left(x^2+x+1\right) - \frac{1}{8}(2d-f)\log\left(x^2-x+1\right) - \frac{dx^3-2fx^3-2ex^2-dx-fx-e}{6(x^4+x^2+1)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="giac")
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36*sqrt(3)*(
4*d + 8*e + f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d - f)*log(x^2 + x +
1) - 1/8*(2*d - f)*log(x^2 - x + 1) - 1/6*(d*x^3 - 2*f*x^3 - 2*e*x^2 - d*x
- f*x - e)/(x^4 + x^2 + 1)
```

Mupad [B] (verification not implemented)

Time = 0.18 (sec) , antiderivative size = 201, normalized size of antiderivative = 1.22

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^2}\,dx = \frac{\left(\frac{f}{3}-\frac{d}{6}\right)x^3+\frac{e\,x^2}{3}+\left(\frac{d}{6}+\frac{f}{6}\right)x+\frac{e}{6}}{x^4+x^2+1} - \ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{8}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}\right) - \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{8}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{f}{8}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{d}{4}-\frac{f}{8}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}\right)$$

```
[In] int((d + e*x + f*x^2)/(x^2 + x^4 + 1)^2,x)
```

```
[Out] (e/6 - x^3*(d/6 - f/3) + (e*x^2)/3 + x*(d/6 + f/6))/(x^2 + x^4 + 1) - log(x
 - (3^(1/2)*1i)/2 - 1/2)*(d/4 - f/8 + (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9
+ (3^(1/2)*f*1i)/72) - log(x - (3^(1/2)*1i)/2 + 1/2)*(f/8 - d/4 + (3^(1/2)*
d*1i)/18 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72) + log(x + (3^(1/2)*1i)/2 -
 1/2)*(f/8 - d/4 + (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72
) + log(x + (3^(1/2)*1i)/2 + 1/2)*(d/4 - f/8 + (3^(1/2)*d*1i)/18 - (3^(1/2)
*e*1i)/9 + (3^(1/2)*f*1i)/72)
```

3.33 $\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^2} dx$

Optimal result

Integrand size = 26, antiderivative size = 179

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^2} dx = \frac{x(d+f-(d-2f)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+(2e-g)x^2}{6(1+x^2+x^4)} - \frac{(4d+f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right)$$

```
[Out] 1/6*x*(d+f-(d-2*f)*x^2)/(x^4+x^2+1)+1/6*(e-2*g+(2*e-g)*x^2)/(x^4+x^2+1)-1/8
*(2*d-f)*ln(x^2-x+1)+1/8*(2*d-f)*ln(x^2+x+1)-1/36*(4*d+f)*arctan(1/3*(1-2*x
)*3^(1/2))*3^(1/2)+1/36*(4*d+f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/9*(2*
e-g)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 179, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.346$, Rules used

$= \{1687, 1192, 1183, 648, 632, 210, 642, 1261, 652\}$

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^2}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(4d+f)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(4d+f)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(x^2-x+1\right) + \frac{1}{8}(2d-f)\log\left(x^2+x+1\right) + \frac{x(-(x^2(d-2f))+d+f)}{6\left(x^4+x^2+1\right)} + \frac{x^2(2e-g)+e-2g}{6\left(x^4+x^2+1\right)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (x*(d + f - (d - 2*f)*x^2))/(6*(1 + x^2 + x^4)) + (e - 2*g + (2*e - g)*x^2)
/(6*(1 + x^2 + x^4)) - ((4*d + f)*ArcTan[(1 - 2*x)/Sqrt[3]])/(12*Sqrt[3]) +
 ((4*d + f)*ArcTan[(1 + 2*x)/Sqrt[3]])/(12*Sqrt[3]) + ((2*e - g)*ArcTan[(1
+ 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((2*d - f)*Log[1 - x + x^2])/8 + ((2*d - f
)*Log[1 + x + x^2])/8
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{(1+x^2+x^4)^2}\,dx+\int \frac{x(e+gx^2)}{(1+x^2+x^4)^2}\,dx\\
&= \frac{x(d+f-(d-2f)x^2)}{6\,(1+x^2+x^4)}+\frac{1}{6}\int \frac{5d-f+(-d+2f)x^2}{1+x^2+x^4}\,dx\\
&\quad+\frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(1+x+x^2)^2}\,dx,x,x^2\right)
\end{aligned}$$

$$= \frac{x(d+f-(d-2f)x^2)}{6\left(1+x^2+x^4\right)} + \frac{e-2g+(2e-g)x^2}{6\left(1+x^2+x^4\right)} + \frac{1}{12}\int \frac{5d-f-(6d-3f)x}{1-x+x^2}\,dx + \frac{1}{12}\int \frac{5d-f+(6d-3f)x}{1+x+x^2}\,dx + \frac{1}{6}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right)$$

$$= \frac{x(d+f-(d-2f)x^2)}{6\left(1+x^2+x^4\right)} + \frac{e-2g+(2e-g)x^2}{6\left(1+x^2+x^4\right)} + \frac{1}{8}(2d-f)\int \frac{1+2x}{1+x+x^2}\,dx + \frac{1}{8}(-2d+f)\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{24}(4d+f)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{24}(4d+f)\int \frac{1}{1+x+x^2}\,dx + \frac{1}{3}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right)$$

$$= \frac{x(d+f-(d-2f)x^2)}{6\left(1+x^2+x^4\right)} + \frac{e-2g+(2e-g)x^2}{6\left(1+x^2+x^4\right)} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right) + \frac{1}{12}(-4d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) + \frac{1}{12}(-4d-f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)$$

$$= \frac{x(d+f-(d-2f)x^2)}{6\left(1+x^2+x^4\right)} + \frac{e-2g+(2e-g)x^2}{6\left(1+x^2+x^4\right)} - \frac{(4d+f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f)\log\left(1+x+x^2\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.27 (sec) , antiderivative size = 200, normalized size of antiderivative = 1.12

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\left(\frac{6(e+2ex^2-g(2+x^2)+x(d+f-dx^2+2fx^2))}{1+x^2+x^4} - \frac{\left(\left(-11i+\sqrt{3}\right)d-2\left(-2i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(\left(11i+\sqrt{3}\right)d-2\left(2i+\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 4\sqrt{3}(2e-g)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4)^2,x]
```

```
[Out] ((6*(e + 2*e*x^2 - g*(2 + x^2) + x*(d + f - d*x^2 + 2*f*x^2)))/(1 + x^2 + x
^4) - (((-11*I + Sqrt[3])*d - 2*(-2*I + Sqrt[3])*f)*ArcTan[((-I + Sqrt[3])*
x)/2])/Sqrt[(1 + I*Sqrt[3])/6] - (((11*I + Sqrt[3])*d - 2*(2*I + Sqrt[3])*f
)*ArcTan[((I + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 4*Sqrt[3]*(2*e - g
)*ArcTan[Sqrt[3]/(1 + 2*x^2)])/36
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 172, normalized size of antiderivative = 0.96

method	result
default	$-\frac{\left(\frac{d}{3}-\frac{e}{3}-\frac{g}{3}-\frac{2f}{3}\right)x-\frac{2d}{3}-\frac{e}{3}+\frac{2g}{3}+\frac{f}{3}}{4(x^2-x+1)}-\frac{(6d-3f)\ln(x^2-x+1)}{24}-\frac{\left(-2d-4e-\frac{f}{2}+2g\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{18}+\frac{\left(-\frac{d}{3}-\frac{e}{3}-\frac{g}{3}+\frac{2f}{3}\right.}{4x^2+}$
risch	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*((1/3*d-1/3*e-1/3*g-2/3*f)*x-2/3*d-1/3*e+2/3*g+1/3*f)/(x^2-x+1)-1/24*(
6*d-3*f)*ln(x^2-x+1)-1/18*(-2*d-4*e-1/2*f+2*g)*3^(1/2)*arctan(1/3*(2*x-1)*3
^(1/2))+1/4*((-1/3*d-1/3*e-1/3*g+2/3*f)*x-2/3*d+1/3*e-2/3*g+1/3*f)/(x^2+x+1
)+1/24*(6*d-3*f)*ln(x^2+x+1)+1/18*(2*d-4*e+1/2*f+2*g)*arctan(1/3*(1+2*x)*3^
(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.46 (sec) , antiderivative size = 239, normalized size of antiderivative = 1.34

$$\int \frac{d+ex+fx^2+gx^3}{\left(1+x^2+x^4\right)^2}\,dx = -\frac{12\,(d-2\,f)x^3-12\,(2\,e-g)x^2-2\,\sqrt{3}((4\,d-8\,e+f+4\,g)x^4+(4\,d-8\,e+f+4\,g)x^2+4\,d-8\,e+f}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="fricas")
```

```
[Out] -1/72*(12*(d - 2*f)*x^3 - 12*(2*e - g)*x^2 - 2*sqrt(3)*((4*d - 8*e + f + 4*
g)*x^4 + (4*d - 8*e + f + 4*g)*x^2 + 4*d - 8*e + f + 4*g)*arctan(1/3*sqrt(3
)*(2*x + 1)) - 2*sqrt(3)*((4*d + 8*e + f - 4*g)*x^4 + (4*d + 8*e + f - 4*g)
*x^2 + 4*d + 8*e + f - 4*g)*arctan(1/3*sqrt(3)*(2*x - 1)) - 12*(d + f)*x -
9*((2*d - f)*x^4 + (2*d - f)*x^2 + 2*d - f)*log(x^2 + x + 1) + 9*((2*d - f)
*x^4 + (2*d - f)*x^2 + 2*d - f)*log(x^2 - x + 1) - 12*e + 24*g)/(x^4 + x^2
+ 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{\left(1+x^2+x^4\right)^2}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 135, normalized size of antiderivative = 0.75

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{\left(1+x^2+x^4\right)^2}\,dx = {} & \frac{1}{36}\,\sqrt{3}(4\,d-8\,e+f+4\,g)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x+1)\right) \\ & +\frac{1}{36}\,\sqrt{3}(4\,d+8\,e+f-4\,g)\arctan\left(\frac{1}{3}\,\sqrt{3}(2\,x-1)\right) \\ & +\frac{1}{8}\,(2\,d-f)\log\left(x^2+x+1\right)-\frac{1}{8}\,(2\,d-f)\log\left(x^2-x+1\right) \\ & -\frac{(d-2\,f)x^3-(2\,e-g)x^2-(d+f)x-e+2\,g}{6\,(x^4+x^2+1)}\end{aligned}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="maxima")
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36*sqr
t(3)*(4*d + 8*e + f - 4*g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d - f)*lo
g(x^2 + x + 1) - 1/8*(2*d - f)*log(x^2 - x + 1) - 1/6*((d - 2*f)*x^3 - (2*e
 - g)*x^2 - (d + f)*x - e + 2*g)/(x^4 + x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.77

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\sqrt{3}(4\,d-8\,e+f+4\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{36}\sqrt{3}(4\,d+8\,e+f-4\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{1}{8}\,(2\,d-f)\log\left(x^2+x+1\right) - \frac{1}{8}\,(2\,d-f)\log\left(x^2-x+1\right) - \frac{dx^3-2\,fx^3-2\,ex^2+gx^2-dx-fx-e+2\,g}{6\,(x^4+x^2+1)}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="giac")
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36*sqr
t(3)*(4*d + 8*e + f - 4*g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d - f)*lo
g(x^2 + x + 1) - 1/8*(2*d - f)*log(x^2 - x + 1) - 1/6*(d*x^3 - 2*f*x^3 - 2*
e*x^2 + g*x^2 - d*x - f*x - e + 2*g)/(x^4 + x^2 + 1)
```

Mupad [B] (verification not implemented)

Time = 8.14 (sec) , antiderivative size = 237, normalized size of antiderivative = 1.32

$$\int \frac{d+ex+fx^2+gx^3}{\left(1+x^2+x^4\right)^2}\,dx = \frac{\left(\frac{f}{3}-\frac{d}{6}\right)\,x^3+\left(\frac{e}{3}-\frac{g}{6}\right)\,x^2+\left(\frac{d}{6}+\frac{f}{6}\right)\,x+\frac{e}{6}-\frac{g}{3}}{x^4+x^2+1}-\ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{d}{4}-\frac{f}{8}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right)-\ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right)+\ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{d}{4}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right)+\ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{d}{4}-\frac{f}{8}+\frac{\sqrt{3}\,d\,1\mathrm{i}}{18}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{72}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^2 + x^4 + 1)^2,x)
```

```
[Out] (e/6 - g/3 - x^3*(d/6 - f/3) + x^2*(e/3 - g/6) + x*(d/6 + f/6))/(x^2 + x^4
+ 1) - log(x - (3^(1/2)*1i)/2 - 1/2)*(d/4 - f/8 + (3^(1/2)*d*1i)/18 + (3^(1
/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 - (3^(1/2)*g*1i)/18) - log(x - (3^(1/2)*1i)
/2 + 1/2)*(f/8 - d/4 + (3^(1/2)*d*1i)/18 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i
)/72 + (3^(1/2)*g*1i)/18) + log(x + (3^(1/2)*1i)/2 - 1/2)*(f/8 - d/4 + (3^(
1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 - (3^(1/2)*g*1i)/18) +
 log(x + (3^(1/2)*1i)/2 + 1/2)*(d/4 - f/8 + (3^(1/2)*d*1i)/18 - (3^(1/2)*e*
1i)/9 + (3^(1/2)*f*1i)/72 + (3^(1/2)*g*1i)/18)
```

3.34 $\int \frac{d+ex+fx^2+gx^3+hx^4}{\left(1+x^2+x^4\right)^2}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 187

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{\left(1+x^2+x^4\right)^2}\,dx = \frac{e-2g+(2e-g)x^2}{6\left(1+x^2+x^4\right)}+\frac{x\left(d+f-2h-(d-2f+h)x^2\right)}{6\left(1+x^2+x^4\right)} - \frac{(4d+f+h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f+h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}}+\frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f+h)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f+h)\log\left(1+x+x^2\right)$$

```
[Out] 1/6*(e-2*g+(2*e-g)*x^2)/(x^4+x^2+1)+1/6*x*(d+f-2*h-(d-2*f+h)*x^2)/(x^4+x^2+
1)-1/8*(2*d-f+h)*ln(x^2-x+1)+1/8*(2*d-f+h)*ln(x^2+x+1)-1/36*(4*d+f+h)*arcta
n(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/36*(4*d+f+h)*arctan(1/3*(1+2*x)*3^(1/2))*3
^(1/2)+1/9*(2*e-g)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.11 (sec) , antiderivative size = 187, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.290, Rules used = {1687, 1692, 1183, 648, 632, 210, 642, 1261, 652}

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(4d+f+h)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(4d+f+h)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{3\sqrt{3}} - \frac{1}{8}\log\left(x^2-x+1\right)(2d-f+h) + \frac{1}{8}\log\left(x^2+x+1\right)(2d-f+h) + \frac{x(-(x^2(d-2f+h))+d+f-2h)}{6\left(x^4+x^2+1\right)} + \frac{x^2(2e-g)+e-2g}{6\left(x^4+x^2+1\right)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (e - 2*g + (2*e - g)*x^2)/(6*(1 + x^2 + x^4)) + (x*(d + f - 2*h - (d - 2*f
+ h)*x^2))/(6*(1 + x^2 + x^4)) - ((4*d + f + h)*ArcTan[(1 - 2*x)/Sqrt[3]])/
(12*Sqrt[3]) + ((4*d + f + h)*ArcTan[(1 + 2*x)/Sqrt[3]])/(12*Sqrt[3]) + ((2
*e - g)*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((2*d - f + h)*Log[1 - x
 + x^2])/8 + ((2*d - f + h)*Log[1 + x + x^2])/8
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
```

```
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{(1+x^2+x^4)^2}\,dx + \int \frac{d+fx^2+hx^4}{(1+x^2+x^4)^2}\,dx \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{6\,(1+x^2+x^4)} + \frac{1}{6}\int \frac{5d-f+2h+(-d+2f-h)x^2}{1+x^2+x^4}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{e-2g+(2e-g)x^2}{6\,(1+x^2+x^4)} + \frac{x(d+f-2h-(d-2f+h)x^2)}{6\,(1+x^2+x^4)} \\
&\quad + \frac{1}{12}\int \frac{5d-f+2h-(6d-3f+3h)x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{12}\int \frac{5d-f+2h+(6d-3f+3h)x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{6}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&= \frac{e-2g+(2e-g)x^2}{6\,(1+x^2+x^4)} + \frac{x(d+f-2h-(d-2f+h)x^2)}{6\,(1+x^2+x^4)} \\
&\quad + \frac{1}{3}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) \\
&\quad + \frac{1}{8}(-2d+f-h)\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{8}(2d-f+h)\int \frac{1+2x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{24}(4d+f+h)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{24}(4d+f+h)\int \frac{1}{1+x+x^2}\,dx \\
&= \frac{e-2g+(2e-g)x^2}{6\,(1+x^2+x^4)} + \frac{x(d+f-2h-(d-2f+h)x^2)}{6\,(1+x^2+x^4)} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} \\
&\quad - \frac{1}{8}(2d-f+h)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f+h)\log\left(1+x+x^2\right) \\
&\quad + \frac{1}{12}(-4d-f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) \\
&\quad + \frac{1}{12}(-4d-f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)
\end{aligned}$$

$$
\begin{aligned}
&=\frac{e-2g+(2e-g)x^2}{6(1+x^2+x^4)}+\frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)}\\
&\quad-\frac{(4d+f+h)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}}+\frac{(4d+f+h)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}}\\
&\quad+\frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{8}(2d-f+h)\log\left(1-x+x^2\right)\\
&\quad+\frac{1}{8}(2d-f+h)\log\left(1+x+x^2\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.39 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.25

$$
\begin{aligned}
&\int\frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx\\
&=\frac{1}{36}\left(-\frac{6(g(2+x^2)-e(1+2x^2)+x(d(-1+x^2)+h(2+x^2)-f(1+2x^2)))}{1+x^2+x^4}\right.\\
&\quad-\frac{\left(\left(-11i+\sqrt{3}\right)d-2\left(-2i+\sqrt{3}\right)f+\left(-5i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}}\\
&\quad-\frac{\left(\left(11i+\sqrt{3}\right)d-2\left(2i+\sqrt{3}\right)f+\left(5i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}}\\
&\quad\left.-4\sqrt{3}(2e-g)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)
\end{aligned}
$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4)^2,x]
```

```
[Out] ((-6*(g*(2 + x^2) - e*(1 + 2*x^2) + x*(d*(-1 + x^2) + h*(2 + x^2) - f*(1 +
2*x^2))))/(1 + x^2 + x^4) - (((-11*I + Sqrt[3])*d - 2*(-2*I + Sqrt[3])*f +
(-5*I + Sqrt[3])*h)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqrt[(1 + I*Sqrt[3])/6] -
 (((11*I + Sqrt[3])*d - 2*(2*I + Sqrt[3])*f + (5*I + Sqrt[3])*h)*ArcTan[((I
 + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 4*Sqrt[3]*(2*e - g)*ArcTan[Sqr
t[3]/(1 + 2*x^2)])/36
```

Maple [A] (verified)

Time = 0.41 (sec) , antiderivative size = 196, normalized size of antiderivative = 1.05

method	result
default	$-\frac{\left(\frac{d}{3}-\frac{2f}{3}-\frac{g}{3}-\frac{e}{3}+\frac{h}{3}\right)x-\frac{2d}{3}+\frac{f}{3}+\frac{2g}{3}-\frac{e}{3}+\frac{h}{3}}{4(x^2-x+1)}-\frac{(6d-3f+3h)\ln(x^2-x+1)}{24}-\frac{\left(-2d-4e-\frac{f}{2}+2g-\frac{h}{2}\right)\sqrt{3}\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{18}+$
risch	Expression too large to display

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*((1/3*d-2/3*f-1/3*g-1/3*e+1/3*h)*x-2/3*d+1/3*f+2/3*g-1/3*e+1/3*h)/(x^2
-x+1)-1/24*(6*d-3*f+3*h)*ln(x^2-x+1)-1/18*(-2*d-4*e-1/2*f+2*g-1/2*h)*3^(1/2
)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*((-1/3*d+2/3*f-1/3*g-1/3*e-1/3*h)*x-2/3*d
+1/3*f-2/3*g+1/3*e+1/3*h)/(x^2+x+1)+1/24*(6*d-3*f+3*h)*ln(x^2+x+1)+1/18*(2*
d-4*e+1/2*f+2*g+1/2*h)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 1.10 (sec) , antiderivative size = 255, normalized size of antiderivative = 1.36

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx = -\frac{12\,(d-2\,f+h)x^3-12\,(2\,e-g)x^2-2\,\sqrt{3}((4\,d-8\,e+f+4\,g+h)x^4+(4\,d-8\,e+f+4\,g+h)x^2+4}{}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="fricas")
```

```
[Out] -1/72*(12*(d - 2*f + h)*x^3 - 12*(2*e - g)*x^2 - 2*sqrt(3)*((4*d - 8*e + f
+ 4*g + h)*x^4 + (4*d - 8*e + f + 4*g + h)*x^2 + 4*d - 8*e + f + 4*g + h)*a
rctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*((4*d + 8*e + f - 4*g + h)*x^4 + (
4*d + 8*e + f - 4*g + h)*x^2 + 4*d + 8*e + f - 4*g + h)*arctan(1/3*sqrt(3)*
(2*x - 1)) - 12*(d + f - 2*h)*x - 9*((2*d - f + h)*x^4 + (2*d - f + h)*x^2
+ 2*d - f + h)*log(x^2 + x + 1) + 9*((2*d - f + h)*x^4 + (2*d - f + h)*x^2
+ 2*d - f + h)*log(x^2 - x + 1) - 12*e + 24*g)/(x^4 + x^2 + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 143, normalized size of antiderivative = 0.76

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx &= \frac{1}{36}\sqrt{3}(4\,d-8\,e+f+4\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)\\ &\quad+\frac{1}{36}\sqrt{3}(4\,d+8\,e+f-4\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)\\ &\quad+\frac{1}{8}(2\,d-f+h)\log\left(x^2+x+1\right)-\frac{1}{8}(2\,d-f+h)\log\left(x^2-x+1\right)\\ &\quad-\frac{(d-2\,f+h)x^3-(2\,e-g)x^2-(d+f-2\,h)x-e+2\,g}{6\,(x^4+x^2+1)}\end{aligned}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="maxima")
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g + h)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36
*sqrt(3)*(4*d + 8*e + f - 4*g + h)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d
 - f + h)*log(x^2 + x + 1) - 1/8*(2*d - f + h)*log(x^2 - x + 1) - 1/6*((d -
 2*f + h)*x^3 - (2*e - g)*x^2 - (d + f - 2*h)*x - e + 2*g)/(x^4 + x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 151, normalized size of antiderivative = 0.81

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx &= \frac{1}{36}\sqrt{3}(4\,d-8\,e+f+4\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)\\ &+\frac{1}{36}\sqrt{3}(4\,d+8\,e+f-4\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)\\ &+\frac{1}{8}\,(2\,d-f+h)\log\left(x^2+x+1\right)-\frac{1}{8}\,(2\,d-f+h)\log\left(x^2-x+1\right)\\ &-\frac{dx^3-2\,fx^3+hx^3-2\,ex^2+gx^2-dx-fx+2\,hx-e+2\,g}{6\,(x^4+x^2+1)}\end{aligned}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="giac")
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g + h)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/36
*sqrt(3)*(4*d + 8*e + f - 4*g + h)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/8*(2*d
 - f + h)*log(x^2 + x + 1) - 1/8*(2*d - f + h)*log(x^2 - x + 1) - 1/6*(d*x^
3 - 2*f*x^3 + h*x^3 - 2*e*x^2 + g*x^2 - d*x - f*x + 2*h*x - e + 2*g)/(x^4 +
 x^2 + 1)
```

Mupad [B] (verification not implemented)

Time = 3.31 (sec) , antiderivative size = 1547, normalized size of antiderivative = 8.27

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^2 + x^4 + 1)^2,x)
```

```
[Out] (e/6 - g/3 + x^2*(e/3 - g/6) + x*(d/6 + f/6 - h/3) - x^3*(d/6 - f/3 + h/6))
/(x^2 + x^4 + 1) - log(60*d*g - 153*d*f - 120*d*e + 24*e*f + 135*d*h - 48*e
*h - 12*f*g - 81*f*h + 24*g*h + 3^(1/2)*d^2*90i + 3^(1/2)*f^2*9i + 3^(1/2)*
h^2*18i - 198*d^2*x - 36*f^2*x - 45*h^2*x + 126*d^2 + 45*f^2 + 36*h^2 + 3^(
1/2)*d*e*56i - 3^(1/2)*d*f*63i - 3^(1/2)*d*g*28i - 3^(1/2)*e*f*40i + 3^(1/2
)*d*h*81i + 3^(1/2)*e*h*32i + 3^(1/2)*f*g*20i - 3^(1/2)*f*h*27i - 3^(1/2)*g
*h*16i - 24*d*e*x + 171*d*f*x + 12*d*g*x + 48*e*f*x - 189*d*h*x - 24*e*h*x
- 24*f*g*x + 81*f*h*x + 12*g*h*x + 3^(1/2)*d^2*x*18i + 3^(1/2)*f^2*x*18i +
3^(1/2)*h^2*x*9i - 3^(1/2)*d*f*x*45i + 3^(1/2)*d*g*x*44i + 3^(1/2)*e*f*x*32
i + 3^(1/2)*d*h*x*27i - 3^(1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h
*x*27i + 3^(1/2)*g*h*x*20i - 3^(1/2)*d*e*x*88i)*(d/4 - f/8 + h/8 + (3^(1/2)
*d*1i)/18 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 - (3^(1/2)*g*1i)/18 + (3^(
1/2)*h*1i)/72) - log(120*d*e - 153*d*f - 60*d*g - 24*e*f + 135*d*h + 48*e*h
 + 12*f*g - 81*f*h - 24*g*h - 3^(1/2)*d^2*90i - 3^(1/2)*f^2*9i - 3^(1/2)*h^
```

```
2*18i + 198*d^2*x + 36*f^2*x + 45*h^2*x + 126*d^2 + 45*f^2 + 36*h^2 + 3^(1/
2)*d*e*56i + 3^(1/2)*d*f*63i - 3^(1/2)*d*g*28i - 3^(1/2)*e*f*40i - 3^(1/2)*
d*h*81i + 3^(1/2)*e*h*32i + 3^(1/2)*f*g*20i + 3^(1/2)*f*h*27i - 3^(1/2)*g*h
*16i - 24*d*e*x - 171*d*f*x + 12*d*g*x + 48*e*f*x + 189*d*h*x - 24*e*h*x -
24*f*g*x - 81*f*h*x + 12*g*h*x + 3^(1/2)*d^2*x*18i + 3^(1/2)*f^2*x*18i + 3^
(1/2)*h^2*x*9i - 3^(1/2)*d*f*x*45i - 3^(1/2)*d*g*x*44i - 3^(1/2)*e*f*x*32i
+ 3^(1/2)*d*h*x*27i + 3^(1/2)*e*h*x*40i + 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h*x
*27i - 3^(1/2)*g*h*x*20i + 3^(1/2)*d*e*x*88i)*(f/8 - d/4 - h/8 + (3^(1/2)*d
*1i)/18 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 + (3^(1/2)*g*1i)/18 + (3^(1/
2)*h*1i)/72) + log(120*d*e - 153*d*f - 60*d*g - 24*e*f + 135*d*h + 48*e*h +
 12*f*g - 81*f*h - 24*g*h + 3^(1/2)*d^2*90i + 3^(1/2)*f^2*9i + 3^(1/2)*h^2*
18i + 198*d^2*x + 36*f^2*x + 45*h^2*x + 126*d^2 + 45*f^2 + 36*h^2 - 3^(1/2)
*d*e*56i - 3^(1/2)*d*f*63i + 3^(1/2)*d*g*28i + 3^(1/2)*e*f*40i + 3^(1/2)*d*
h*81i - 3^(1/2)*e*h*32i - 3^(1/2)*f*g*20i - 3^(1/2)*f*h*27i + 3^(1/2)*g*h*1
6i - 24*d*e*x - 171*d*f*x + 12*d*g*x + 48*e*f*x + 189*d*h*x - 24*e*h*x - 24
*f*g*x - 81*f*h*x + 12*g*h*x - 3^(1/2)*d^2*x*18i - 3^(1/2)*f^2*x*18i - 3^(1
/2)*h^2*x*9i + 3^(1/2)*d*f*x*45i + 3^(1/2)*d*g*x*44i + 3^(1/2)*e*f*x*32i -
3^(1/2)*d*h*x*27i - 3^(1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i + 3^(1/2)*f*h*x*2
7i + 3^(1/2)*g*h*x*20i - 3^(1/2)*d*e*x*88i)*(d/4 - f/8 + h/8 + (3^(1/2)*d*1
i)/18 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 + (3^(1/2)*g*1i)/18 + (3^(1/2)
*h*1i)/72) + log(120*d*e + 153*d*f - 60*d*g - 24*e*f - 135*d*h + 48*e*h + 1
2*f*g + 81*f*h - 24*g*h + 3^(1/2)*d^2*90i + 3^(1/2)*f^2*9i + 3^(1/2)*h^2*18
i + 198*d^2*x + 36*f^2*x + 45*h^2*x - 126*d^2 - 45*f^2 - 36*h^2 + 3^(1/2)*d
*e*56i - 3^(1/2)*d*f*63i - 3^(1/2)*d*g*28i - 3^(1/2)*e*f*40i + 3^(1/2)*d*h*
81i + 3^(1/2)*e*h*32i + 3^(1/2)*f*g*20i - 3^(1/2)*f*h*27i - 3^(1/2)*g*h*16i
 + 24*d*e*x - 171*d*f*x - 12*d*g*x - 48*e*f*x + 189*d*h*x + 24*e*h*x + 24*f
*g*x - 81*f*h*x - 12*g*h*x + 3^(1/2)*d^2*x*18i + 3^(1/2)*f^2*x*18i + 3^(1/2
)*h^2*x*9i - 3^(1/2)*d*f*x*45i + 3^(1/2)*d*g*x*44i + 3^(1/2)*e*f*x*32i + 3^
(1/2)*d*h*x*27i - 3^(1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h*x*27i
 + 3^(1/2)*g*h*x*20i - 3^(1/2)*d*e*x*88i)*(f/8 - d/4 - h/8 + (3^(1/2)*d*1i)
/18 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 - (3^(1/2)*g*1i)/18 + (3^(1/2)*h
*1i)/72)
```

3.35 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 194

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx = \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+i+(2e-g-i)x^2}{6(1+x^2+x^4)} - \frac{(4d+f+h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f+h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(2e-g+2i)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f+h)\log\left(1-x+x^2\right) + \frac{1}{8}(2d-f+h)\log\left(1+x+x^2\right)$$

```
[Out] 1/6*x*(d+f-2*h-(d-2*f+h)*x^2)/(x^4+x^2+1)+1/6*(e-2*g+i+(2*e-g-i)*x^2)/(x^4+
x^2+1)-1/8*(2*d-f+h)*ln(x^2-x+1)+1/8*(2*d-f+h)*ln(x^2+x+1)-1/36*(4*d+f+h)*a
rctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/36*(4*d+f+h)*arctan(1/3*(1+2*x)*3^(1/2
))*3^(1/2)+1/9*(2*e-g+2*i)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 194, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.278, Rules used = {1687, 1692, 1183, 648, 632, 210, 642, 1677, 1674, 12}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(4d+f+h)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(4d+f+h)}{12\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g+2i)}{3\sqrt{3}} - \frac{1}{8}\log\left(x^2-x+1\right)(2d-f+h) + \frac{1}{8}\log\left(x^2+x+1\right)(2d-f+h) + \frac{x(-(x^2(d-2f+h))+d+f-2h)}{6\left(x^4+x^2+1\right)} + \frac{x^2(2e-g-i)+e-2g+i}{6\left(x^4+x^2+1\right)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4)^2,x]
```

```
[Out] (x*(d + f - 2*h - (d - 2*f + h)*x^2))/(6*(1 + x^2 + x^4)) + (e - 2*g + i +
(2*e - g - i)*x^2)/(6*(1 + x^2 + x^4)) - ((4*d + f + h)*ArcTan[(1 - 2*x)/Sq
rt[3]])/(12*Sqrt[3]) + ((4*d + f + h)*ArcTan[(1 + 2*x)/Sqrt[3]])/(12*Sqrt[3
]) + ((2*e - g + 2*i)*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((2*d - f
+ h)*Log[1 - x + x^2])/8 + ((2*d - f + h)*Log[1 + x + x^2])/8
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
```

```
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
```

```
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2+hx^4}{(1+x^2+x^4)^2}\,dx + \int \frac{x(e+gx^2+ix^4)}{(1+x^2+x^4)^2}\,dx \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{1}{6}\int \frac{5d-f+2h+(-d+2f-h)x^2}{1+x^2+x^4}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+i+(2e-g-i)x^2}{6(1+x^2+x^4)} \\
&\quad + \frac{1}{12}\int \frac{5d-f+2h-(6d-3f+3h)x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{12}\int \frac{5d-f+2h+(6d-3f+3h)x}{1+x+x^2}\,dx + \frac{1}{6}\text{Subst}\left(\int \frac{2e-g+2i}{1+x+x^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+i+(2e-g-i)x^2}{6(1+x^2+x^4)} \\
&\quad + \frac{1}{8}(-2d+f-h)\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{8}(2d-f+h)\int \frac{1+2x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{24}(4d+f+h)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{24}(4d+f+h)\int \frac{1}{1+x+x^2}\,dx \\
&\quad + \frac{1}{6}(2e-g+2i)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right)
\end{aligned}$$

$$= \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+i+(2e-g-i)x^2}{6(1+x^2+x^4)} - \frac{1}{8}(2d-f+h)\log(1-x+x^2) + \frac{1}{8}(2d-f+h)\log(1+x+x^2) + \frac{1}{12}(-4d-f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) + \frac{1}{12}(-4d-f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) + \frac{1}{3}(-2e+g-2i)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right)$$

$$= \frac{x(d+f-2h-(d-2f+h)x^2)}{6(1+x^2+x^4)} + \frac{e-2g+i+(2e-g-i)x^2}{6(1+x^2+x^4)} - \frac{(4d+f+h)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(4d+f+h)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{12\sqrt{3}} + \frac{(2e-g+2i)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{8}(2d-f+h)\log(1-x+x^2) + \frac{1}{8}(2d-f+h)\log(1+x+x^2)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.39 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.25

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx = \frac{1}{36}\left(\frac{6(e+i+dx+fx-2hx+2ex^2-ix^2-dx^3+2fx^3-hx^3-g(2+x^2))}{1+x^2+x^4} - \frac{\left(\left(-11i+\sqrt{3}\right)d-2\left(-2i+\sqrt{3}\right)f+\left(-5i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(\left(11i+\sqrt{3}\right)d-2\left(2i+\sqrt{3}\right)f+\left(5i+\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 4\sqrt{3}(2e-g+2i)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4)^2,x]
```

```
[Out] ((6*(e + i + d*x + f*x - 2*h*x + 2*e*x^2 - i*x^2 - d*x^3 + 2*f*x^3 - h*x^3
- g*(2 + x^2)))/(1 + x^2 + x^4) - (((-11*I + Sqrt[3])*d - 2*(-2*I + Sqrt[3]
)*f + (-5*I + Sqrt[3])*h)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqrt[(1 + I*Sqrt[3]
)/6] - (((11*I + Sqrt[3])*d - 2*(2*I + Sqrt[3])*f + (5*I + Sqrt[3])*h)*ArcT
an[((I + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 4*Sqrt[3]*(2*e - g + 2*i
)*ArcTan[Sqrt[3]/(1 + 2*x^2)])/36
```

Maple [A] (verified)

Time = 0.51 (sec) , antiderivative size = 214, normalized size of antiderivative = 1.10

method	result
default	$-\frac{\left(\frac{d}{3}-\frac{e}{3}-\frac{g}{3}+\frac{h}{3}-\frac{2f}{3}+\frac{2i}{3}\right)x-\frac{2d}{3}-\frac{e}{3}+\frac{2g}{3}+\frac{h}{3}+\frac{f}{3}-\frac{i}{3}}{4(x^2-x+1)}-\frac{(6d-3f+3h)\ln(x^2-x+1)}{24}-\frac{\left(-2d-4e-\frac{f}{2}+2g-\frac{h}{2}-4i\right)\sqrt{3}\ \arctan\left(\frac{(2x-1}{3}\right.}{18}$
risch	Expression too large to display

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/4*((1/3*d-1/3*e-1/3*g+1/3*h-2/3*f+2/3*i)*x-2/3*d-1/3*e+2/3*g+1/3*h+1/3*f
-1/3*i)/(x^2-x+1)-1/24*(6*d-3*f+3*h)*ln(x^2-x+1)-1/18*(-2*d-4*e-1/2*f+2*g-1
/2*h-4*i)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/4*((-1/3*d-1/3*e-1/3*g-1/3*
h+2/3*f+2/3*i)*x-2/3*d+1/3*e-2/3*g+1/3*h+1/3*f+1/3*i)/(x^2+x+1)+1/24*(6*d-3
*f+3*h)*ln(x^2+x+1)+1/18*(2*d-4*e+1/2*f+2*g+1/2*h-4*i)*arctan(1/3*(1+2*x)*3
^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 4.87 (sec) , antiderivative size = 279, normalized size of antiderivative = 1.44

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx = -\frac{12\,(d-2\,f+h)x^3-12\,(2\,e-g-i)x^2-2\,\sqrt{3}((4\,d-8\,e+f+4\,g+h-8\,i)x^4+(4\,d-8\,e+f+4\,g}{}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="frica
s")
```

```
[Out] -1/72*(12*(d - 2*f + h)*x^3 - 12*(2*e - g - i)*x^2 - 2*sqrt(3)*((4*d - 8*e
+ f + 4*g + h - 8*i)*x^4 + (4*d - 8*e + f + 4*g + h - 8*i)*x^2 + 4*d - 8*e
+ f + 4*g + h - 8*i)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*((4*d + 8*e
+ f - 4*g + h + 8*i)*x^4 + (4*d + 8*e + f - 4*g + h + 8*i)*x^2 + 4*d + 8*e
+ f - 4*g + h + 8*i)*arctan(1/3*sqrt(3)*(2*x - 1)) - 12*(d + f - 2*h)*x - 9
*((2*d - f + h)*x^4 + (2*d - f + h)*x^2 + 2*d - f + h)*log(x^2 + x + 1) + 9
*((2*d - f + h)*x^4 + (2*d - f + h)*x^2 + 2*d - f + h)*log(x^2 - x + 1) - 1
2*e + 24*g - 12*i)/(x^4 + x^2 + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 155, normalized size of antiderivative = 0.80

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx \\
&= \frac{1}{36}\sqrt{3}(4\,d-8\,e+f+4\,g+h-8\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) \\
&\quad+\frac{1}{36}\sqrt{3}(4\,d+8\,e+f-4\,g+h+8\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) \\
&\quad+\frac{1}{8}(2\,d-f+h)\log\left(x^2+x+1\right)-\frac{1}{8}(2\,d-f+h)\log\left(x^2-x+1\right) \\
&\quad-\frac{(d-2\,f+h)x^3-(2\,e-g-i)x^2-(d+f-2\,h)x-e+2\,g-i}{6\,(x^4+x^2+1)}
\end{aligned}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="maxim
a")
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g + h - 8*i)*arctan(1/3*sqrt(3)*(2*x + 1))
+ 1/36*sqrt(3)*(4*d + 8*e + f - 4*g + h + 8*i)*arctan(1/3*sqrt(3)*(2*x - 1)
) + 1/8*(2*d - f + h)*log(x^2 + x + 1) - 1/8*(2*d - f + h)*log(x^2 - x + 1)
 - 1/6*((d - 2*f + h)*x^3 - (2*e - g - i)*x^2 - (d + f - 2*h)*x - e + 2*g -
 i)/(x^4 + x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 165, normalized size of antiderivative = 0.85

$$\begin{aligned}&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx\\&=\frac{1}{36}\sqrt{3}(4\,d-8\,e+f+4\,g+h-8\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)\\&\quad+\frac{1}{36}\sqrt{3}(4\,d+8\,e+f-4\,g+h+8\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)\\&\quad+\frac{1}{8}\,(2\,d-f+h)\log\left(x^2+x+1\right)-\frac{1}{8}\,(2\,d-f+h)\log\left(x^2-x+1\right)\\&\quad-\frac{dx^3-2\,fx^3+hx^3-2\,ex^2+gx^2+ix^2-dx-fx+2\,hx-e+2\,g-i}{6\,(x^4+x^2+1)}\end{aligned}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^2,x, algorithm="giac"
)
```

```
[Out] 1/36*sqrt(3)*(4*d - 8*e + f + 4*g + h - 8*i)*arctan(1/3*sqrt(3)*(2*x + 1))
+ 1/36*sqrt(3)*(4*d + 8*e + f - 4*g + h + 8*i)*arctan(1/3*sqrt(3)*(2*x - 1)
) + 1/8*(2*d - f + h)*log(x^2 + x + 1) - 1/8*(2*d - f + h)*log(x^2 - x + 1)
 - 1/6*(d*x^3 - 2*f*x^3 + h*x^3 - 2*e*x^2 + g*x^2 + i*x^2 - d*x - f*x + 2*h
*x - e + 2*g - i)/(x^4 + x^2 + 1)
```

Mupad [B] (verification not implemented)

Time = 13.24 (sec) , antiderivative size = 1894, normalized size of antiderivative = 9.76

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^2}\,dx=\text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^2 + x^4 + 1)^2,x)
```

```
[Out] (e/6 - g/3 + i/6 + x*(d/6 + f/6 - h/3) - x^3*(d/6 - f/3 + h/6) - x^2*(g/6 -
 e/3 + i/6))/(x^2 + x^4 + 1) - log(60*d*g - 153*d*f - 120*d*e + 24*e*f + 13
5*d*h - 120*d*i - 48*e*h - 12*f*g - 81*f*h + 24*f*i + 24*g*h - 48*h*i + 3^(
1/2)*d^2*90i + 3^(1/2)*f^2*9i + 3^(1/2)*h^2*18i - 198*d^2*x - 36*f^2*x - 45
*h^2*x + 126*d^2 + 45*f^2 + 36*h^2 + 3^(1/2)*d*e*56i - 3^(1/2)*d*f*63i - 3^
(1/2)*d*g*28i - 3^(1/2)*e*f*40i + 3^(1/2)*d*h*81i + 3^(1/2)*d*i*56i + 3^(1/
2)*e*h*32i + 3^(1/2)*f*g*20i - 3^(1/2)*f*h*27i - 3^(1/2)*f*i*40i - 3^(1/2)*
g*h*16i + 3^(1/2)*h*i*32i - 24*d*e*x + 171*d*f*x + 12*d*g*x + 48*e*f*x - 18
9*d*h*x - 24*d*i*x - 24*e*h*x - 24*f*g*x + 81*f*h*x + 48*f*i*x + 12*g*h*x -
 24*h*i*x + 3^(1/2)*d^2*x*18i + 3^(1/2)*f^2*x*18i + 3^(1/2)*h^2*x*9i - 3^(1
```

```
/2)*d*f*x*45i + 3^(1/2)*d*g*x*44i + 3^(1/2)*e*f*x*32i + 3^(1/2)*d*h*x*27i -
 3^(1/2)*d*i*x*88i - 3^(1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h*x*
27i + 3^(1/2)*f*i*x*32i + 3^(1/2)*g*h*x*20i - 3^(1/2)*h*i*x*40i - 3^(1/2)*d
*e*x*88i)*(d/4 - f/8 + h/8 + (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9 + (3^(1/2
)*f*1i)/72 - (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/72 + (3^(1/2)*i*1i)/9) - lo
g(120*d*e - 153*d*f - 60*d*g - 24*e*f + 135*d*h + 120*d*i + 48*e*h + 12*f*g
 - 81*f*h - 24*f*i - 24*g*h + 48*h*i - 3^(1/2)*d^2*90i - 3^(1/2)*f^2*9i - 3
^(1/2)*h^2*18i + 198*d^2*x + 36*f^2*x + 45*h^2*x + 126*d^2 + 45*f^2 + 36*h^
2 + 3^(1/2)*d*e*56i + 3^(1/2)*d*f*63i - 3^(1/2)*d*g*28i - 3^(1/2)*e*f*40i -
 3^(1/2)*d*h*81i + 3^(1/2)*d*i*56i + 3^(1/2)*e*h*32i + 3^(1/2)*f*g*20i + 3^
(1/2)*f*h*27i - 3^(1/2)*f*i*40i - 3^(1/2)*g*h*16i + 3^(1/2)*h*i*32i - 24*d*
e*x - 171*d*f*x + 12*d*g*x + 48*e*f*x + 189*d*h*x - 24*d*i*x - 24*e*h*x - 2
4*f*g*x - 81*f*h*x + 48*f*i*x + 12*g*h*x - 24*h*i*x + 3^(1/2)*d^2*x*18i + 3
^(1/2)*f^2*x*18i + 3^(1/2)*h^2*x*9i - 3^(1/2)*d*f*x*45i - 3^(1/2)*d*g*x*44i
 - 3^(1/2)*e*f*x*32i + 3^(1/2)*d*h*x*27i + 3^(1/2)*d*i*x*88i + 3^(1/2)*e*h*
x*40i + 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h*x*27i - 3^(1/2)*f*i*x*32i - 3^(1/2)
*g*h*x*20i + 3^(1/2)*h*i*x*40i + 3^(1/2)*d*e*x*88i)*(f/8 - d/4 - h/8 + (3^(
1/2)*d*1i)/18 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 + (3^(1/2)*g*1i)/18 +
(3^(1/2)*h*1i)/72 - (3^(1/2)*i*1i)/9) + log(120*d*e - 153*d*f - 60*d*g - 24
*e*f + 135*d*h + 120*d*i + 48*e*h + 12*f*g - 81*f*h - 24*f*i - 24*g*h + 48*
h*i + 3^(1/2)*d^2*90i + 3^(1/2)*f^2*9i + 3^(1/2)*h^2*18i + 198*d^2*x + 36*f
^2*x + 45*h^2*x + 126*d^2 + 45*f^2 + 36*h^2 - 3^(1/2)*d*e*56i - 3^(1/2)*d*f
*63i + 3^(1/2)*d*g*28i + 3^(1/2)*e*f*40i + 3^(1/2)*d*h*81i - 3^(1/2)*d*i*56
i - 3^(1/2)*e*h*32i - 3^(1/2)*f*g*20i - 3^(1/2)*f*h*27i + 3^(1/2)*f*i*40i +
 3^(1/2)*g*h*16i - 3^(1/2)*h*i*32i - 24*d*e*x - 171*d*f*x + 12*d*g*x + 48*e
*f*x + 189*d*h*x - 24*d*i*x - 24*e*h*x - 24*f*g*x - 81*f*h*x + 48*f*i*x + 1
2*g*h*x - 24*h*i*x - 3^(1/2)*d^2*x*18i - 3^(1/2)*f^2*x*18i - 3^(1/2)*h^2*x*
9i + 3^(1/2)*d*f*x*45i + 3^(1/2)*d*g*x*44i + 3^(1/2)*e*f*x*32i - 3^(1/2)*d*
h*x*27i - 3^(1/2)*d*i*x*88i - 3^(1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i + 3^(1/
2)*f*h*x*27i + 3^(1/2)*f*i*x*32i + 3^(1/2)*g*h*x*20i - 3^(1/2)*h*i*x*40i -
3^(1/2)*d*e*x*88i)*(d/4 - f/8 + h/8 + (3^(1/2)*d*1i)/18 - (3^(1/2)*e*1i)/9
+ (3^(1/2)*f*1i)/72 + (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/72 - (3^(1/2)*i*1i
)/9) + log(120*d*e + 153*d*f - 60*d*g - 24*e*f - 135*d*h + 120*d*i + 48*e*h
 + 12*f*g + 81*f*h - 24*f*i - 24*g*h + 48*h*i + 3^(1/2)*d^2*90i + 3^(1/2)*f
^2*9i + 3^(1/2)*h^2*18i + 198*d^2*x + 36*f^2*x + 45*h^2*x - 126*d^2 - 45*f^
2 - 36*h^2 + 3^(1/2)*d*e*56i - 3^(1/2)*d*f*63i - 3^(1/2)*d*g*28i - 3^(1/2)*
e*f*40i + 3^(1/2)*d*h*81i + 3^(1/2)*d*i*56i + 3^(1/2)*e*h*32i + 3^(1/2)*f*g
*20i - 3^(1/2)*f*h*27i - 3^(1/2)*f*i*40i - 3^(1/2)*g*h*16i + 3^(1/2)*h*i*32
i + 24*d*e*x - 171*d*f*x - 12*d*g*x - 48*e*f*x + 189*d*h*x + 24*d*i*x + 24*
e*h*x + 24*f*g*x - 81*f*h*x - 48*f*i*x - 12*g*h*x + 24*h*i*x + 3^(1/2)*d^2*
x*18i + 3^(1/2)*f^2*x*18i + 3^(1/2)*h^2*x*9i - 3^(1/2)*d*f*x*45i + 3^(1/2)*
d*g*x*44i + 3^(1/2)*e*f*x*32i + 3^(1/2)*d*h*x*27i - 3^(1/2)*d*i*x*88i - 3^(
1/2)*e*h*x*40i - 3^(1/2)*f*g*x*16i - 3^(1/2)*f*h*x*27i + 3^(1/2)*f*i*x*32i
+ 3^(1/2)*g*h*x*20i - 3^(1/2)*h*i*x*40i - 3^(1/2)*d*e*x*88i)*(f/8 - d/4 - h
/8 + (3^(1/2)*d*1i)/18 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/72 - (3^(1/2)*g*
```

```
1i)/18 + (3^(1/2)*h*1i)/72 + (3^(1/2)*i*1i)/9)
```

3.36 $\int \frac{d+ex}{(a+bx^2+cx^4)^2} dx$

Optimal result

Integrand size = 20, antiderivative size = 330

$$\begin{aligned}\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx =& -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)}+\frac{dx(b^2-2ac+bcx^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} \\ &+\frac{\sqrt{c}\left(b^2-12ac+b\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} \\ &-\frac{\sqrt{c}\left(b^2-12ac-b\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} \\ &+\frac{2ce\,\mathrm{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}\end{aligned}$$

```
[Out] -1/2*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*d*x*(b*c*x^2-2*a*c+b^2)
/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+2*c*e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2
))/(-4*a*c+b^2)^(3/2)+1/4*d*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))
^(1/2))*c^(1/2)*(b^2-12*a*c+b*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)^(3/2)*2^(1
/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)-1/4*d*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+
b^2)^(1/2))^(1/2))*c^(1/2)*(b^2-12*a*c-b*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)
^(3/2)*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.50 (sec) , antiderivative size = 330, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.450$, Rules used = {1687, 12, 1106, 1180, 211, 1121, 628, 632, 212}

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \frac{\sqrt{c}d\left(b\sqrt{b^2-4ac}-12ac+b^2\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{c}d\left(-b\sqrt{b^2-4ac}-12ac+b^2\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{\sqrt{b^2-4ac}+b}} + \frac{2ce\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{3/2}} + \frac{dx\left(-2ac+b^2+bcx^2\right)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{e\left(b+2cx^2\right)}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}$$

```
[In] Int[(d + e*x)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] -1/2*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (d*x*(b^2 - 2*
a*c + b*c*x^2))/(2*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt[c]*(b^2 - 1
2*a*c + b*Sqrt[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2
 - 4*a*c]]])/(2*Sqrt[2]*a*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]])
- (Sqrt[c]*(b^2 - 12*a*c - b*Sqrt[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x
)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^2 - 4*a*c)^(3/2)*Sqrt[b + S
qrt[b^2 - 4*a*c]]) + (2*c*e*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2
- 4*a*c)^(3/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d}{(a+bx^2+cx^4)^2}\,dx + \int \frac{ex}{(a+bx^2+cx^4)^2}\,dx$$

$$= d\int \frac{1}{\left(a+bx^2+cx^4\right)^2}\,dx + e\int \frac{x}{\left(a+bx^2+cx^4\right)^2}\,dx$$

$$= \frac{dx(b^2-2ac+bcx^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{d\int \frac{b^2-2ac-2\left(b^2-4ac\right)-bcx^2}{a+bx^2+cx^4}\,dx}{2a\left(b^2-4ac\right)} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{\left(a+bx+cx^2\right)^2}\,dx, x, x^2\right)$$

$$= -\frac{e(b+2cx^2)}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{dx(b^2-2ac+bcx^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{\left(c\left(b^2-12ac-b\sqrt{b^2-4ac}\right)d\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)^{3/2}} + \frac{\left(c\left(b^2-12ac+b\sqrt{b^2-4ac}\right)d\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)^{3/2}} - \frac{(ce)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{dx(b^2-2ac+bcx^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\sqrt{c}\left(b^2-12ac+b\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{c}\left(b^2-12ac-b\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{dx(b^2-2ac+bcx^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\sqrt{c}\left(b^2-12ac+b\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{\sqrt{c}\left(b^2-12ac-b\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{2ce\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{3/2}}$$

Mathematica [A] (verified)

Time = 0.50 (sec) , antiderivative size = 341, normalized size of antiderivative = 1.03

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \frac{1}{4}\left(\frac{2abe+4acx(d+ex)-2bdx(b+cx^2)}{a(-b^2+4ac)(a+bx^2+cx^4)} + \frac{\sqrt{2}\sqrt{c}\left(b^2-12ac+b\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{2}\sqrt{c}\left(-b^2+12ac+b\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{4ce\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{3/2}} + \frac{4ce\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{3/2}}\right)$$

```
[In] Integrate[(d + e*x)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((2*a*b*e + 4*a*c*x*(d + e*x) - 2*b*d*x*(b + c*x^2))/(a*(-b^2 + 4*a*c)*(a +
 b*x^2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(b^2 - 12*a*c + b*Sqrt[b^2 - 4*a*c])*d*
ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(
3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[2]*Sqrt[c]*(-b^2 + 12*a*c + b*Sqr
t[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/
(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (4*c*e*Log[-b + Sqrt[
b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) + (4*c*e*Log[b + Sqrt[b^2 - 4*
a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.25 (sec) , antiderivative size = 209, normalized size of antiderivative = 0.63

method	result
risch	$\frac{-\frac{bcd\,x^3}{2a(4ac-b^2)}+\frac{c\,x^2e}{4ac-b^2}+\frac{(2ac-b^2)dx}{2a(4ac-b^2)}+\frac{be}{8ac-2b^2}}{c\,x^4+b\,x^2+a}+\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2b+a\right)}\frac{\left(-\frac{c\_R^2bd}{a(4ac-b^2)}+\frac{4ce\_R}{4ac-b^2}+\frac{d(6ac-b^2)}{a(4ac-b^2)}\right)\ln\left(x-\_R\right.}{2c\_R^3+\_Rb}\right.}{4}$
default	$16c^2\left(-\frac{\frac{\frac{\left(4ac\sqrt{-4ac+b^2}-b^2\sqrt{-4ac+b^2}+4abc-b^3\right)dx}{16a\,c^2}-\frac{e(4ac-b^2)}{8c^2}}{x^2+\frac{b}{2c}-\frac{\sqrt{-4ac+b^2}}{2c}}+\frac{2ae\sqrt{-4ac+b^2}\,\ln\left(-2c\,x^2+\sqrt{-4ac+b^2}-b\right)+\frac{\left(-12acd\sqrt{-4ac+b^2}+b^2d\sqrt{\right.}{}}{8ac}}{4(4ac-b^2)^2}\right.$

```
[In] int((e*x+d)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (-1/2/a*b*c*d/(4*a*c-b^2)*x^3+c/(4*a*c-b^2)*x^2*e+1/2*(2*a*c-b^2)*d/a/(4*a*
c-b^2)*x+1/2/(4*a*c-b^2)*b*e)/(c*x^4+b*x^2+a)+1/4*sum((-c/a/(4*a*c-b^2)*_R^
2*b*d+4*c/(4*a*c-b^2)*e*_R+d*(6*a*c-b^2)/a/(4*a*c-b^2))/(2*_R^3*c+_R*b)*ln(
x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 235.03 (sec) , antiderivative size = 1678440, normalized size of antiderivative = 5086.18

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((e*x+d)/(c*x**4+b*x**2+a)**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \int \frac{ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
[Out] 1/2*(b*c*d*x^3 - 2*a*c*e*x^2 - a*b*e + (b^2 - 2*a*c)*d*x)/((a*b^2*c - 4*a^2
*c^2)*x^4 + a^2*b^2 - 4*a^3*c + (a*b^3 - 4*a^2*b*c)*x^2) + 1/2*integrate((b
*c*d*x^2 - 4*a*c*e*x + (b^2 - 6*a*c)*d)/(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*
a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3426 vs. $2(278) = 556$.

Time = 1.34 (sec) , antiderivative size = 3426, normalized size of antiderivative = 10.38

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="giac")
[Out] 1/2*(b*c*d*x^3 - 2*a*c*e*x^2 + b^2*d*x - 2*a*c*d*x - a*b*e)/((c*x^4 + b*x^2
 + a)*(a*b^2 - 4*a^2*c)) - 1/16*((2*b^3*c^2 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b^2 - 4*a^2*c)^2*d - 2*(s
qrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*
c - 2*a*b^6*c + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 20
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b
```

```
*c^3 - 10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 128*a^3*b^2
*c^3 + 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 192*a^4*c^4 + 2
*(b^2 - 4*a*c)*a*b^4*c - 20*(b^2 - 4*a*c)*a^2*b^2*c^2 + 48*(b^2 - 4*a*c)*a^
3*c^3)*d*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b
^3*c^4 - 384*a^5*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^2*b^7 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^3*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^2*b^6*c - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^4*b^3*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^3*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2
*b^5*c^2 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
5*b*c^3 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*
b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*
b^3*c^3 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*
b*c^4 - 2*(b^2 - 4*a*c)*a^2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^
2 - 4*a*c)*a^4*b*c^4)*d)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c + sqr
t((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*
b^2*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 48*a^5*b^2*c^
2 + 16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 - 8*a^4*b^2*c^
3 + 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/16*((2*b^3*c^2 - 8*a*b*c^3
 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b^2 -
 4*a^2*c)^2*d - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a*b^5*c + 2*a*b^6*c + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^3*b^2*c^2 + 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 +
sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 28*a^2*b^4*c^2 - 96*sqr
t(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^3*b*c^3 - 10*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b
^2*c^3 + 128*a^3*b^2*c^3 + 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c
^4 - 192*a^4*c^4 - 2*(b^2 - 4*a*c)*a*b^4*c + 20*(b^2 - 4*a*c)*a^2*b^2*c^2 -
 48*(b^2 - 4*a*c)*a^3*c^3)*d*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^2 - 40*a^3
*b^5*c^3 + 224*a^4*b^3*c^4 - 384*a^5*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^2*b^5*c^2 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^5*b*c^3 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b^5*c^2 + 32*(b^2 - 4*a*c)*
```

```
a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*a^4*b*c^4)*d)*arctan(2*sqrt(1/2)*x/sqrt((a*b
^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*
c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b
^5*c + 48*a^5*b^2*c^2 + 16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*
b*c^3 - 8*a^4*b^2*c^3 + 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/4*((b^
3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)
*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (a*b^5*c^2 - 8*a^2*b^3*c^3 - 2
*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^
4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e)*log(
x^2 + 1/2*(a*b^3 - 4*a^2*b*c + sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*
a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c
 - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*
b^2 - 4*a^2*c)) - 1/4*((b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2
- 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (a*b
^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3
*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*
sqrt(b^2 - 4*a*c))*e)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^
2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c
^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c
^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 8.37 (sec) , antiderivative size = 2382, normalized size of antiderivative = 7.22

$$\int \frac{d+ex}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] ((b*e)/(2*(4*a*c - b^2)) + (c*e*x^2)/(4*a*c - b^2) + (d*x*(2*a*c - b^2))/(2
*a*(4*a*c - b^2)) - (b*c*d*x^3)/(2*a*(4*a*c - b^2)))/(a + b*x^2 + c*x^4) +
symsum(log((5*b^3*c^4*d^3 - 96*a^2*c^5*d*e^2 - 36*a*b*c^5*d^3 + 16*a*b^2*c^
4*d*e^2)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - root(
1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 -
 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^
3*b^12*z^4 + 61440*a^5*b*c^5*d^2*z^2 + 432*a*b^9*c*d^2*z^2 + 24576*a^5*b^2*
c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^
4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 -
32768*a^6*c^5*e^2*z^2 - 16*b^11*d^2*z^2 - 672*a*b^6*c^2*d^2*e*z - 15872*a^3
*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z + 18432*a^4*c^5*d^2*e*z + 32*b^
8*c*d^2*e*z - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 - 16*b^5*c^2*d^
2*e^2 + 360*a*b^2*c^4*d^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5
*d^4, z, k)*(root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680
*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^
```

```
9*c^6*z^4 - 256*a^3*b^12*z^4 + 61440*a^5*b*c^5*d^2*z^2 + 432*a*b^9*c*d^2*z^
2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*
e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*
b^7*c^2*d^2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*b^11*d^2*z^2 - 672*a*b^6*c^2*d
^2*e*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z + 18432*a^4*c
^5*d^2*e*z + 32*b^8*c*d^2*e*z - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e
^2 - 16*b^5*c^2*d^2*e^2 + 360*a*b^2*c^4*d^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*
d^4 - 1296*a^2*c^5*d^4, z, k)*((x*(1024*a^5*c^6*e - 16*a^2*b^6*c^3*e + 192*
a^3*b^4*c^4*e - 768*a^4*b^2*c^5*e))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c
 + 48*a^4*b^2*c^2)) - (6144*a^5*c^6*d - 288*a^2*b^6*c^3*d + 1920*a^3*b^4*c^
4*d - 5632*a^4*b^2*c^5*d + 16*a*b^8*c^2*d)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^
3*b^4*c + 48*a^4*b^2*c^2)) + (root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4
*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c
*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 61440*a^5*b*c^5*d^2*z^2 + 4
32*a*b^9*c*d^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 +
 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^
2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*b^11*d^2*z^2
 - 672*a*b^6*c^2*d^2*e*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*
e*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 960*a^2*b*c^4*d^2*e^2 + 24
0*a*b^3*c^3*d^2*e^2 - 16*b^5*c^2*d^2*e^2 + 360*a*b^2*c^4*d^4 - 256*a^3*c^4*
e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*x*(4096*a^6*b*c^6 + 16*a^2*b
^9*c^2 - 256*a^3*b^7*c^3 + 1536*a^4*b^5*c^4 - 4096*a^5*b^3*c^5))/(2*(a^2*b^
6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) + (32*a*b^5*c^3*d*e + 102
4*a^3*b*c^5*d*e - 384*a^2*b^3*c^4*d*e)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^
4*c + 48*a^4*b^2*c^2)) - (x*(288*a^3*c^6*d^2 - b^6*c^3*d^2 + 18*a*b^4*c^4*d
^2 - 64*a^3*b*c^5*e^2 - 128*a^2*b^2*c^5*d^2 + 16*a^2*b^3*c^4*e^2))/(2*(a^2*
b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) - (x*(16*a^2*c^5*e^3 -
b^3*c^4*d^2*e + 12*a*b*c^5*d^2*e))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c
+ 48*a^4*b^2*c^2)))*root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 +
 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 104
8576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 61440*a^5*b*c^5*d^2*z^2 + 432*a*b^9*c
*d^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b
^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 46
08*a^2*b^7*c^2*d^2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*b^11*d^2*z^2 - 672*a*b^
6*c^2*d^2*e*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z + 1843
2*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^
3*d^2*e^2 - 16*b^5*c^2*d^2*e^2 + 360*a*b^2*c^4*d^4 - 256*a^3*c^4*e^4 - 25*b
^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k), k, 1, 4)
```

3.37 $\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2} \, dx$

Optimal result

Integrand size = 25, antiderivative size = 368

$$
\begin{aligned}
\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2} \, dx = & -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} \\
& + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} \\
& + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} \\
& + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} \\
& + \frac{2ce\,\mathrm{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}
\end{aligned}
$$

```
[Out] -1/2*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*x*(b^2*d-2*a*c*d-a*b*f+
c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+2*c*e*arctanh((2*c*x^2+b
)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(3/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-
4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b*d-2*a*f+(4*a*b*f-12*a*c*d+b^2*d)/(-4*a*
c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/4*arcta
n(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b*d-2*a*f+(-4*a*
b*f+12*a*c*d-b^2*d)/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/(b+(-4*a*c+b
^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.56 (sec) , antiderivative size = 368, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.360, Rules used = {1687, 1192, 1180, 211, 12, 1121, 628, 632, 212}

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx = \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}} + \frac{2ce\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}} + \frac{x(cx^2(bd-2af)-abf-2acd+b^2d)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)}$$

```
[In] Int[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] -1/2*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (x*(b^2*d - 2*
a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(2*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4
)) + (Sqrt[c]*(b*d - 2*a*f + (b^2*d - 12*a*c*d + 4*a*b*f)/Sqrt[b^2 - 4*a*c]
)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^
2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(b*d - 2*a*f - (b^2*d -
12*a*c*d + 4*a*b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^2 - 4*a*c)*Sqrt[b + Sqrt[b^2 - 4*a*c]]
) + (2*c*e*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(3/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{ex}{(a+bx^2+cx^4)^2}\,dx + \int \frac{d+fx^2}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{\int \frac{-b^2d+6acd-abf-c(bd-2af)x^2}{a+bx^2+cx^4}\,dx}{2a(b^2-4ac)} + e\int \frac{x}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right) + \frac{\left(c\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)} + \frac{\left(c\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(ce)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{2ce\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$

Mathematica [A] (verified)

Time = 0.72 (sec) , antiderivative size = 398, normalized size of antiderivative = 1.08

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx$$
$$= \frac{1}{4}\left(\frac{2ab(e+fx)-2bdx(b+cx^2)+4acx(d+x(e+fx))}{a(-b^2+4ac)(a+bx^2+cx^4)}\right.$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(b^2d+b\left(\sqrt{b^2-4ac}d+4af\right)-2a\left(6cd+\sqrt{b^2-4ac}f\right)\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(-b^2d+12acd+b\sqrt{b^2-4ac}d-4abf-2a\sqrt{b^2-4ac}f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$\left. - \frac{4ce\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{3/2}} + \frac{4ce\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{3/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((2*a*b*(e + f*x) - 2*b*d*x*(b + c*x^2) + 4*a*c*x*(d + x*(e + f*x)))/(a*(-b
^2 + 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(b^2*d + b*(Sqrt[b^2 -
4*a*c]*d + 4*a*f) - 2*a*(6*c*d + Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt
[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b
^2 - 4*a*c]]) + (Sqrt[2]*Sqrt[c]*(-(b^2*d) + 12*a*c*d + b*Sqrt[b^2 - 4*a*c]
*d - 4*a*b*f - 2*a*Sqrt[b^2 - 4*a*c]*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
 Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) -
 (4*c*e*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) + (4*c*e
*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.26 (sec) , antiderivative size = 232, normalized size of antiderivative = 0.63

method	result
risch	$\frac{\frac{c(2af-bd)x^3}{2a(4ac-b^2)}+\frac{c\,x^2e}{4ac-b^2}+\frac{(abf+2acd-b^2d)x}{2a(4ac-b^2)}+\frac{be}{8ac-2b^2}}{c\,x^4+b\,x^2+a}+\frac{\left(\sum_{\_R=\mathrm{RootOf}(c\_Z^4+\_Z^2b+a)}\frac{\left(\frac{c(2af-bd)\_R^2}{a(4ac-b^2)}+\frac{4ce\_R}{4ac-b^2}-\frac{abf-6acd+b^2d}{a(4ac-b^2)}\right.}{2c\_R^3+\_Rb}\right.}{4}$
default	$16c^2\left(-\frac{-\frac{\left(-4acd\sqrt{-4ac+b^2}+b^2d\sqrt{-4ac+b^2}+8a^2cf-2a\,b^2f-4abcd+b^3d\right)x}{16ac}-\frac{e(4ac-b^2)}{8c}}{x^2+\frac{b}{2c}-\frac{\sqrt{-4ac+b^2}}{2c}}+\frac{2ae\sqrt{-4ac+b^2}\,\ln\left(-2c\,x^2+\sqrt{-4ac+b^2}-b\right)+\cdots}{\cdots}}{4c(4ac-b^2)^2}\right.$

```
[In] int((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (1/2*c*(2*a*f-b*d)/a/(4*a*c-b^2)*x^3+c/(4*a*c-b^2)*x^2*e+1/2*(a*b*f+2*a*c*d
-b^2*d)/a/(4*a*c-b^2)*x+1/2/(4*a*c-b^2)*b*e)/(c*x^4+b*x^2+a)+1/4*sum((c*(2*
a*f-b*d)/a/(4*a*c-b^2)*_R^2+4*c/(4*a*c-b^2)*e*_R-(a*b*f-6*a*c*d+b^2*d)/a/(4
*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(c*x**4+b*x**2+a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx = \int \frac{fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] -1/2*(2*a*c*e*x^2 - (b*c*d - 2*a*c*f)*x^3 + a*b*e + (a*b*f - (b^2 - 2*a*c)*
d)*x)/((a*b^2*c - 4*a^2*c^2)*x^4 + a^2*b^2 - 4*a^3*c + (a*b^3 - 4*a^2*b*c)*
x^2) - 1/2*integrate((4*a*c*e*x - a*b*f - (b*c*d - 2*a*c*f)*x^2 - (b^2 - 6*
a*c)*d)/(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5156 vs. $2(320) = 640$.

Time = 1.74 (sec) , antiderivative size = 5156, normalized size of antiderivative = 14.01

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/2*(b*c*d*x^3 - 2*a*c*f*x^3 - 2*a*c*e*x^2 + b^2*d*x - 2*a*c*d*x - a*b*f*x
- a*b*e)/((c*x^4 + b*x^2 + a)*(a*b^2 - 4*a^2*c)) + 1/16*((2*b^3*c^2 - 8*a*b
*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b
^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^2*c)^2*f + 2*(sqrt(2)*
```

```
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 2*a
*b^6*c + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 20*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a*b^4*c^2 + 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 -
10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 128*a^3*b^2*c^3 +
24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 192*a^4*c^4 + 2*(b^2 -
 4*a*c)*a*b^4*c - 20*(b^2 - 4*a*c)*a^2*b^2*c^2 + 48*(b^2 - 4*a*c)*a^3*c^3)*
d*abs(a*b^2 - 4*a^2*c) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5
 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*a^2*b^5*c + 16*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3
*b^2*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*a^3*b^3
*c^2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 32*a^4*b*c^3 +
 2*(b^2 - 4*a*c)*a^2*b^3*c - 8*(b^2 - 4*a*c)*a^3*b*c^2)*f*abs(a*b^2 - 4*a^2
*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c^4 - 384*a^5*b*c^5 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^7 + 20*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c + 2*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c - 112*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - 32*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 - sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^2 + 192*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*c^3 + 96*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 16*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 48*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^
2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*a^4*b*c^4)*d +
4*(2*a^3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^3*b^4*c^2 + 8*(b^2 - 4
*a*c)*a^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c + sqrt((
a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2
*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 48*a^5*b^2*c^2 +
 16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 - 8*a^4*b^2*c^3 +
 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/16*((2*b^3*c^2 - 8*a*b*c^3 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b^2 - 4*
```

a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^2*c)^2*f - 2*(sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*a*b^6*c
+ 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 20*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^4*c^2 - 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 10*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 128*a^3*b^2*c^3 + 24*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 192*a^4*c^4 - 2*(b^2 - 4*a*c)
*a*b^4*c + 20*(b^2 - 4*a*c)*a^2*b^2*c^2 - 48*(b^2 - 4*a*c)*a^3*c^3)*d*abs(a
*b^2 - 4*a^2*c) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 - 8*sq
rt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*a^2*b^5*c + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^
2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 16*a^3*b^3*c^2 -
4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 32*a^4*b*c^3 - 2*(b^2
 - 4*a*c)*a^2*b^3*c + 8*(b^2 - 4*a*c)*a^3*b*c^2)*f*abs(a*b^2 - 4*a^2*c) + (
2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c^4 - 384*a^5*b*c^5 - sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7 + 20*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c + 2*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c - 112*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - 32*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 - sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^2 + 192*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^3 + 96*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 48*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b^5*c
^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*a^4*b*c^4)*d + 4*(2*a^
3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^3*b^4*c^2 + 8*(b^2 - 4*a*c)*a
^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c - sqrt((a*b^3 -
 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*
a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 48*a^5*b^2*c^2 + 16*a^4
*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 - 8*a^4*b^2*c^3 + 16*a^5

```
*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/4*((b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3
+ b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^
2 - 4*a^2*c) - (a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*
a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^
3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c +
 sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))
/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*
a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c)) - 1/4*((b^3*c^
2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqr
t(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b
^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^
2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e)*log(x^2
+ 1/2*(a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*
c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2
*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2
- 4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 8.40 (sec) , antiderivative size = 4707, normalized size of antiderivative = 12.79

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] symsum(log((5*b^3*c^4*d^3 + 8*a^3*c^4*f^3 - 96*a^2*c^5*d*e^2 + 72*a^2*c^5*d
^2*f - 3*b^4*c^3*d^2*f + 6*a^2*b^2*c^3*f^3 - 36*a*b*c^5*d^3 + 16*a*b^2*c^4*
d*e^2 + 18*a*b^2*c^4*d^2*f + 3*a*b^3*c^3*d*f^2 - 60*a^2*b*c^4*d*f^2 + 16*a^
2*b*c^4*e^2*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) -
 root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3
*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 -
256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072
*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 +
 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 81
92*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z
^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4
*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10
*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4
096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768
*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*
a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z +
 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z +
32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3
*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*
```

c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*((32*a*b^5*c^3*d*e - 512*a^4*c^5*e*f + 1024*a^3*b*c^5*d*e - 384*a^2*b^3*c^4*d*e + 32*a^2*b^4*c^3*e*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*((x*(1024*a^5*c^6*e - 16*a^2*b^6*c^3*e + 192*a^3*b^4*c^4*e - 768*a^4*b^2*c^5*e))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - (6144*a^5*c^6*d - 288*a^2*b^6*c^3*d + 1920*a^3*b^4*c^4*d - 5632*a^4*b^2*c^5*d + 16*a^2*b^7*c^2*f - 192*a^3*b^5*c^3*f + 768*a^4*b^3*c^4*f + 16*a*b^8*c^2*d - 1024*a^5*b*c^5*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2* c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*x*(4096*a^6*b*c^6 + 16*a^2*b^9*c^2 - 256*a^3*b^7*c^3 + 1536*a^4*b^5*c^4 - 4096*a^5*b^3*c^5))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) + (x*(b^6*c^3*d^2 - 288*a^3*c^6*d^2 + 32*a^4*c^5*f^2 - 18*a*b^4*c^4*d^2 + 64*a^3*b*c^5*e^2 + 128*a^2*b^2*c^5*d^2 - 16*a^2*b^3*c^4*e^2 + 10*a^2*b^4*c^3*f^2 - 48*a^3*b^2*c^4*f^2 + 2*a*b^5*c^3*d*f + 160*a^3*b*c^5*d*f - 48*a^2*b^3*c^4*d*f))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) - (x*(16*a^2*c^5*e^3 - b^3*c^4*d^2*e + 12*a*b*c^5*d^2*e - 24*a^2*c^5*d*e*f + 8*a^2*b*c^4*e*f^2 - 2*a*b^2*c^4*d*e*f))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)))*root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k), k, 1, 4) + ((b*e)/(2*(4*a*c - b^2)) + (c*e*x^2)/(4*a*c - b^2) + (x*(2*a*c*d - b^2*d + a*b*f))/(2*a*(4*a*c - b^2)) - (c*x^3*(b*d - 2*a*f))/(2*a*(4*a*c - b^2)))/(a + b*x^2 + c*x^4)

3.38 $\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx$

Optimal result

Integrand size = 30, antiderivative size = 386

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx = \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{be-2ag+(2ce-bg)x^2}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$

```
[Out] 1/2*x*(b^2*d-2*a*c*d-a*b*f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+
a)+1/2*(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+(-b*g+2*c
*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(3/2)+1/4*arctan(x
*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b*d-2*a*f+(4*a*b*f-
12*a*c*d+b^2*d)/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/(b-(-4*a*c+b^2)^
(1/2))^(1/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(
1/2)*(b*d-2*a*f+(-4*a*b*f+12*a*c*d-b^2*d)/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2
)*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.26 (sec) , antiderivative size = 386, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.267, Rules used = {1687, 1192, 1180, 211, 1261, 652, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx = \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}} + \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}} + \frac{x(cx^2(bd-2af)-abf-2acd+b^2d)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{-2ag+x^2(2ce-bg)+be}{2(b^2-4ac)(a+bx^2+cx^4)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] (x*(b^2*d - 2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(2*a*(b^2 - 4*a*c)*(a +
 b*x^2 + c*x^4)) - (b*e - 2*a*g + (2*c*e - b*g)*x^2)/(2*(b^2 - 4*a*c)*(a +
b*x^2 + c*x^4)) + (Sqrt[c]*(b*d - 2*a*f + (b^2*d - 12*a*c*d + 4*a*b*f)/Sqrt
[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*
Sqrt[2]*a*(b^2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(b*d - 2*a*
f - (b^2*d - 12*a*c*d + 4*a*b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]
*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^2 - 4*a*c)*Sqrt[b + Sqrt[
b^2 - 4*a*c]]) + ((2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(
b^2 - 4*a*c)^(3/2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
```

```
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2}{(a+bx^2+cx^4)^2}\,dx + \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{\left(a+bx+cx^2\right)^2}\,dx,x,x^2\right) - \frac{\int \frac{-b^2d+6acd-abf-c(bd-2af)x^2}{a+bx^2+cx^4}\,dx}{2a\left(b^2-4ac\right)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{be-2ag+(2ce-bg)x^2}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\left(c\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)} + \frac{\left(c\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)} - \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{be-2ag+(2ce-bg)x^2}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{b^2-4ac}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{be-2ag+(2ce-bg)x^2}{2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\left(b^2-4ac\right)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{3/2}}$$

Mathematica [A] (verified)

Time = 0.74 (sec) , antiderivative size = 421, normalized size of antiderivative = 1.09

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx = \frac{1}{4}\left(\frac{-4a^2g-2bdx(b+cx^2)+4acx(d+x(e+fx))+2ab(e+x(f-gx))}{a(-b^2+4ac)(a+bx^2+cx^4)} + \frac{\sqrt{2}\sqrt{c}\left(b^2d+b\left(\sqrt{b^2-4ac}\,d+4af\right)-2a\left(6cd+\sqrt{b^2-4ac}\,f\right)\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{2}\sqrt{c}\left(-b^2d+12acd+b\sqrt{b^2-4ac}\,d-4abf-2a\sqrt{b^2-4ac}\,f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} + \frac{2(-2ce+bg)\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{3/2}} - \frac{2(-2ce+bg)\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{3/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((-4*a^2*g - 2*b*d*x*(b + c*x^2) + 4*a*c*x*(d + x*(e + f*x)) + 2*a*b*(e + x
*(f - g*x)))/(a*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(b^2
*d + b*(Sqrt[b^2 - 4*a*c]*d + 4*a*f) - 2*a*(6*c*d + Sqrt[b^2 - 4*a*c]*f))*A
rcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3
/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[2]*Sqrt[c]*(-(b^2*d) + 12*a*c*d +
b*Sqrt[b^2 - 4*a*c]*d - 4*a*b*f - 2*a*Sqrt[b^2 - 4*a*c]*f)*ArcTan[(Sqrt[2]*
Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sq
rt[b^2 - 4*a*c]]) + (2*(-2*c*e + b*g)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2]
)/(b^2 - 4*a*c)^(3/2) - (2*(-2*c*e + b*g)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x
^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.28 (sec) , antiderivative size = 252, normalized size of antiderivative = 0.65

method	result
risch	$\frac{\frac{c(2af-bd)x^3}{2a(4ac-b^2)}-\frac{(bg-2ec)x^2}{2(4ac-b^2)}+\frac{(abf+2acd-b^2d)x}{2a(4ac-b^2)}-\frac{2ag-be}{2(4ac-b^2)}}{cx^4+bx^2+a}+\frac{\left(\sum_{\_R=\text{RootOf}(c\_Z^4+\_Z^2b+a)}\frac{\left(\frac{c(2af-bd)\_R^2}{a(4ac-b^2)}-\frac{2(bg-2ec)\_R}{4ac-b^2}-\frac{ab}{}\right.}{2c\_R^3+\_R}\right.}{4}$
default	$16c^2\left(-\frac{-\frac{\left(-4acd\sqrt{-4ac+b^2}+b^2d\sqrt{-4ac+b^2}+8a^2cf-2ab^2f-4abcd+b^3d\right)x}{16ac}-\frac{4\sqrt{-4ac+b^2}\,acg-\sqrt{-4ac+b^2}\,b^2g-4abgc+8ac^2e+b^3g-2b^2ce}{16c^2}}{x^2+\frac{b}{2c}-\frac{\sqrt{-4ac+b^2}}{2c}}\right.$

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (1/2*c*(2*a*f-b*d)/a/(4*a*c-b^2)*x^3-1/2*(b*g-2*c*e)/(4*a*c-b^2)*x^2+1/2*(a
*b*f+2*a*c*d-b^2*d)/a/(4*a*c-b^2)*x-1/2*(2*a*g-b*e)/(4*a*c-b^2))/(c*x^4+b*x
^2+a)+1/4*sum((c*(2*a*f-b*d)/a/(4*a*c-b^2)*_R^2-2*(b*g-2*c*e)/(4*a*c-b^2)*_
R-(a*b*f-6*a*c*d+b^2*d)/a/(4*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_
Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx=\int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] 1/2*((b*c*d - 2*a*c*f)*x^3 - a*b*e + 2*a^2*g - (2*a*c*e - a*b*g)*x^2 - (a*b
*f - (b^2 - 2*a*c)*d)*x)/((a*b^2*c - 4*a^2*c^2)*x^4 + a^2*b^2 - 4*a^3*c + (
a*b^3 - 4*a^2*b*c)*x^2) - 1/2*integrate(-(a*b*f + (b*c*d - 2*a*c*f)*x^2 + (
b^2 - 6*a*c)*d - 2*(2*a*c*e - a*b*g)*x)/(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*
a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5573 vs. 2(338) = 676.

Time = 1.79 (sec) , antiderivative size = 5573, normalized size of antiderivative = 14.44

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx=\text{Too large to display}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/2*(b*c*d*x^3 - 2*a*c*f*x^3 - 2*a*c*e*x^2 + a*b*g*x^2 + b^2*d*x - 2*a*c*d*
x - a*b*f*x - a*b*e + 2*a^2*g)/((c*x^4 + b*x^2 + a)*(a*b^2 - 4*a^2*c)) + 1/
16*((2*b^3*c^2 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*
c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^
2 - 4*a*c)*b*c^2)*(a*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^
2*c)^2*f + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a*b^5*c - 2*a*b^6*c + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^3*b^2*c^2 + 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(
2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 28*a^2*b^4*c^2 - 96*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^3*b*c^3 - 10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^
3 - 128*a^3*b^2*c^3 + 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 +
192*a^4*c^4 + 2*(b^2 - 4*a*c)*a*b^4*c - 20*(b^2 - 4*a*c)*a^2*b^2*c^2 + 48*(
```

b^2 - 4*a*c)*a^3*c^3)*d*abs(a*b^2 - 4*a^2*c) + 2*(sqrt(2)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^2*b^5 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*
c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*a^2*b^5*c + 16*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
2*b^3*c^2 + 16*a^3*b^3*c^2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*
b*c^3 - 32*a^4*b*c^3 + 2*(b^2 - 4*a*c)*a^2*b^3*c - 8*(b^2 - 4*a*c)*a^3*b*c^
2)*f*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c
^4 - 384*a^5*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^2*b^7 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^3*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2
*b^6*c - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*
b^3*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*
b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5
*c^2 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*
c^3 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*
c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*
c^3 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^
4 - 2*(b^2 - 4*a*c)*a^2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 -
4*a*c)*a^4*b*c^4)*d + 4*(2*a^3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^
3*b^4*c^2 + 8*(b^2 - 4*a*c)*a^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^
3 - 4*a^2*b*c + sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c
 - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^
5*c + 48*a^5*b^2*c^2 + 16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b
*c^3 - 8*a^4*b^2*c^3 + 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) + 1/16*((2*
b^3*c^2 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*
c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c - sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a
*c)*b*c^2)*(a*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^2*c)^2*
f + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*b^5*c + 2*a*b^6*c + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*
c^2 + 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt

(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^3*b*c^3 - 10*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 128
*a^3*b^2*c^3 + 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 192*a^4
c^4 - 2(b^2 - 4*a*c)*a*b^4*c + 20*(b^2 - 4*a*c)*a^2*b^2*c^2 - 48*(b^2 - 4
*a*c)*a^3*c^3)*d*abs(a*b^2 - 4*a^2*c) + 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^2*b^5 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*a^2*b^5*c + 16*sqrt(2)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c
^2 - 16*a^3*b^3*c^2 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 +
32*a^4*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c + 8*(b^2 - 4*a*c)*a^3*b*c^2)*f*ab
s(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c^4 - 38
4*a^5*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2
*b^7 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5
*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c
- 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2
- 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^2 +
192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^3 + 9
6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 1
6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 4
8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(
b^2 - 4*a*c)*a^2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*
a^4*b*c^4)*d + 4*(2*a^3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 - sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^3*b^4*c
^2 + 8*(b^2 - 4*a*c)*a^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a
^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^
2*c^2)))/(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 4
8*a^5*b^2*c^2 + 16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 -
8*a^4*b^2*c^3 + 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/8*(2*(b^3*c^2
- 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 - (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(
b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (b^4*c - 4*a*b^2*c^2 - 2*b^3*c^2 + b
^2*c^3 + (b^3*c - 4*a*b*c^2 - 2*b^2*c^2 + b*c^3)*sqrt(b^2 - 4*a*c))*g*abs(a
*b^2 - 4*a^2*c) - 2*(a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4
+ 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2
*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (a*b^6*c - 8*a^2*b^4*c^2 - 2
*a*b^5*c^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3*c^3 + a*b^4*c^3 - 4*a^2*b^2*c^4 + (
a*b^5*c - 4*a^2*b^3*c^2 - 2*a*b^4*c^2 + a*b^3*c^3)*sqrt(b^2 - 4*a*c))*g)*lo

```
g(x^2 + 1/2*(a*b^3 - 4*a^2*b*c + sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 -
4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2
*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(
a*b^2 - 4*a^2*c)) - 1/8*(2*(b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*
c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) -
(b^4*c - 4*a*b^2*c^2 - 2*b^3*c^2 + b^2*c^3 - (b^3*c - 4*a*b*c^2 - 2*b^2*c^2
 + b*c^3)*sqrt(b^2 - 4*a*c))*g*abs(a*b^2 - 4*a^2*c) + 2*(a*b^5*c^2 - 8*a^2*
b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*
c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*
c))*e - (a*b^6*c - 8*a^2*b^4*c^2 - 2*a*b^5*c^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3
*c^3 + a*b^4*c^3 - 4*a^2*b^2*c^4 - (a*b^5*c - 4*a^2*b^3*c^2 - 2*a*b^4*c^2 +
 a*b^3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c - sqrt((
a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2
*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c
^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 8.61 (sec) , antiderivative size = 7373, normalized size of antiderivative = 19.10

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] symsum(log((5*b^3*c^4*d^3 + 8*a^3*c^4*f^3 - 96*a^2*c^5*d*e^2 + 72*a^2*c^5*d
^2*f - 3*b^4*c^3*d^2*f + 6*a^2*b^2*c^3*f^3 - 36*a*b*c^5*d^3 + 16*a*b^2*c^4*
d*e^2 + 18*a*b^2*c^4*d^2*f + 3*a*b^3*c^3*d*f^2 - 60*a^2*b*c^4*d*f^2 + 4*a*b
^4*c^2*d*g^2 + 16*a^2*b*c^4*e^2*f - 24*a^2*b^2*c^3*d*g^2 + 4*a^2*b^3*c^2*f*
g^2 - 16*a*b^3*c^3*d*e*g + 96*a^2*b*c^4*d*e*g - 16*a^2*b^2*c^3*e*f*g)/(8*(a
^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - root(1572864*a^8*b^
2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8
*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 3
2768*a^6*b*c^4*e*g*z^2 - 512*a^3*b^7*c*e*g*z^2 + 576*a^2*b^8*c*d*f*z^2 - 24
576*a^5*b^3*c^3*e*g*z^2 + 6144*a^4*b^5*c^2*e*g*z^2 + 24576*a^5*b^2*c^4*d*f*
z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 - 1536*a^4*b^6*c*
g^2*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5
*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^6*b^2*c^3*g^2*z^2 + 6144*a^5*b^4*c^
2*g^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5
*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 614
40*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z
^2 - 32*a*b^10*d*f*z^2 + 128*a^3*b^8*g^2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a
^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 + 384*a^2*b^6*c*d*f*g*z - 4096*a^4*b*c^4*d
*e*f*z + 64*a*b^7*c*d*e*f*z + 2048*a^4*b^2*c^3*d*f*g*z - 1536*a^3*b^4*c^2*d
*f*g*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 1024*a^5*b*c^
```

3*f^2*g*z + 192*a^3*b^5*c*f^2*g*z - 9216*a^4*b*c^4*d^2*g*z + 32*a^2*b^6*c*e
*f^2*z - 672*a*b^6*c^2*d^2*e*z + 336*a*b^7*c*d^2*g*z - 768*a^4*b^3*c^2*f^2*
g*z + 7936*a^3*b^3*c^3*d^2*g*z - 2496*a^2*b^5*c^2*d^2*g*z + 1536*a^4*b^2*c^
3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*
b^4*c^3*d^2*e*z - 32*a*b^8*d*f*g*z - 16*a^2*b^7*f^2*g*z - 2048*a^5*c^4*e*f^
2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 16*b^9*d^2*g*z - 768*a^3*b
*c^3*d*e*f*g + 32*a*b^5*c*d*e*f*g - 192*a^2*b^3*c^2*d*e*f*g + 16*a^2*b^4*c*
e*f^2*g + 48*a^2*b^4*c*d*f*g^2 - 240*a*b^4*c^2*d^2*e*g - 32*a*b^4*c^2*d*e^2
*f + 192*a^3*b^2*c^2*e*f^2*g + 192*a^3*b^2*c^2*d*f*g^2 + 960*a^2*b^2*c^3*d^
2*e*g + 192*a^2*b^2*c^3*d*e^2*f - 48*a^3*b^3*c*f^2*g^2 - 192*a^3*b*c^3*e^2*
f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2
 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 512*a^3*b*c^3*e^3*g + 128*
a^3*b^3*c*e*g^3 + 60*a*b^5*c*d^2*g^2 + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3
*d^3*f + 224*a^3*b*c^3*d*f^3 - 384*a^3*b^2*c^2*e^2*g^2 - 240*a^2*b^3*c^2*d^
2*g^2 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 + 16*b^6*c*d^2*e*g
 - 8*a*b^6*d*f*g^2 - 18*a*b^5*c*d*f^3 - 4*a^2*b^5*f^2*g^2 - 288*a^3*c^4*d^2
*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c
*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 4*b^7*d^2*g^2 - 16*a^4*c^3
*f^4 - 16*a^3*b^4*g^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4
, z, k)*(root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6
*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^
6*z^4 - 256*a^3*b^12*z^4 + 32768*a^6*b*c^4*e*g*z^2 - 512*a^3*b^7*c*e*g*z^2
+ 576*a^2*b^8*c*d*f*z^2 - 24576*a^5*b^3*c^3*e*g*z^2 + 6144*a^4*b^5*c^2*e*g*
z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c
^3*d*f*z^2 - 1536*a^4*b^6*c*g^2*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b
*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^6*b^2*c
^3*g^2*z^2 + 6144*a^5*b^4*c^2*g^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4
*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 5
12*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*
z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 + 128*a^3*b^8*g^2*z^2 -
32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 + 384*a^2*b^6*
c*d*f*g*z - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 2048*a^4*b^2*c^3*
d*f*g*z - 1536*a^3*b^4*c^2*d*f*g*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5
*c^2*d*e*f*z + 1024*a^5*b*c^3*f^2*g*z + 192*a^3*b^5*c*f^2*g*z - 9216*a^4*b*
c^4*d^2*g*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 336*a*b^7*c*d^
2*g*z - 768*a^4*b^3*c^2*f^2*g*z + 7936*a^3*b^3*c^3*d^2*g*z - 2496*a^2*b^5*c
^2*d^2*g*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3
*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 32*a*b^8*d*f*g*z - 16*a^2*b^7
*f^2*g*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z
- 16*b^9*d^2*g*z - 768*a^3*b*c^3*d*e*f*g + 32*a*b^5*c*d*e*f*g - 192*a^2*b^3
*c^2*d*e*f*g + 16*a^2*b^4*c*e*f^2*g + 48*a^2*b^4*c*d*f*g^2 - 240*a*b^4*c^2*
d^2*e*g - 32*a*b^4*c^2*d*e^2*f + 192*a^3*b^2*c^2*e*f^2*g + 192*a^3*b^2*c^2*
d*f*g^2 + 960*a^2*b^2*c^3*d^2*e*g + 192*a^2*b^2*c^3*d*e^2*f - 48*a^3*b^3*c*
f^2*g^2 - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d
*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f

+ 512*a^3*b*c^3*e^3*g + 128*a^3*b^3*c*e*g^3 + 60*a*b^5*c*d^2*g^2 + 2016*a^2
*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 384*a^3*b^2*c^2*
e^2*g^2 - 240*a^2*b^3*c^2*d^2*g^2 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^
3*d^2*f^2 + 16*b^6*c*d^2*e*g - 8*a*b^6*d*f*g^2 - 18*a*b^5*c*d*f^3 - 4*a^2*b
^5*f^2*g^2 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4
+ 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4
- 4*b^7*d^2*g^2 - 16*a^4*c^3*f^4 - 16*a^3*b^4*g^4 - 256*a^3*c^4*e^4 - 25*b^
4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*((x*(2048*a^5*c^6*e - 32*a^2*b^6*c^3*e
+ 384*a^3*b^4*c^4*e - 1536*a^4*b^2*c^5*e + 16*a^2*b^7*c^2*g - 192*a^3*b^5*c
^3*g + 768*a^4*b^3*c^4*g - 1024*a^5*b*c^5*g))/(4*(a^2*b^6 - 64*a^5*c^3 - 12
*a^3*b^4*c + 48*a^4*b^2*c^2)) - (6144*a^5*c^6*d - 288*a^2*b^6*c^3*d + 1920*
a^3*b^4*c^4*d - 5632*a^4*b^2*c^5*d + 16*a^2*b^7*c^2*f - 192*a^3*b^5*c^3*f +
768*a^4*b^3*c^4*f + 16*a*b^8*c^2*d - 1024*a^5*b*c^5*f)/(8*(a^2*b^6 - 64*a^
5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (root(1572864*a^8*b^2*c^5*z^4 - 9
83040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 61
44*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 32768*a^6*b*c^
4*e*g*z^2 - 512*a^3*b^7*c*e*g*z^2 + 576*a^2*b^8*c*d*f*z^2 - 24576*a^5*b^3*c
^3*e*g*z^2 + 6144*a^4*b^5*c^2*e*g*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^
3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 - 1536*a^4*b^6*c*g^2*z^2 + 122
88*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 43
2*a*b^9*c*d^2*z^2 - 8192*a^6*b^2*c^3*g^2*z^2 + 6144*a^5*b^4*c^2*g^2*z^2 - 8
192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*
z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^
4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^1
0*d*f*z^2 + 128*a^3*b^8*g^2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^
2 - 16*b^11*d^2*z^2 + 384*a^2*b^6*c*d*f*g*z - 4096*a^4*b*c^4*d*e*f*z + 64*a
*b^7*c*d*e*f*z + 2048*a^4*b^2*c^3*d*f*g*z - 1536*a^3*b^4*c^2*d*f*g*z + 3072
*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 1024*a^5*b*c^3*f^2*g*z + 1
92*a^3*b^5*c*f^2*g*z - 9216*a^4*b*c^4*d^2*g*z + 32*a^2*b^6*c*e*f^2*z - 672*
a*b^6*c^2*d^2*e*z + 336*a*b^7*c*d^2*g*z - 768*a^4*b^3*c^2*f^2*g*z + 7936*a^
3*b^3*c^3*d^2*g*z - 2496*a^2*b^5*c^2*d^2*g*z + 1536*a^4*b^2*c^3*e*f^2*z - 3
84*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e
*z - 32*a*b^8*d*f*g*z - 16*a^2*b^7*f^2*g*z - 2048*a^5*c^4*e*f^2*z + 18432*a
^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 16*b^9*d^2*g*z - 768*a^3*b*c^3*d*e*f*g
+ 32*a*b^5*c*d*e*f*g - 192*a^2*b^3*c^2*d*e*f*g + 16*a^2*b^4*c*e*f^2*g + 48*
a^2*b^4*c*d*f*g^2 - 240*a*b^4*c^2*d^2*e*g - 32*a*b^4*c^2*d*e^2*f + 192*a^3*
b^2*c^2*e*f^2*g + 192*a^3*b^2*c^2*d*f*g^2 + 960*a^2*b^2*c^3*d^2*e*g + 192*a
^2*b^2*c^3*d*e^2*f - 48*a^3*b^3*c*f^2*g^2 - 192*a^3*b*c^3*e^2*f^2 + 198*a*b
^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*
c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 512*a^3*b*c^3*e^3*g + 128*a^3*b^3*c*e*g
^3 + 60*a*b^5*c*d^2*g^2 + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*
a^3*b*c^3*d*f^3 - 384*a^3*b^2*c^2*e^2*g^2 - 240*a^2*b^3*c^2*d^2*g^2 - 16*a^
2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 + 16*b^6*c*d^2*e*g - 8*a*b^6*d*
f*g^2 - 18*a*b^5*c*d*f^3 - 4*a^2*b^5*f^2*g^2 - 288*a^3*c^4*d^2*f^2 - 16*b^5
*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*

a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 4*b^7*d^2*g^2 - 16*a^4*c^3*f^4 - 16*a^3
*b^4*g^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*x*(81
92*a^6*b*c^6 + 32*a^2*b^9*c^2 - 512*a^3*b^7*c^3 + 3072*a^4*b^5*c^4 - 8192*a
^5*b^3*c^5))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) -
(512*a^4*c^5*e*f - 32*a*b^5*c^3*d*e - 1024*a^3*b*c^5*d*e + 16*a*b^6*c^2*d*g
 - 256*a^4*b*c^4*f*g + 384*a^2*b^3*c^4*d*e - 192*a^2*b^4*c^3*d*g - 32*a^2*b
^4*c^3*e*f + 512*a^3*b^2*c^4*d*g + 16*a^2*b^5*c^2*f*g)/(8*(a^2*b^6 - 64*a^5
*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (x*(2*b^6*c^3*d^2 - 576*a^3*c^6*d^
2 + 64*a^4*c^5*f^2 - 36*a*b^4*c^4*d^2 + 128*a^3*b*c^5*e^2 + 256*a^2*b^2*c^5
*d^2 - 32*a^2*b^3*c^4*e^2 + 20*a^2*b^4*c^3*f^2 - 96*a^3*b^2*c^4*f^2 - 8*a^2
*b^5*c^2*g^2 + 32*a^3*b^3*c^3*g^2 + 4*a*b^5*c^3*d*f + 320*a^3*b*c^5*d*f - 9
6*a^2*b^3*c^4*d*f + 32*a^2*b^4*c^3*e*g - 128*a^3*b^2*c^4*e*g))/(4*(a^2*b^6
 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) - (x*(32*a^2*c^5*e^3 - 2*b^
3*c^4*d^2*e + b^4*c^3*d^2*g - 4*a^2*b^3*c^2*g^3 + 24*a*b*c^5*d^2*e - 48*a^2
*c^5*d*e*f - 12*a*b^2*c^4*d^2*g + 16*a^2*b*c^4*e*f^2 - 48*a^2*b*c^4*e^2*g +
 24*a^2*b^2*c^3*e*g^2 - 8*a^2*b^2*c^3*f^2*g - 4*a*b^2*c^4*d*e*f + 2*a*b^3*c
^3*d*f*g + 24*a^2*b*c^4*d*f*g))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 4
8*a^4*b^2*c^2)))*root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 32
7680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 104857
6*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 32768*a^6*b*c^4*e*g*z^2 - 512*a^3*b^7*c*
e*g*z^2 + 576*a^2*b^8*c*d*f*z^2 - 24576*a^5*b^3*c^3*e*g*z^2 + 6144*a^4*b^5*
c^2*e*g*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a
^4*b^4*c^3*d*f*z^2 - 1536*a^4*b^6*c*g^2*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 614
40*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a
^6*b^2*c^3*g^2*z^2 + 6144*a^5*b^4*c^2*g^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 +
1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2
*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*
c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 + 128*a^3*b^8*g^
2*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 + 384*
a^2*b^6*c*d*f*g*z - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 2048*a^4*
b^2*c^3*d*f*g*z - 1536*a^3*b^4*c^2*d*f*g*z + 3072*a^3*b^3*c^3*d*e*f*z - 768
*a^2*b^5*c^2*d*e*f*z + 1024*a^5*b*c^3*f^2*g*z + 192*a^3*b^5*c*f^2*g*z - 921
6*a^4*b*c^4*d^2*g*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 336*a*
b^7*c*d^2*g*z - 768*a^4*b^3*c^2*f^2*g*z + 7936*a^3*b^3*c^3*d^2*g*z - 2496*a
^2*b^5*c^2*d^2*g*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 1
5872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 32*a*b^8*d*f*g*z - 16
*a^2*b^7*f^2*g*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*
d^2*e*z - 16*b^9*d^2*g*z - 768*a^3*b*c^3*d*e*f*g + 32*a*b^5*c*d*e*f*g - 192
*a^2*b^3*c^2*d*e*f*g + 16*a^2*b^4*c*e*f^2*g + 48*a^2*b^4*c*d*f*g^2 - 240*a*
b^4*c^2*d^2*e*g - 32*a*b^4*c^2*d*e^2*f + 192*a^3*b^2*c^2*e*f^2*g + 192*a^3*
b^2*c^2*d*f*g^2 + 960*a^2*b^2*c^3*d^2*e*g + 192*a^2*b^2*c^3*d*e^2*f - 48*a^
3*b^3*c*f^2*g^2 - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b
^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*
d*e^2*f + 512*a^3*b*c^3*e^3*g + 128*a^3*b^3*c*e*g^3 + 60*a*b^5*c*d^2*g^2 +
2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 384*a^3*

```
b^2*c^2*e^2*g^2 - 240*a^2*b^3*c^2*d^2*g^2 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^
2*b^2*c^3*d^2*f^2 + 16*b^6*c*d^2*e*g - 8*a*b^6*d*f*g^2 - 18*a*b^5*c*d*f^3 -
 4*a^2*b^5*f^2*g^2 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*
c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*
c^4*d^4 - 4*b^7*d^2*g^2 - 16*a^4*c^3*f^4 - 16*a^3*b^4*g^4 - 256*a^3*c^4*e^4
 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k), k, 1, 4) + ((b*e - 2*a*g)/(2*(
4*a*c - b^2)) + (x^2*(2*c*e - b*g))/(2*(4*a*c - b^2)) + (x*(2*a*c*d - b^2*d
 + a*b*f))/(2*a*(4*a*c - b^2)) - (c*x^3*(b*d - 2*a*f))/(2*a*(4*a*c - b^2)))
/(a + b*x^2 + c*x^4)
```

3.39 $\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2} dx$

Optimal result

Integrand size = 35, antiderivative size = 439

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2} dx \\
&= -\frac{be-2ag+(2ce-bg)x^2}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} \\
&\quad + \frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad + \frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}
\end{aligned}$$

```
[Out] 1/2*(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*x*(b^2*d
-a*b*f-2*a*(-a*h+c*d)+(a*b*h-2*a*c*f+b*c*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^
2+a)+(-b*g+2*c*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(3/2
)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*(b*c*d-2*a*c*f
+a*b*h+(4*a*b*c*f+b^2*(-a*h+c*d)-4*a*c*(a*h+3*c*d))/(-4*a*c+b^2)^(1/2))/a/(
-4*a*c+b^2)*2^(1/2)/c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/4*arctan(x*2^(1/
2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*(b*c*d-2*a*c*f+a*b*h+(-4*a*b*c*f-b
^2*(-a*h+c*d)+4*a*c*(a*h+3*c*d))/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)
/c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 1.22 (sec) , antiderivative size = 439, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.229, Rules used = {1687, 1692, 1180, 211, 1261, 652, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2}\,dx$$
$$= \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{b^2(cd-ah)+4abcf-4ac(ah+3cd)}{\sqrt{b^2-4ac}}+abh-2acf+bcd\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{b^2(cd-ah)+4abcf-4ac(ah+3cd)}{\sqrt{b^2-4ac}}+abh-2acf+bcd\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$
$$+\frac{x(x^2(abh-2acf+bcd)-abf-2a(cd-ah)+b^2d)}{2a\,(b^2-4ac)\,(a+bx^2+cx^4)}-\frac{-2ag+x^2(2ce-bg)+be}{2\,(b^2-4ac)\,(a+bx^2+cx^4)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] -1/2*(b*e - 2*a*g + (2*c*e - b*g)*x^2)/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4))
+ (x*(b^2*d - a*b*f - 2*a*(c*d - a*h) + (b*c*d - 2*a*c*f + a*b*h)*x^2))/(2*
a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + ((b*c*d - 2*a*c*f + a*b*h + (4*a*b*c
*f + b^2*(c*d - a*h) - 4*a*c*(3*c*d + a*h))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt
[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*Sqrt[c]*(b^2 - 4*
a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((b*c*d - 2*a*c*f + a*b*h - (4*a*b*c*f
+ b^2*(c*d - a*h) - 4*a*c*(3*c*d + a*h))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]
*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*Sqrt[c]*(b^2 - 4*a*c
)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/Sqrt[
b^2 - 4*a*c]])/(b^2 - 4*a*c)^(3/2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^2}\,dx + \int \frac{d+fx^2+hx^4}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(a+bx+cx^2)^2}\,dx, x, x^2\right) - \frac{\int \frac{-b^2d-abf+2a(3cd+ah)+(-bcd+2acf-abh)x^2}{a+bx^2+cx^4}\,dx}{2a(b^2-4ac)}$$

$$= -\frac{be-2ag+(2ce-bg)x^2}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{2(b^2-4ac)} + \frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)} + \frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)}$$

$$= -\frac{be-2ag+(2ce-bg)x^2}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce-bg)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{b^2-4ac}$$

$$= -\frac{be - 2ag + (2ce - bg)x^2}{2\left(b^2 - 4ac\right)\left(a + bx^2 + cx^4\right)} + \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{2a\left(b^2 - 4ac\right)\left(a + bx^2 + cx^4\right)} + \frac{\left(bcd - 2acf + abh + \frac{4abcf + b^2(cd - ah) - 4ac(3cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b - \sqrt{b^2 - 4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\left(b^2 - 4ac\right)\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\left(bcd - 2acf + abh - \frac{4abcf + b^2(cd - ah) - 4ac(3cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b + \sqrt{b^2 - 4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\left(b^2 - 4ac\right)\sqrt{b + \sqrt{b^2 - 4ac}}} + \frac{(2ce - bg)\tanh^{-1}\left(\frac{b + 2cx^2}{\sqrt{b^2 - 4ac}}\right)}{\left(b^2 - 4ac\right)^{3/2}}$$

Mathematica [A] (verified)

Time = 1.09 (sec) , antiderivative size = 489, normalized size of antiderivative = 1.11

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4}{\left(a + bx^2 + cx^4\right)^2}\,dx$$

$$= \frac{1}{4}\left(\frac{-4a^2(g + hx) - 2bdx(b + cx^2) + 4acx(d + x(e + fx)) + 2ab(e + x(f - x(g + hx)))}{a\left(-b^2 + 4ac\right)\left(a + bx^2 + cx^4\right)} + \frac{\sqrt{2}\left(b^2(cd - ah) - 2ac\left(6cd + \sqrt{b^2 - 4ac}f + 2ah\right) + b\left(c\sqrt{b^2 - 4ac}d + 4acf + a\sqrt{b^2 - 4ac}h\right)\right)\arctan\left(\ldots\right)}{a\sqrt{c}\left(b^2 - 4ac\right)^{3/2}\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\sqrt{2}\left(b^2(-cd + ah) + 2ac\left(6cd - \sqrt{b^2 - 4ac}f + 2ah\right) + b\left(c\sqrt{b^2 - 4ac}d - 4acf + a\sqrt{b^2 - 4ac}h\right)\right)\arctan\left(\ldots\right)}{a\sqrt{c}\left(b^2 - 4ac\right)^{3/2}\sqrt{b + \sqrt{b^2 - 4ac}}} + \frac{2(-2ce + bg)\log\left(-b + \sqrt{b^2 - 4ac} - 2cx^2\right)}{\left(b^2 - 4ac\right)^{3/2}} - \frac{2(-2ce + bg)\log\left(b + \sqrt{b^2 - 4ac} + 2cx^2\right)}{\left(b^2 - 4ac\right)^{3/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((-4*a^2*(g + h*x) - 2*b*d*x*(b + c*x^2) + 4*a*c*x*(d + x*(e + f*x)) + 2*a*
b*(e + x*(f - x*(g + h*x))))/(a*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt
[2]*(b^2*(c*d - a*h) - 2*a*c*(6*c*d + Sqrt[b^2 - 4*a*c]*f + 2*a*h) + b*(c*S
qrt[b^2 - 4*a*c]*d + 4*a*c*f + a*Sqrt[b^2 - 4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt
[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a*Sqrt[c]*(b^2 - 4*a*c)^(3/2)*Sqrt[b
- Sqrt[b^2 - 4*a*c]]) + (Sqrt[2]*(b^2*(-(c*d) + a*h) + 2*a*c*(6*c*d - Sqrt[
b^2 - 4*a*c]*f + 2*a*h) + b*(c*Sqrt[b^2 - 4*a*c]*d - 4*a*c*f + a*Sqrt[b^2 -
```

```
4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a*Sqr
t[c]*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + (2*(-2*c*e + b*g)*L
og[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) - (2*(-2*c*e + b*
g)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.15 (sec) , antiderivative size = 274, normalized size of antiderivative = 0.62

method	result
risch	$\frac{-\frac{(abh-2acf+bcd)x^3}{2a(4ac-b^2)}-\frac{(bg-2ec)x^2}{2(4ac-b^2)}-\frac{(2a^2h-abf-2acd+b^2d)x}{2a(4ac-b^2)}-\frac{2ag-be}{2(4ac-b^2)}}{cx^4+bx^2+a}+\frac{\left(\sum_{\_R=\mathrm{RootOf}(c\_Z^4+\_Z^2b+a)}\frac{\left(-\frac{(abh-2acf+bcd)\_R^2}{a(4ac-b^2)}\right.}{}\right.}{4}$
default	$\frac{-\frac{(abh-2acf+bcd)x^3}{2a(4ac-b^2)}-\frac{(bg-2ec)x^2}{2(4ac-b^2)}-\frac{(2a^2h-abf-2acd+b^2d)x}{2a(4ac-b^2)}-\frac{2ag-be}{2(4ac-b^2)}}{cx^4+bx^2+a}+\frac{2c\left(\frac{\left(-4\sqrt{-4ac+b^2}\,abcg+8\sqrt{-4ac+b^2}\,a c^2 e\right)\ln\left(2cx^2+\sqrt{-4\right.}{4c}\right.}{}$

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (-1/2/a*(a*b*h-2*a*c*f+b*c*d)/(4*a*c-b^2)*x^3-1/2*(b*g-2*c*e)/(4*a*c-b^2)*x
^2-1/2*(2*a^2*h-a*b*f-2*a*c*d+b^2*d)/a/(4*a*c-b^2)*x-1/2*(2*a*g-b*e)/(4*a*c
-b^2))/(c*x^4+b*x^2+a)+1/4*sum((-1/a*(a*b*h-2*a*c*f+b*c*d)/(4*a*c-b^2)*_R^2
-2*(b*g-2*c*e)/(4*a*c-b^2)*_R+(2*a^2*h-a*b*f+6*a*c*d-b^2*d)/a/(4*a*c-b^2))/
(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas"
)
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2}\,dx = \int \frac{hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="maxima"
)
```

```
[Out] 1/2*((b*c*d - 2*a*c*f + a*b*h)*x^3 - a*b*e + 2*a^2*g - (2*a*c*e - a*b*g)*x^
2 - (a*b*f - 2*a^2*h - (b^2 - 2*a*c)*d)*x)/((a*b^2*c - 4*a^2*c^2)*x^4 + a^2
*b^2 - 4*a^3*c + (a*b^3 - 4*a^2*b*c)*x^2) + 1/2*integrate((a*b*f - 2*a^2*h
+ (b*c*d - 2*a*c*f + a*b*h)*x^2 + (b^2 - 6*a*c)*d - 2*(2*a*c*e - a*b*g)*x)/
(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 7495 vs. 2(391) = 782.

Time = 1.73 (sec) , antiderivative size = 7495, normalized size of antiderivative = 17.07

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] 1/2*(b*c*d*x^3 - 2*a*c*f*x^3 + a*b*h*x^3 - 2*a*c*e*x^2 + a*b*g*x^2 + b^2*d*
x - 2*a*c*d*x - a*b*f*x + 2*a^2*h*x - a*b*e + 2*a^2*g)/((c*x^4 + b*x^2 + a)
*(a*b^2 - 4*a^2*c)) + 1/16*((2*b^3*c^3 - 8*a*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*b*c^3 - 2*(b^2 - 4*a*c)*b*c^3)*(a*b^2 - 4*a^2*c)^2*d - 2*(
2*a*b^2*c^3 - 8*a^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a*b^2*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
```

*c)*a^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^3 -
2*(b^2 - 4*a*c)*a*c^3)*(a*b^2 - 4*a^2*c)^2*f + (2*a*b^3*c^2 - 8*a^2*b*c^3 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3 + 4*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c + 2*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c - sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - 2*(b^2 - 4*a*c)*a*b*c^2)*
(a*b^2 - 4*a^2*c)^2*h + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c
- 14*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 - 2*sqrt(2)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 - 2*a*b^6*c^2 + 64*sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 + 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^2*b^3*c^3 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 28*a^2*
b^4*c^3 - 96*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 - 48*sqrt(2)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 10*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*b^2*c^4 - 128*a^3*b^2*c^4 + 24*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^3*c^5 + 192*a^4*c^5 + 2*(b^2 - 4*a*c)*a*b^4*c^2 - 20*(b^2 - 4*
a*c)*a^2*b^2*c^3 + 48*(b^2 - 4*a*c)*a^3*c^4)*d*abs(a*b^2 - 4*a^2*c) + 2*(sq
rt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 8*sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^3*b^3*c^2 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
2*b^4*c^2 - 2*a^2*b^5*c^2 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*
b*c^3 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 + sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 16*a^3*b^3*c^3 - 4*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 32*a^4*b*c^4 + 2*(b^2 - 4*a*c)*a^2*b
^3*c^2 - 8*(b^2 - 4*a*c)*a^3*b*c^3)*f*abs(a*b^2 - 4*a^2*c) - 4*(sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^4*b^2*c^2 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2
- 2*a^3*b^4*c^2 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*c^3 + 8*s
qrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^3*b^2*c^3 + 16*a^4*b^2*c^3 - 4*sqrt(2)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^4*c^4 - 32*a^5*c^4 + 2*(b^2 - 4*a*c)*a^3*b^2*c^2 - 8*(b^2 -
4*a*c)*a^4*c^3)*h*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^3 - 40*a^3*b^5*c^4 +
224*a^4*b^3*c^5 - 384*a^5*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^2*b^7*c + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^3*b^5*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^2*b^6*c^2 - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^4*b^3*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^3*b^4*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^2*b^5*c^3 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^5*b*c^4 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^4*b^2*c^4 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^3*b^3*c^4 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^4*b*c^5 - 2*(b^2 - 4*a*c)*a^2*b^5*c^3 + 32*(b^2 - 4*a*c)*a^3*
b^3*c^4 - 96*(b^2 - 4*a*c)*a^4*b*c^5)*d + 4*(2*a^3*b^6*c^3 - 16*a^4*b^4*c^4
+ 32*a^5*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^3*b^6*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*

```
a^4*b^4*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^3*b^5*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^5*b^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
4*b^3*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b
^4*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^
2*c^4 - 2*(b^2 - 4*a*c)*a^3*b^4*c^3 + 8*(b^2 - 4*a*c)*a^4*b^2*c^4)*f - (2*a
^3*b^7*c^2 - 8*a^4*b^5*c^3 - 32*a^5*b^3*c^4 + 128*a^6*b*c^5 - sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^7 + 4*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c + 16*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 - 2*(b^2 - 4*a*c)*a^3*b^5*c^2 + 32*
(b^2 - 4*a*c)*a^5*b*c^4)*h)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c +
sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/
(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6*c - 12*a^4*b^4*c^2 - 2*a^3*b^5*c^2 + 48*a
^5*b^2*c^3 + 16*a^4*b^3*c^3 + a^3*b^4*c^3 - 64*a^6*c^4 - 32*a^5*b*c^4 - 8*a
^4*b^2*c^4 + 16*a^5*c^5)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/16*((2*b^3*c^3 -
8*a*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c
 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + 2*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^2 - sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^3 - 2*(b^2 - 4*a*c)*
b*c^3)*(a*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^3 - 8*a^2*c^4 - sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c + 4*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*c^3 - 2*(b^2 - 4*a*c)*a*c^3)*(a*b^2 - 4*a^2*c)^
2*f + (2*a*b^3*c^2 - 8*a^2*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*a^2*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
*b*c^2 - 2*(b^2 - 4*a*c)*a*b*c^2)*(a*b^2 - 4*a^2*c)^2*h - 2*(sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^2*b^4*c^2 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 2*a
*b^6*c^2 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 + 20*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a*b^4*c^3 - 28*a^2*b^4*c^3 - 96*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^4*c^4 - 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4
- 10*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 128*a^3*b^2*c^4
+ 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 192*a^4*c^5 - 2*(b^2
 - 4*a*c)*a*b^4*c^2 + 20*(b^2 - 4*a*c)*a^2*b^2*c^3 - 48*(b^2 - 4*a*c)*a^3*c
^4)*d*abs(a*b^2 - 4*a^2*c) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2
*b^5*c - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 - 2*sqrt(2)*
```

```
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 2*a^2*b^5*c^2 + 16*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^3*b^2*c^3 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3
 - 16*a^3*b^3*c^3 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 32
*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 8*(b^2 - 4*a*c)*a^3*b*c^3)*f*abs
(a*b^2 - 4*a^2*c) + 4*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c -
8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 - 2*sqrt(2)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 2*a^3*b^4*c^2 + 16*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^5*c^3 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
4*b*c^3 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 16*a^4*b^2*
c^3 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 32*a^5*c^4 - 2*(b
^2 - 4*a*c)*a^3*b^2*c^2 + 8*(b^2 - 4*a*c)*a^4*c^3)*h*abs(a*b^2 - 4*a^2*c) +
 (2*a^2*b^7*c^3 - 40*a^3*b^5*c^4 + 224*a^4*b^3*c^5 - 384*a^5*b*c^6 - sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7*c + 20*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 + 2*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c^2 - 112*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^3 - 32*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^3 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^3 + 192*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 + 96*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^4 + 16*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^4 - 48*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^5 - 2*(b^2 - 4*a*c)*a
^2*b^5*c^3 + 32*(b^2 - 4*a*c)*a^3*b^3*c^4 - 96*(b^2 - 4*a*c)*a^4*b*c^5)*d +
 4*(2*a^3*b^6*c^3 - 16*a^4*b^4*c^4 + 32*a^5*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^4 - 2*(b^2 - 4*a*c)*a^3*b^4*c^3 + 8*(
b^2 - 4*a*c)*a^4*b^2*c^4)*f - (2*a^3*b^7*c^2 - 8*a^4*b^5*c^3 - 32*a^5*b^3*c
^4 + 128*a^6*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^3*b^7 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^4*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*
b^6*c + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^
3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c
^2 - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^3
 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3
 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 -
 2*(b^2 - 4*a*c)*a^3*b^5*c^2 + 32*(b^2 - 4*a*c)*a^5*b*c^4)*h)*arctan(2*sqrt
(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 -
 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6*c - 12*
a^4*b^4*c^2 - 2*a^3*b^5*c^2 + 48*a^5*b^2*c^3 + 16*a^4*b^3*c^3 + a^3*b^4*c^3
```

```
 - 64*a^6*c^4 - 32*a^5*b*c^4 - 8*a^4*b^2*c^4 + 16*a^5*c^5)*abs(a*b^2 - 4*a^
2*c)*abs(c)) - 1/8*(2*(b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 - (b^2*c^2 -
 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (b^4*
c - 4*a*b^2*c^2 - 2*b^3*c^2 + b^2*c^3 + (b^3*c - 4*a*b*c^2 - 2*b^2*c^2 + b*
c^3)*sqrt(b^2 - 4*a*c))*g*abs(a*b^2 - 4*a^2*c) - 2*(a*b^5*c^2 - 8*a^2*b^3*c
^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 +
 (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e
 + (a*b^6*c - 8*a^2*b^4*c^2 - 2*a*b^5*c^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3*c^3
+ a*b^4*c^3 - 4*a^2*b^2*c^4 + (a*b^5*c - 4*a^2*b^3*c^2 - 2*a*b^4*c^2 + a*b^
3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c + sqrt((a*b^3
 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c -
4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 +
a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c)) - 1/8*(2*(b^3*c^2 - 4*a*b*
c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*
a*c))*e*abs(a*b^2 - 4*a^2*c) - (b^4*c - 4*a*b^2*c^2 - 2*b^3*c^2 + b^2*c^3 -
 (b^3*c - 4*a*b*c^2 - 2*b^2*c^2 + b*c^3)*sqrt(b^2 - 4*a*c))*g*abs(a*b^2 - 4
*a^2*c) - 2*(a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a^2
*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3*c
^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (a*b^6*c - 8*a^2*b^4*c^2 - 2*a*b^5*c
^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3*c^3 + a*b^4*c^3 - 4*a^2*b^2*c^4 - (a*b^5*c
- 4*a^2*b^3*c^2 - 2*a*b^4*c^2 + a*b^3*c^3)*sqrt(b^2 - 4*a*c))*g)*log(x^2 +
1/2*(a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)
*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a
*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 -
4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 8.80 (sec) , antiderivative size = 13024, normalized size of antiderivative = 29.67

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4}{(a + bx^2 + cx^4)^2} \, dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] ((b*e - 2*a*g)/(2*(4*a*c - b^2)) + (x^2*(2*c*e - b*g))/(2*(4*a*c - b^2)) -
(x*(b^2*d + 2*a^2*h - 2*a*c*d - a*b*f))/(2*a*(4*a*c - b^2)) - (x^3*(b*c*d -
 2*a*c*f + a*b*h))/(2*a*(4*a*c - b^2)))/(a + b*x^2 + c*x^4) + symsum(log((5
*b^3*c^4*d^3 + 8*a^3*c^4*f^3 - 96*a^2*c^5*d*e^2 + 72*a^2*c^5*d^2*f - 3*a^3*
b^3*c*h^3 - 4*a^4*b*c^2*h^3 - 3*b^4*c^3*d^2*f - 32*a^3*c^4*e^2*h + b^5*c^2*
d^2*h + 8*a^4*c^3*f*h^2 + 6*a^2*b^2*c^3*f^3 - 36*a*b*c^5*d^3 + a*b^5*c*d*h^
2 + 48*a^3*c^4*d*f*h + 16*a*b^2*c^4*d*e^2 + 18*a*b^2*c^4*d^2*f + 3*a*b^3*c^
3*d*f^2 - 60*a^2*b*c^4*d*f^2 + 4*a*b^4*c^2*d*g^2 + 16*a^2*b*c^4*e^2*f - a*b
^3*c^3*d^2*h - 60*a^2*b*c^4*d^2*h - 28*a^3*b*c^3*d*h^2 + a^2*b^4*c*f*h^2 -
28*a^3*b*c^3*f^2*h - 24*a^2*b^2*c^3*d*g^2 - 9*a^2*b^3*c^2*d*h^2 + 4*a^2*b^3
```

*c^2*f*g^2 - 5*a^2*b^3*c^2*f^2*h + 18*a^3*b^2*c^2*f*h^2 - 8*a^3*b^2*c^2*g^2
*h - 16*a*b^3*c^3*d*e*g + 96*a^2*b*c^4*d*e*g - 4*a*b^4*c^2*d*f*h + 32*a^3*b
*c^3*e*g*h + 52*a^2*b^2*c^3*d*f*h - 16*a^2*b^2*c^3*e*f*g)/(8*(a^2*b^6 - 64*
a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - root(1572864*a^8*b^2*c^6*z^4 -
983040*a^7*b^4*c^5*z^4 + 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6
144*a^4*b^10*c^2*z^4 - 256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 192*a^3*b
^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^2*b
^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b^
6*c^2*f*h*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g*z^2 + 153
60*a^4*b^5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*c^2*d*h*z^
2 - 512*a^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^3*b^6*c^3*
d*f*z^2 + 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 12288*a^7*b*
c^4*h^2*z^2 + 128*a^3*b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*
c*f^2*z^2 + 61440*a^5*b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 16384*a^7*c^5
*f*h*z^2 - 49152*a^6*c^6*d*f*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*
c^2*h^2*z^2 - 8192*a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^
4*b^6*c^2*g^2*z^2 - 8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 2
4576*a^5*b^2*c^5*e^2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z
^2 - 61440*a^4*b^3*c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c
^3*d^2*z^2 - 16*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2
 - 6144*a^5*b*c^4*d*g*h*z + 96*a^2*b^7*c*d*g*h*z - 4096*a^4*b*c^5*d*e*f*z +
 64*a*b^7*c^2*d*e*f*z - 32*a*b^8*c*d*f*g*z + 4608*a^4*b^3*c^3*d*g*h*z - 115
2*a^3*b^5*c^2*d*g*h*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z
 + 2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*
f*g*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^
3*d*e*f*z - 1024*a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 1024*a^5*b*c^4
*f^2*g*z - 32*a^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2
*g*z + 336*a*b^7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^5*c^5*d*e*h*
z + 768*a^5*b^3*c^2*g*h^2*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^
2*g*z + 384*a^4*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^
4*d^2*g*z - 2496*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b
^4*c^3*e*f^2*z + 32*a^2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*
a^2*b^4*c^4*d^2*e*z + 16*a^3*b^7*g*h^2*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*
c^5*e*f^2*z + 32*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z
 - 256*a^4*b*c^3*e*f*g*h - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g - 1
92*a^3*b^3*c^2*e*f*g*h + 896*a^3*b^2*c^3*d*e*g*h - 96*a^2*b^4*c^2*d*e*g*h -
 192*a^2*b^3*c^3*d*e*f*g + 48*a^3*b^4*c*f*g^2*h + 16*a^3*b^4*c*e*g*h^2 + 24
*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d^2*f*h + 800*a^4*b*c^3*d*f*h^2 - 102*a
*b^5*c^2*d^2*f*h - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4
*c^3*d^2*e*g - 32*a*b^4*c^3*d*e^2*f + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^
2 + 64*a^4*b^2*c^2*f*g^2*h + 192*a^4*b^2*c^2*e*g*h^2 - 224*a^3*b^3*c^2*d*g^
2*h + 192*a^3*b^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 336*a^3*b^3*c^2*d
*f*h^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2
*e*f^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^
3*d*e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*

c^4*d*e^2*f - 48*a^4*b^3*c*g^2*h^2 + 80*a^3*b^3*c^2*f^3*h - 42*a^3*b^4*c*f^
2*h^2 - 192*a^4*b*c^3*e^2*h^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^3
 - 192*a^2*b^2*c^4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 +
60*a*b^5*c^2*d^2*g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960*
a^2*b*c^5*d^2*e^2 + 240*a*b^3*c^4*d^2*e^2 + 256*a^4*c^4*e^2*f*h - 192*a^4*c
^4*d*f^2*h + 16*b^6*c^2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h +
 80*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^2*f + 512*a^3*b*c
^4*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 +
2016*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5
*c^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*
c^2*e^2*h^2 - 464*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^
4*c^2*d^2*h^2 - 240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*
b^2*c^4*d^2*f^2 + 6*b^7*c*d^2*f*h - 2*a*b^7*d*f*h^2 - 32*a^5*c^3*f^2*h^2 -
4*a^3*b^5*g^2*h^2 - 864*a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d
^2*f^2 - 16*b^5*c^3*d^2*e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 - 10*b
^6*c^2*d^3*h + 6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3 - 4
*b^7*c*d^2*g^2 + 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4 - 16
*a^3*b^4*c*g^4 + 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 16*a^
4*c^4*f^4 - 256*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b^6*f
^2*h^2 - b^8*d^2*h^2, z, k)*(root(1572864*a^8*b^2*c^6*z^4 - 983040*a^7*b^4*
c^5*z^4 + 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b^10*c^
2*z^4 - 256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 192*a^3*b^8*c*f*h*z^2 +
57344*a^6*b*c^5*d*h*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^2*b^9*c*d*h*z^2 -
32*a*b^10*c*d*f*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b^6*c^2*f*h*z^2 -
 49152*a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g*z^2 + 15360*a^4*b^5*c^3*
d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*c^2*d*h*z^2 - 512*a^3*b^7
*c^2*e*g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^3*b^6*c^3*d*f*z^2 + 2048*
a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 12288*a^7*b*c^4*h^2*z^2 + 1
28*a^3*b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*c*f^2*z^2 + 614
40*a^5*b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 16384*a^7*c^5*f*h*z^2 - 4915
2*a^6*c^6*d*f*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8
192*a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z
^2 - 8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^
5*e^2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*
b^3*c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 16
*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2 - 6144*a^5*b*c
^4*d*g*h*z + 96*a^2*b^7*c*d*g*h*z - 4096*a^4*b*c^5*d*e*f*z + 64*a*b^7*c^2*d
*e*f*z - 32*a*b^8*c*d*f*g*z + 4608*a^4*b^3*c^3*d*g*h*z - 1152*a^3*b^5*c^2*d
*g*h*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z + 2048*a^4*b^2
*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*f*g*z - 192*a^2
*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^3*d*e*f*z - 102
4*a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 1024*a^5*b*c^4*f^2*g*z - 32*a
^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2*g*z + 336*a*b^
7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^5*c^5*d*e*h*z + 768*a^5*b^3
*c^2*g*h^2*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^2*g*z + 384*a^4

*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 249
6*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*z
+ 32*a^2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d^2
*e*z + 16*a^3*b^7*g*h^2*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*c^5*e*f^2*z + 3
2*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z - 256*a^4*b*c^
3*e*f*g*h - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g - 192*a^3*b^3*c^2*
e*f*g*h + 896*a^3*b^2*c^3*d*e*g*h - 96*a^2*b^4*c^2*d*e*g*h - 192*a^2*b^3*c^
3*d*e*f*g + 48*a^3*b^4*c*f*g^2*h + 16*a^3*b^4*c*e*g*h^2 + 24*a^2*b^5*c*d*g^
2*h + 2208*a^3*b*c^4*d^2*f*h + 800*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2*f*
h - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4*c^3*d^2*e*g -
32*a*b^4*c^3*d*e^2*f + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^2 + 64*a^4*b^2*
c^2*f*g^2*h + 192*a^4*b^2*c^2*e*g*h^2 - 224*a^3*b^3*c^2*d*g^2*h + 192*a^3*b
^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 336*a^3*b^3*c^2*d*f*h^2 + 192*a^
3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2*e*f^2*g - 12*a
^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^3*d*e^2*h + 48*
a^2*b^4*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*c^4*d*e^2*f - 4
8*a^4*b^3*c*g^2*h^2 + 80*a^3*b^3*c^2*f^3*h - 42*a^3*b^4*c*f^2*h^2 - 192*a^4
*b*c^3*e^2*h^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^3 - 192*a^2*b^2*
c^4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 + 60*a*b^5*c^2*d^
2*g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960*a^2*b*c^5*d^2*e
^2 + 240*a*b^3*c^4*d^2*e^2 + 256*a^4*c^4*e^2*f*h - 192*a^4*c^4*d*f^2*h + 16
*b^6*c^2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h + 80*a^4*b^3*c*f
*h^3 + 6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^2*f + 512*a^3*b*c^4*e^3*g + 132*
a*b^4*c^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 + 2016*a^2*b*c^5*
d^3*f - 496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5*c^2*d*f^3 - 19
2*a^4*b^2*c^2*f^2*h^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 - 4
64*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2 -
240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f^2
+ 6*b^7*c*d^2*f*h - 2*a*b^7*d*f*h^2 - 32*a^5*c^3*f^2*h^2 - 4*a^3*b^5*g^2*h
^2 - 864*a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d^2*f^2 - 16*b^5
*c^3*d^2*e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 - 10*b^6*c^2*d^3*h +
6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3 - 4*b^7*c*d^2*g^2
+ 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4 - 16*a^3*b^4*c*g^4
+ 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 16*a^4*c^4*f^4 - 256
*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b^6*f^2*h^2 - b^8*d^
2*h^2, z, k)*((x*(2048*a^5*c^6*e - 32*a^2*b^6*c^3*e + 384*a^3*b^4*c^4*e - 1
536*a^4*b^2*c^5*e + 16*a^2*b^7*c^2*g - 192*a^3*b^5*c^3*g + 768*a^4*b^3*c^4*
g - 1024*a^5*b*c^5*g))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2
*c^2)) - (6144*a^5*c^6*d + 2048*a^6*c^5*h - 288*a^2*b^6*c^3*d + 1920*a^3*b^
4*c^4*d - 5632*a^4*b^2*c^5*d + 16*a^2*b^7*c^2*f - 192*a^3*b^5*c^3*f + 768*a
^4*b^3*c^4*f - 32*a^3*b^6*c^2*h + 384*a^4*b^4*c^3*h - 1536*a^5*b^2*c^4*h +
16*a*b^8*c^2*d - 1024*a^5*b*c^5*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c
+ 48*a^4*b^2*c^2)) + (root(1572864*a^8*b^2*c^6*z^4 - 983040*a^7*b^4*c^5*z^4
+ 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b^10*c^2*z^4 -
256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 192*a^3*b^8*c*f*h*z^2 + 57344*a

^6*b*c^5*d*h*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^2*b^9*c*d*h*z^2 - 32*a*b^
10*c*d*f*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b^6*c^2*f*h*z^2 - 49152*
a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g*z^2 + 15360*a^4*b^5*c^3*d*h*z^2
 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*c^2*d*h*z^2 - 512*a^3*b^7*c^2*e*
g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^3*b^6*c^3*d*f*z^2 + 2048*a^4*b^4
*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 12288*a^7*b*c^4*h^2*z^2 + 128*a^3*
b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*c*f^2*z^2 + 61440*a^5*
b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 16384*a^7*c^5*f*h*z^2 - 49152*a^6*c
^6*d*f*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8192*a^6
*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z^2 - 81
92*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^5*e^2*z
^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*b^3*c^5
*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 16*a^3*b^
9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2 - 6144*a^5*b*c^4*d*g*
h*z + 96*a^2*b^7*c*d*g*h*z - 4096*a^4*b*c^5*d*e*f*z + 64*a*b^7*c^2*d*e*f*z
 - 32*a*b^8*c*d*f*g*z + 4608*a^4*b^3*c^3*d*g*h*z - 1152*a^3*b^5*c^2*d*g*h*z
 - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z + 2048*a^4*b^2*c^4*d*
f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*f*g*z - 192*a^2*b^6*c^
2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^3*d*e*f*z - 1024*a^6*b
*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 1024*a^5*b*c^4*f^2*g*z - 32*a^3*b^6*
c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2*g*z + 336*a*b^7*c^2*d
^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^5*c^5*d*e*h*z + 768*a^5*b^3*c^2*g*
h^2*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^2*g*z + 384*a^4*b^4*c^
2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 2496*a^2*b
^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*z + 32*a^
2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d^2*e*z +
16*a^3*b^7*g*h^2*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*c^5*e*f^2*z + 32*b^8*c
^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z - 256*a^4*b*c^3*e*f*g
*h - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g - 192*a^3*b^3*c^2*e*f*g*h
 + 896*a^3*b^2*c^3*d*e*g*h - 96*a^2*b^4*c^2*d*e*g*h - 192*a^2*b^3*c^3*d*e*f
*g + 48*a^3*b^4*c*f*g^2*h + 16*a^3*b^4*c*e*g*h^2 + 24*a^2*b^5*c*d*g^2*h + 2
208*a^3*b*c^4*d^2*f*h + 800*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2*f*h - 30*
a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4*c^3*d^2*e*g - 32*a*b^
4*c^3*d*e^2*f + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^2 + 64*a^4*b^2*c^2*f*g
^2*h + 192*a^4*b^2*c^2*e*g*h^2 - 224*a^3*b^3*c^2*d*g^2*h + 192*a^3*b^2*c^3*
e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 336*a^3*b^3*c^2*d*f*h^2 + 192*a^3*b^2*c
^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2*e*f^2*g - 12*a^2*b^4*
c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^3*d*e^2*h + 48*a^2*b^4
*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*c^4*d*e^2*f - 48*a^4*b
^3*c*g^2*h^2 + 80*a^3*b^3*c^2*f^3*h - 42*a^3*b^4*c*f^2*h^2 - 192*a^4*b*c^3*
e^2*h^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^3 - 192*a^2*b^2*c^4*d^3
*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 + 60*a*b^5*c^2*d^2*g^2 +
 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960*a^2*b*c^5*d^2*e^2 + 24
0*a*b^3*c^4*d^2*e^2 + 256*a^4*c^4*e^2*f*h - 192*a^4*c^4*d*f^2*h + 16*b^6*c^
2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h + 80*a^4*b^3*c*f*h^3 +

6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^2*f + 512*a^3*b*c^4*e^3*g + 132*a*b^4*c
^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 + 2016*a^2*b*c^5*d^3*f -
496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5*c^2*d*f^3 - 192*a^4*b
^2*c^2*f^2*h^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 - 464*a^3*
b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2 - 240*a^
2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f^2 + 6*b^
7*c*d^2*f*h - 2*a*b^7*d*f*h^2 - 32*a^5*c^3*f^2*h^2 - 4*a^3*b^5*g^2*h^2 - 86
4*a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d^2*f^2 - 16*b^5*c^3*d^
2*e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 - 10*b^6*c^2*d^3*h + 6*a^3*b
^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3 - 4*b^7*c*d^2*g^2 + 30*b^
5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4 - 16*a^3*b^4*c*g^4 + 360*a
*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 16*a^4*c^4*f^4 - 256*a^3*c^
5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b^6*f^2*h^2 - b^8*d^2*h^2,
z, k)*x*(8192*a^6*b*c^6 + 32*a^2*b^9*c^2 - 512*a^3*b^7*c^3 + 3072*a^4*b^5*c
^4 - 8192*a^5*b^3*c^5))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^
2*c^2))) - (512*a^4*c^5*e*f - 32*a*b^5*c^3*d*e - 1024*a^3*b*c^5*d*e + 16*a*
b^6*c^2*d*g - 512*a^4*b*c^4*e*h - 256*a^4*b*c^4*f*g + 384*a^2*b^3*c^4*d*e -
192*a^2*b^4*c^3*d*g - 32*a^2*b^4*c^3*e*f + 512*a^3*b^2*c^4*d*g + 16*a^2*b^
5*c^2*f*g + 128*a^3*b^3*c^3*e*h - 64*a^3*b^4*c^2*g*h + 256*a^4*b^2*c^3*g*h)
/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (x*(2*b^6*c^3
*d^2 - 576*a^3*c^6*d^2 + 64*a^4*c^5*f^2 - 64*a^5*c^4*h^2 - 36*a*b^4*c^4*d^2
+ 128*a^3*b*c^5*e^2 + 2*a^2*b^6*c*h^2 + 256*a^2*b^2*c^5*d^2 - 32*a^2*b^3*c
^4*e^2 + 20*a^2*b^4*c^3*f^2 - 96*a^3*b^2*c^4*f^2 - 8*a^2*b^5*c^2*g^2 + 32*a
^3*b^3*c^3*g^2 - 4*a^3*b^4*c^2*h^2 - 384*a^4*c^5*d*h + 4*a*b^5*c^3*d*f + 32
0*a^3*b*c^5*d*f + 64*a^4*b*c^4*f*h - 96*a^2*b^3*c^4*d*f + 8*a^2*b^4*c^3*d*h
+ 32*a^2*b^4*c^3*e*g + 64*a^3*b^2*c^4*d*h - 128*a^3*b^2*c^4*e*g - 12*a^2*b
^5*c^2*f*h + 32*a^3*b^3*c^3*f*h))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c +
48*a^4*b^2*c^2))) - (x*(32*a^2*c^5*e^3 - 2*b^3*c^4*d^2*e + b^4*c^3*d^2*g -
4*a^2*b^3*c^2*g^3 + 24*a*b*c^5*d^2*e - 48*a^2*c^5*d*e*f - 16*a^3*c^4*e*f*h
- 12*a*b^2*c^4*d^2*g + 16*a^2*b*c^4*e*f^2 - 48*a^2*b*c^4*e^2*g + 8*a^3*b*c
^3*e*h^2 - a^2*b^4*c*g*h^2 + 24*a^2*b^2*c^3*e*g^2 - 8*a^2*b^2*c^3*f^2*g + 2
*a^2*b^3*c^2*e*h^2 - 4*a^3*b^2*c^2*g*h^2 - 4*a*b^2*c^4*d*e*f + 2*a*b^3*c^3*
d*f*g + 32*a^2*b*c^4*d*e*h + 24*a^2*b*c^4*d*f*g + 8*a^3*b*c^3*f*g*h - 16*a^
2*b^2*c^3*d*g*h - 12*a^2*b^2*c^3*e*f*h + 6*a^2*b^3*c^2*f*g*h))/(4*(a^2*b^6
- 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)))*root(1572864*a^8*b^2*c^6*z^
4 - 983040*a^7*b^4*c^5*z^4 + 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4
+ 6144*a^4*b^10*c^2*z^4 - 256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 192*a
^3*b^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a
^2*b^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^
4*b^6*c^2*f*h*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g*z^2 +
15360*a^4*b^5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*c^2*d*
h*z^2 - 512*a^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^3*b^6*
c^3*d*f*z^2 + 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 12288*a^
7*b*c^4*h^2*z^2 + 128*a^3*b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*
b^9*c*f^2*z^2 + 61440*a^5*b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 16384*a^7

*c^5*f*h*z^2 - 49152*a^6*c^6*d*f*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*
b^5*c^2*h^2*z^2 - 8192*a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 153
6*a^4*b^6*c^2*g^2*z^2 - 8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2
 + 24576*a^5*b^2*c^5*e^2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e
^2*z^2 - 61440*a^4*b^3*c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b
^7*c^3*d^2*z^2 - 16*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2
*z^2 - 6144*a^5*b*c^4*d*g*h*z + 96*a^2*b^7*c*d*g*h*z - 4096*a^4*b*c^5*d*e*f
*z + 64*a*b^7*c^2*d*e*f*z - 32*a*b^8*c*d*f*g*z + 4608*a^4*b^3*c^3*d*g*h*z -
 1152*a^3*b^5*c^2*d*g*h*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e
*h*z + 2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^
2*d*f*g*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^
5*c^3*d*e*f*z - 1024*a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 1024*a^5*b
*c^4*f^2*g*z - 32*a^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5
*d^2*g*z + 336*a*b^7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^5*c^5*d*
e*h*z + 768*a^5*b^3*c^2*g*h^2*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^
3*f^2*g*z + 384*a^4*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^
3*c^4*d^2*g*z - 2496*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a
^3*b^4*c^3*e*f^2*z + 32*a^2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4
992*a^2*b^4*c^4*d^2*e*z + 16*a^3*b^7*g*h^2*z + 2048*a^6*c^4*e*h^2*z - 2048*
a^5*c^5*e*f^2*z + 32*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2
*g*z - 256*a^4*b*c^3*e*f*g*h - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g
 - 192*a^3*b^3*c^2*e*f*g*h + 896*a^3*b^2*c^3*d*e*g*h - 96*a^2*b^4*c^2*d*e*g
*h - 192*a^2*b^3*c^3*d*e*f*g + 48*a^3*b^4*c*f*g^2*h + 16*a^3*b^4*c*e*g*h^2
+ 24*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d^2*f*h + 800*a^4*b*c^3*d*f*h^2 - 1
02*a*b^5*c^2*d^2*f*h - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a
*b^4*c^3*d^2*e*g - 32*a*b^4*c^3*d*e^2*f + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*
f*g^2 + 64*a^4*b^2*c^2*f*g^2*h + 192*a^4*b^2*c^2*e*g*h^2 - 224*a^3*b^3*c^2*
d*g^2*h + 192*a^3*b^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 336*a^3*b^3*c
^2*d*f*h^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4
*c^2*e*f^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^
3*c^3*d*e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g + 192*a^2*
b^2*c^4*d*e^2*f - 48*a^4*b^3*c*g^2*h^2 + 80*a^3*b^3*c^2*f^3*h - 42*a^3*b^4*
c*f^2*h^2 - 192*a^4*b*c^3*e^2*h^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d
*h^3 - 192*a^2*b^2*c^4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^
2 + 60*a*b^5*c^2*d^2*g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 -
960*a^2*b*c^5*d^2*e^2 + 240*a*b^3*c^4*d^2*e^2 + 256*a^4*c^4*e^2*f*h - 192*a
^4*c^4*d*f^2*h + 16*b^6*c^2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3
*h + 80*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^2*f + 512*a^3
*b*c^4*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^
2 + 2016*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4*d*f^3 - 18*a
*b^5*c^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*
b^3*c^2*e^2*h^2 - 464*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^
2*b^4*c^2*d^2*h^2 - 240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*
a^2*b^2*c^4*d^2*f^2 + 6*b^7*c*d^2*f*h - 2*a*b^7*d*f*h^2 - 32*a^5*c^3*f^2*h^
2 - 4*a^3*b^5*g^2*h^2 - 864*a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c

```
^5*d^2*f^2 - 16*b^5*c^3*d^2*e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 -
10*b^6*c^2*d^3*h + 6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3
 - 4*b^7*c*d^2*g^2 + 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4
- 16*a^3*b^4*c*g^4 + 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 1
6*a^4*c^4*f^4 - 256*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b
^6*f^2*h^2 - b^8*d^2*h^2, z, k), k, 1, 4)
```

3.40 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2} dx$

Optimal result

Integrand size = 40, antiderivative size = 468

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2} dx = \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{2acg-b(ce+ai)-(2c^2e-bcg+b^2i-2aci)x^2}{2c(b^2-4ac)(a+bx^2+cx^4)} + \frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce-bg+2ai)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$

```
[Out] 1/2*x*(b^2*d-a*b*f-2*a*(-a*h+c*d)+(a*b*h-2*a*c*f+b*c*d)*x^2)/a/(-4*a*c+b^2)
/(c*x^4+b*x^2+a)+1/2*(2*a*c*g-b*(a*i+c*e)-(-2*a*c*i+b^2*i-b*c*g+2*c^2*e)*x^
2)/c/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+(2*a*i-b*g+2*c*e)*arctanh((2*c*x^2+b)/(-4
*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(3/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c
+b^2)^(1/2))^(1/2))*(b*c*d-2*a*c*f+a*b*h+(4*a*b*c*f+b^2*(-a*h+c*d)-4*a*c*(a
*h+3*c*d))/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/c^(1/2)/(b-(-4*a*c+b^
2)^(1/2))^(1/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*
(b*c*d-2*a*c*f+a*b*h+(-4*a*b*c*f-b^2*(-a*h+c*d)+4*a*c*(a*h+3*c*d))/(-4*a*c+
b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.67 (sec) , antiderivative size = 468, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.225$, Rules used = {1687, 1692, 1180, 211, 1677, 1674, 12, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx$$
$$=\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{b^2(cd-ah)+4abcf-4ac(ah+3cd)}{\sqrt{b^2-4ac}}+abh-2acf+bcd\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{b^2(cd-ah)+4abcf-4ac(ah+3cd)}{\sqrt{b^2-4ac}}+abh-2acf+bcd\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)(2ai-bg+2ce)}{(b^2-4ac)^{3/2}}$$
$$+\frac{-(x^2(-2aci+b^2i-bcg+2c^2e))-b(ai+ce)+2acg}{2c\,(b^2-4ac)\,(a+bx^2+cx^4)}$$
$$+\frac{x(x^2(abh-2acf+bcd)-abf-2a(cd-ah)+b^2d)}{2a\,(b^2-4ac)\,(a+bx^2+cx^4)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] (x*(b^2*d - a*b*f - 2*a*(c*d - a*h) + (b*c*d - 2*a*c*f + a*b*h)*x^2))/(2*a*
(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (2*a*c*g - b*(c*e + a*i) - (2*c^2*e -
b*c*g + b^2*i - 2*a*c*i)*x^2)/(2*c*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + ((b
*c*d - 2*a*c*f + a*b*h + (4*a*b*c*f + b^2*(c*d - a*h) - 4*a*c*(3*c*d + a*h)
)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]
])/(2*Sqrt[2]*a*Sqrt[c]*(b^2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((b*c*
d - 2*a*c*f + a*b*h - (4*a*b*c*f + b^2*(c*d - a*h) - 4*a*c*(3*c*d + a*h))/S
qrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/
(2*Sqrt[2]*a*Sqrt[c]*(b^2 - 4*a*c)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((2*c*e -
 b*g + 2*a*i)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(3/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2+hx^4}{\left(a+bx^2+cx^4\right)^2}\,dx+\int \frac{x(e+gx^2+ix^4)}{\left(a+bx^2+cx^4\right)^2}\,dx\\
&= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}\\
&\quad+\frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{\left(a+bx+cx^2\right)^2}\,dx,x,x^2\right)\\
&\quad-\frac{\int \frac{-b^2d-abf+2a(3cd+ah)+(-bcd+2acf-abh)x^2}{a+bx^2+cx^4}\,dx}{2a\left(b^2-4ac\right)}\\
&= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}\\
&\quad+\frac{2acg-b(ce+ai)-\left(2c^2e-bcg+b^2i-2aci\right)x^2}{2c\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}-\frac{\text{Subst}\left(\int \frac{2ce-bg+2ai}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)}\\
&\quad+\frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)}\\
&\quad+\frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a\left(b^2-4ac\right)}\\
&= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{2a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}\\
&\quad+\frac{2acg-b(ce+ai)-\left(2c^2e-bcg+b^2i-2aci\right)x^2}{2c\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}\\
&\quad+\frac{\left(bcd-2acf+abh+\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{b-\sqrt{b^2-4ac}}}\\
&\quad+\frac{\left(bcd-2acf+abh-\frac{4abcf+b^2(cd-ah)-4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{b+\sqrt{b^2-4ac}}}\\
&\quad-\frac{(2ce-bg+2ai)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)}
\end{aligned}$$

$$= \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{2a(b^2 - 4ac)(a + bx^2 + cx^4)} + \frac{2acg - b(ce + ai) - (2c^2e - bcg + b^2i - 2aci)x^2}{2c(b^2 - 4ac)(a + bx^2 + cx^4)} + \frac{\left(bcd - 2acf + abh + \frac{4abcf + b^2(cd-ah) - 4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2 - 4ac)\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\left(bcd - 2acf + abh - \frac{4abcf + b^2(cd-ah) - 4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2 - 4ac)\sqrt{b + \sqrt{b^2 - 4ac}}} + \frac{(2ce - bg + 2ai)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b + 2cx^2\right)}{b^2 - 4ac}$$

$$= \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{2a(b^2 - 4ac)(a + bx^2 + cx^4)} + \frac{2acg - b(ce + ai) - (2c^2e - bcg + b^2i - 2aci)x^2}{2c(b^2 - 4ac)(a + bx^2 + cx^4)} + \frac{\left(bcd - 2acf + abh + \frac{4abcf + b^2(cd-ah) - 4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2 - 4ac)\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\left(bcd - 2acf + abh - \frac{4abcf + b^2(cd-ah) - 4ac(3cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2 - 4ac)\sqrt{b + \sqrt{b^2 - 4ac}}} + \frac{(2ce - bg + 2ai)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2 - 4ac)^{3/2}}$$

Mathematica [A] (verified)

Time = 1.21 (sec) , antiderivative size = 524, normalized size of antiderivative = 1.12

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx$$
$$=\frac{1}{4}\left(\frac{2(-bcdx(b+cx^2)+a^2(bi-2c(g+x(h+ix)))+a(b^2ix^2+2c^2x(d+x(e+fx))+bc(e+x(f-x(g}{ac\left(-b^2+4ac\right)\left(a+bx^2+cx^4\right)}\right.$$
$$+\frac{\sqrt{2}\big(b^2(cd-ah)-2ac\big(6cd+\sqrt{b^2-4ac}f+2ah\big)+b\big(c\sqrt{b^2-4ac}d+4acf+a\sqrt{b^2-4ac}h\big)\big)\arctan\Big(}{a\sqrt{c}\left(b^2-4ac\right)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{2}\big(b^2(-cd+ah)+2ac\big(6cd-\sqrt{b^2-4ac}f+2ah\big)+b\big(c\sqrt{b^2-4ac}d-4acf+a\sqrt{b^2-4ac}h\big)\big)\arctan}{a\sqrt{c}\left(b^2-4ac\right)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+\frac{2(-2ce+bg-2ai)\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{\left(b^2-4ac\right)^{3/2}}$$
$$\left.+\frac{2(2ce-bg+2ai)\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{\left(b^2-4ac\right)^{3/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^2,x
]
```

```
[Out] ((2*(-(b*c*d*x*(b + c*x^2)) + a^2*(b*i - 2*c*(g + x*(h + i*x))) + a*(b^2*i*
x^2 + 2*c^2*x*(d + x*(e + f*x)) + b*c*(e + x*(f - x*(g + h*x))))))/(a*c*(-b
^2 + 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt[2]*(b^2*(c*d - a*h) - 2*a*c*(6*c*d
 + Sqrt[b^2 - 4*a*c]*f + 2*a*h) + b*(c*Sqrt[b^2 - 4*a*c]*d + 4*a*c*f + a*Sq
rt[b^2 - 4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]]
)/(a*Sqrt[c]*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[2]*(b
^2*(-(c*d) + a*h) + 2*a*c*(6*c*d - Sqrt[b^2 - 4*a*c]*f + 2*a*h) + b*(c*Sqrt
[b^2 - 4*a*c]*d - 4*a*c*f + a*Sqrt[b^2 - 4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]
*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a*Sqrt[c]*(b^2 - 4*a*c)^(3/2)*Sqrt[b + S
qrt[b^2 - 4*a*c]]) + (2*(-2*c*e + b*g - 2*a*i)*Log[-b + Sqrt[b^2 - 4*a*c] -
 2*c*x^2])/(b^2 - 4*a*c)^(3/2) + (2*(2*c*e - b*g + 2*a*i)*Log[b + Sqrt[b^2
- 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.15 (sec) , antiderivative size = 304, normalized size of antiderivative = 0.65

method	result
risch	$\frac{-\frac{(abh-2acf+bcd)x^3}{2a(4ac-b^2)}-\frac{(2aci-b^2i+gbc-2e\,c^2)x^2}{2c(4ac-b^2)}-\frac{(2a^2h-abf-2acd+b^2d)x}{2a(4ac-b^2)}+\frac{abi-2acg+ebc}{2c(4ac-b^2)}}{c\,x^4+b\,x^2+a}+\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2b+a\right)}\frac{\left(-\frac{(abh-2}{a}\right.}{}\right.}{}$
default	$\frac{-\frac{(abh-2acf+bcd)x^3}{2a(4ac-b^2)}-\frac{(2aci-b^2i+gbc-2e\,c^2)x^2}{2c(4ac-b^2)}-\frac{(2a^2h-abf-2acd+b^2d)x}{2a(4ac-b^2)}+\frac{abi-2acg+ebc}{2c(4ac-b^2)}}{c\,x^4+b\,x^2+a}+\frac{2c\left(\frac{\left(8\sqrt{-4ac+b^2}\,a^2ci-4\sqrt{-4ac+b^2}\,abcg+8\sqrt{}\right.}{4}\right.}{}$

[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)

[Out] (-1/2/a*(a*b*h-2*a*c*f+b*c*d)/(4*a*c-b^2)*x^3-1/2*(2*a*c*i-b^2*i+b*c*g-2*c^2*e)/c/(4*a*c-b^2)*x^2-1/2*(2*a^2*h-a*b*f-2*a*c*d+b^2*d)/a/(4*a*c-b^2)*x+1/2/c*(a*b*i-2*a*c*g+b*c*e)/(4*a*c-b^2))/(c*x^4+b*x^2+a)+1/4*sum((-1/a*(a*b*h-2*a*c*f+b*c*d)/(4*a*c-b^2)*_R^2+2*(2*a*i-b*g+2*c*e)/(4*a*c-b^2)*_R+(2*a^2*h-a*b*f+6*a*c*d-b^2*d)/a/(4*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx = \text{Timed out}$$

[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas")

[Out] Timed out

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx = \int \frac{ix^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="m
axima")
[Out] -1/2*(a*b*c*e - 2*a^2*c*g + a^2*b*i - (b*c^2*d - 2*a*c^2*f + a*b*c*h)*x^3 +
 (2*a*c^2*e - a*b*c*g + (a*b^2 - 2*a^2*c)*i)*x^2 + (a*b*c*f - 2*a^2*c*h - (
b^2*c - 2*a*c^2)*d)*x)/(a^2*b^2*c - 4*a^3*c^2 + (a*b^2*c^2 - 4*a^2*c^3)*x^4
 + (a*b^3*c - 4*a^2*b*c^2)*x^2) + 1/2*integrate((a*b*f - 2*a^2*h + (b*c*d -
 2*a*c*f + a*b*h)*x^2 + (b^2 - 6*a*c)*d - 2*(2*a*c*e - a*b*g + 2*a^2*i)*x)/
(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 7962 vs. $2(420) = 840$.

Time = 1.91 (sec) , antiderivative size = 7962, normalized size of antiderivative = 17.01

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x, algorithm="g
iac")
[Out] 1/2*(b*c^2*d*x^3 - 2*a*c^2*f*x^3 + a*b*c*h*x^3 - 2*a*c^2*e*x^2 + a*b*c*g*x^
2 - a*b^2*i*x^2 + 2*a^2*c*i*x^2 + b^2*c*d*x - 2*a*c^2*d*x - a*b*c*f*x + 2*a
^2*c*h*x - a*b*c*e + 2*a^2*c*g - a^2*b*i)/((c*x^4 + b*x^2 + a)*(a*b^2*c - 4
*a^2*c^2)) + 1/16*((2*b^3*c^3 - 8*a*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
```

```
a*c)*c)*b*c^3 - 2*(b^2 - 4*a*c)*b*c^3)*(a*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c
^3 - 8*a^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a*b^2*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c
^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^3 - 2*(b^2 -
4*a*c)*a*c^3)*(a*b^2 - 4*a^2*c)^2*f + (2*a*b^3*c^2 - 8*a^2*b*c^3 - sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3 + 4*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c + 2*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - 2*(b^2 - 4*a*c)*a*b*c^2)*(a*b^2 -
4*a^2*c)^2*h + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 14*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 - 2*sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a*b^5*c^2 - 2*a*b^6*c^2 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a^3*b^2*c^3 + 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3
*c^3 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 28*a^2*b^4*c^3 -
 96*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 - 48*sqrt(2)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^2*b^2*c^4 - 128*a^3*b^2*c^4 + 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^3*c^5 + 192*a^4*c^5 + 2*(b^2 - 4*a*c)*a*b^4*c^2 - 20*(b^2 - 4*a*c)*a^2*
b^2*c^3 + 48*(b^2 - 4*a*c)*a^3*c^4)*d*abs(a*b^2 - 4*a^2*c) + 2*(sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^3*b^3*c^2 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2
 - 2*a^2*b^5*c^2 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + 8
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 + sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 16*a^3*b^3*c^3 - 4*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^3*b*c^4 - 32*a^4*b*c^4 + 2*(b^2 - 4*a*c)*a^2*b^3*c^2 -
8*(b^2 - 4*a*c)*a^3*b*c^3)*f*abs(a*b^2 - 4*a^2*c) - 4*(sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^4*b^2*c^2 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 - 2*a^3*
b^4*c^2 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*c^3 + 8*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^3*b^2*c^3 + 16*a^4*b^2*c^3 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^4*c^4 - 32*a^5*c^4 + 2*(b^2 - 4*a*c)*a^3*b^2*c^2 - 8*(b^2 - 4*a*c)*a
^4*c^3)*h*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^3 - 40*a^3*b^5*c^4 + 224*a^4*
b^3*c^5 - 384*a^5*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^2*b^7*c + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^3*b^5*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^2*b^6*c^2 - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^4*b^3*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^3*b^4*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^2*b^5*c^3 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^5*b*c^4 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^4*b^2*c^4 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^3*b^3*c^4 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^4*b*c^5 - 2*(b^2 - 4*a*c)*a^2*b^5*c^3 + 32*(b^2 - 4*a*c)*a^3*b^3*c^4 -
```

96*(b^2 - 4*a*c)*a^4*b*c^5)*d + 4*(2*a^3*b^6*c^3 - 16*a^4*b^4*c^4 + 32*a^5
*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^
6*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c
^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^
2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^
3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^3
 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^3 +
4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^4 - 2
*(b^2 - 4*a*c)*a^3*b^4*c^3 + 8*(b^2 - 4*a*c)*a^4*b^2*c^4)*f - (2*a^3*b^7*c^
2 - 8*a^4*b^5*c^3 - 32*a^5*b^3*c^4 + 128*a^6*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^7 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^6*b*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 - 2*(b^2 - 4*a*c)*a^3*b^5*c^2 + 32*(b^2 - 4*
a*c)*a^5*b*c^4)*h)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c + sqrt((a*b
^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c
- 4*a^2*c^2)))/((a^3*b^6*c - 12*a^4*b^4*c^2 - 2*a^3*b^5*c^2 + 48*a^5*b^2*c^
3 + 16*a^4*b^3*c^3 + a^3*b^4*c^3 - 64*a^6*c^4 - 32*a^5*b*c^4 - 8*a^4*b^2*c^
4 + 16*a^5*c^5)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/16*((2*b^3*c^3 - 8*a*b*c^4
 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c + 4*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + 2*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^2 - sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^3 - 2*(b^2 - 4*a*c)*b*c^3)*(a
*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^3 - 8*a^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a*c^3 - 2*(b^2 - 4*a*c)*a*c^3)*(a*b^2 - 4*a^2*c)^2*f + (2*
a*b^3*c^2 - 8*a^2*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^2*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b
^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 -
2*(b^2 - 4*a*c)*a*b*c^2)*(a*b^2 - 4*a^2*c)^2*h - 2*(sqrt(2)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a*b^6*c - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b
^4*c^2 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 2*a*b^6*c^2
+ 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 + 20*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^4*c^3 - 28*a^2*b^4*c^3 - 96*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^4*c^4 - 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 10*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 128*a^3*b^2*c^4 + 24*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 192*a^4*c^5 - 2*(b^2 - 4*a*c)

```
*a*b^4*c^2 + 20*(b^2 - 4*a*c)*a^2*b^2*c^3 - 48*(b^2 - 4*a*c)*a^3*c^4)*d*abs
(a*b^2 - 4*a^2*c) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c -
8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 - 2*sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 2*a^2*b^5*c^2 + 16*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^3*b^2*c^3 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 16*a^3*
b^3*c^3 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 32*a^4*b*c^
4 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 8*(b^2 - 4*a*c)*a^3*b*c^3)*f*abs(a*b^2 -
4*a^2*c) + 4*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 8*sqrt(2)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 - 2*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^3*b^3*c^2 + 2*a^3*b^4*c^2 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^5*c^3 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 +
 sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 16*a^4*b^2*c^3 - 4*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 32*a^5*c^4 - 2*(b^2 - 4*a*
c)*a^3*b^2*c^2 + 8*(b^2 - 4*a*c)*a^4*c^3)*h*abs(a*b^2 - 4*a^2*c) + (2*a^2*b
^7*c^3 - 40*a^3*b^5*c^4 + 224*a^4*b^3*c^5 - 384*a^5*b*c^6 - sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7*c + 20*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 + 2*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c^2 - 112*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^3 - 32*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^3 - sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^3 + 192*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 + 96*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^4 + 16*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^4 - 48*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^5 - 2*(b^2 - 4*a*c)*a^2*b^5*c^
3 + 32*(b^2 - 4*a*c)*a^3*b^3*c^4 - 96*(b^2 - 4*a*c)*a^4*b*c^5)*d + 4*(2*a^3
*b^6*c^3 - 16*a^4*b^4*c^4 + 32*a^5*b^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^3*b^4*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^4*b^2*c^4 - 2*(b^2 - 4*a*c)*a^3*b^4*c^3 + 8*(b^2 - 4*a
*c)*a^4*b^2*c^4)*f - (2*a^3*b^7*c^2 - 8*a^4*b^5*c^3 - 32*a^5*b^3*c^4 + 128*
a^6*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b
^7 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c
+ 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c + 1
6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^2 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^2 - 64*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^3 - 32*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^3 + 16*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^4 - 2*(b^2 -
 4*a*c)*a^3*b^5*c^2 + 32*(b^2 - 4*a*c)*a^5*b*c^4)*h)*arctan(2*sqrt(1/2)*x/s
```

```
qrt((a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)
*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2)))/((a^3*b^6*c - 12*a^4*b^4*c
^2 - 2*a^3*b^5*c^2 + 48*a^5*b^2*c^3 + 16*a^4*b^3*c^3 + a^3*b^4*c^3 - 64*a^6
*c^4 - 32*a^5*b*c^4 - 8*a^4*b^2*c^4 + 16*a^5*c^5)*abs(a*b^2 - 4*a^2*c)*abs(
c)) - 1/8*(2*(b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2 - 4*a*c^3
 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (b^4*c - 4*a*b
^2*c^2 - 2*b^3*c^2 + b^2*c^3 + (b^3*c - 4*a*b*c^2 - 2*b^2*c^2 + b*c^3)*sqrt
(b^2 - 4*a*c))*g*abs(a*b^2 - 4*a^2*c) + 2*(a*b^3*c - 4*a^2*b*c^2 - 2*a*b^2*
c^2 + a*b*c^3 + (a*b^2*c - 4*a^2*c^2 - 2*a*b*c^2 + a*c^3)*sqrt(b^2 - 4*a*c)
)*i*abs(a*b^2 - 4*a^2*c) - 2*(a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*
a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^
2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (a*b^6*c - 8*a^2*b^
4*c^2 - 2*a*b^5*c^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3*c^3 + a*b^4*c^3 - 4*a^2*b^
2*c^4 - (a*b^5*c - 4*a^2*b^3*c^2 - 2*a*b^4*c^2 + a*b^3*c^3)*sqrt(b^2 - 4*a*
c))*g - 2*(a^2*b^5*c - 8*a^3*b^3*c^2 - 2*a^2*b^4*c^2 + 16*a^4*b*c^3 + 8*a^3
*b^2*c^3 + a^2*b^3*c^3 - 4*a^3*b*c^4 + (a^2*b^4*c - 4*a^3*b^2*c^2 - 2*a^2*b
^3*c^2 + a^2*b^2*c^3)*sqrt(b^2 - 4*a*c))*i)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*
c + sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2
)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 +
 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c)) - 1/8*(2*(b
^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4
)*sqrt(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (b^4*c - 4*a*b^2*c^2 - 2*b^3*
c^2 + b^2*c^3 - (b^3*c - 4*a*b*c^2 - 2*b^2*c^2 + b*c^3)*sqrt(b^2 - 4*a*c))*
g*abs(a*b^2 - 4*a^2*c) + 2*(a*b^3*c - 4*a^2*b*c^2 - 2*a*b^2*c^2 + a*b*c^3 -
 (a*b^2*c - 4*a^2*c^2 - 2*a*b*c^2 + a*c^3)*sqrt(b^2 - 4*a*c))*i*abs(a*b^2 -
 4*a^2*c) - 2*(a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*a
^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 + (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^3
*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e + (a*b^6*c - 8*a^2*b^4*c^2 - 2*a*b^5
*c^2 + 16*a^3*b^2*c^3 + 8*a^2*b^3*c^3 + a*b^4*c^3 - 4*a^2*b^2*c^4 - (a*b^5*
c - 4*a^2*b^3*c^2 - 2*a*b^4*c^2 + a*b^3*c^3)*sqrt(b^2 - 4*a*c))*g - 2*(a^2*
b^5*c - 8*a^3*b^3*c^2 - 2*a^2*b^4*c^2 + 16*a^4*b*c^3 + 8*a^3*b^2*c^3 + a^2*
b^3*c^3 - 4*a^3*b*c^4 - (a^2*b^4*c - 4*a^3*b^2*c^2 - 2*a^2*b^3*c^2 + a^2*b^
2*c^3)*sqrt(b^2 - 4*a*c))*i)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c - sqrt((a*b^3
 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c -
4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 +
a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 9.41 (sec) , antiderivative size = 18449, normalized size of antiderivative = 39.42

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] ((b*c*e - 2*a*c*g + a*b*i)/(2*c*(4*a*c - b^2)) - (x*(b^2*d + 2*a^2*h - 2*a*
c*d - a*b*f))/(2*a*(4*a*c - b^2)) + (x^2*(2*c^2*e + b^2*i - b*c*g - 2*a*c*i
))/(2*c*(4*a*c - b^2)) - (x^3*(b*c*d - 2*a*c*f + a*b*h))/(2*a*(4*a*c - b^2)
))/(a + b*x^2 + c*x^4) + symsum(log((5*b^3*c^4*d^3 + 8*a^3*c^4*f^3 - 96*a^2
*c^5*d*e^2 + 72*a^2*c^5*d^2*f - 3*a^3*b^3*c*h^3 - 4*a^4*b*c^2*h^3 - 3*b^4*c
^3*d^2*f - 32*a^3*c^4*e^2*h - 96*a^4*c^3*d*i^2 + b^5*c^2*d^2*h + 8*a^4*c^3*
f*h^2 - 32*a^5*c^2*h*i^2 + 6*a^2*b^2*c^3*f^3 - 36*a*b*c^5*d^3 + a*b^5*c*d*h
^2 - 192*a^3*c^4*d*e*i + 48*a^3*c^4*d*f*h - 64*a^4*c^3*e*h*i + 16*a*b^2*c^4
*d*e^2 + 18*a*b^2*c^4*d^2*f + 3*a*b^3*c^3*d*f^2 - 60*a^2*b*c^4*d*f^2 + 4*a*
b^4*c^2*d*g^2 + 16*a^2*b*c^4*e^2*f - a*b^3*c^3*d^2*h - 60*a^2*b*c^4*d^2*h -
 28*a^3*b*c^3*d*h^2 + a^2*b^4*c*f*h^2 - 28*a^3*b*c^3*f^2*h + 16*a^4*b*c^2*f
*i^2 - 24*a^2*b^2*c^3*d*g^2 - 9*a^2*b^3*c^2*d*h^2 + 4*a^2*b^3*c^2*f*g^2 + 1
6*a^3*b^2*c^2*d*i^2 - 5*a^2*b^3*c^2*f^2*h + 18*a^3*b^2*c^2*f*h^2 - 8*a^3*b^
2*c^2*g^2*h - 16*a*b^3*c^3*d*e*g + 96*a^2*b*c^4*d*e*g - 4*a*b^4*c^2*d*f*h +
 96*a^3*b*c^3*d*g*i + 32*a^3*b*c^3*e*f*i + 32*a^3*b*c^3*e*g*h + 32*a^4*b*c^
2*g*h*i + 32*a^2*b^2*c^3*d*e*i + 52*a^2*b^2*c^3*d*f*h - 16*a^2*b^2*c^3*e*f*
g - 16*a^2*b^3*c^2*d*g*i - 16*a^3*b^2*c^2*f*g*i)/(8*(a^2*b^6 - 64*a^5*c^3 -
 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - root(1572864*a^8*b^2*c^6*z^4 - 983040*a^
7*b^4*c^5*z^4 + 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b
^10*c^2*z^4 - 256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 32768*a^7*b*c^4*g*
i*z^2 - 512*a^4*b^7*c*g*i*z^2 + 192*a^3*b^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h
*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^2*b^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2
 - 24576*a^6*b^3*c^3*g*i*z^2 + 6144*a^5*b^5*c^2*g*i*z^2 + 49152*a^6*b^2*c^4
*e*i*z^2 - 12288*a^5*b^4*c^3*e*i*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*
b^6*c^2*f*h*z^2 + 1024*a^4*b^6*c^2*e*i*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 24
576*a^5*b^3*c^4*e*g*z^2 + 15360*a^4*b^5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*
z^2 - 2048*a^3*b^7*c^2*d*h*z^2 - 512*a^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^
5*d*f*z^2 - 3072*a^3*b^6*c^3*d*f*z^2 + 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b
^8*c^2*d*f*z^2 + 512*a^5*b^6*c*i^2*z^2 + 12288*a^7*b*c^4*h^2*z^2 + 128*a^3*
b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*c*f^2*z^2 + 61440*a^5*
b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 65536*a^7*c^5*e*i*z^2 - 16384*a^7*c
^5*f*h*z^2 - 49152*a^6*c^6*d*f*z^2 + 24576*a^7*b^2*c^3*i^2*z^2 - 6144*a^6*b
^4*c^2*i^2*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8192
*a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z^2
 - 8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^5*e
^2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*b^3
```

*c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 32768*a^8*c^4*i^2*z^2 - 16*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2 - 192*a^3*b^6*c*d*h*i*z - 6144*a^5*b*c^4*d*g*h*z - 4096*a^5*b*c^4*d*f*i*z + 96*a^2*b^7*c*d*g*h*z + 64*a^2*b^7*c*d*f*i*z - 4096*a^4*b*c^5*d*e*f*z + 64*a*b^7*c^2*d*e*f*z - 32*a*b^8*c*d*f*g*z - 9216*a^5*b^2*c^3*d*h*i*z + 2304*a^4*b^4*c^2*d*h*i*z + 4608*a^4*b^3*c^3*d*g*h*z + 3072*a^4*b^3*c^3*d*f*i*z - 1152*a^3*b^5*c^2*d*g*h*z - 768*a^3*b^5*c^2*d*f*i*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z + 2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*f*g*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^3*d*e*f*z + 384*a^5*b^4*c*h^2*i*z - 1024*a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 32*a^3*b^6*c*f^2*i*z + 1024*a^5*b*c^4*f^2*g*z - 32*a^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2*g*z + 336*a*b^7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^6*c^4*d*h*i*z + 12288*a^5*c^5*d*e*h*z + 32*a*b^8*c*d^2*i*z - 1536*a^6*b^2*c^2*h^2*i*z + 1536*a^5*b^2*c^3*f^2*i*z + 768*a^5*b^3*c^2*g*h^2*z - 384*a^4*b^4*c^2*f^2*i*z - 15872*a^4*b^2*c^4*d^2*i*z + 4992*a^3*b^4*c^3*d^2*i*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^2*g*z - 672*a^2*b^6*c^2*d^2*i*z + 384*a^4*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 2496*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*z + 32*a^2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d^2*e*z + 2048*a^7*c^3*h^2*i*z - 32*a^4*b^6*h^2*i*z - 2048*a^6*c^4*f^2*i*z + 16*a^3*b^7*g*h^2*z + 18432*a^5*c^5*d^2*i*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*c^5*e*f^2*z + 32*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z - 256*a^5*b*c^2*f*g*h*i - 192*a^4*b^3*c*f*g*h*i - 96*a^3*b^4*c*d*g*h*i - 1792*a^4*b*c^3*d*e*h*i - 768*a^4*b*c^3*d*f*g*i - 256*a^4*b*c^3*e*f*g*h + 32*a^2*b^5*c*d*f*g*i - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g + 896*a^4*b^2*c^2*d*g*h*i + 384*a^4*b^2*c^2*e*f*h*i - 192*a^3*b^3*c^2*e*f*g*h - 192*a^3*b^3*c^2*d*f*g*i + 192*a^3*b^3*c^2*d*e*h*i + 896*a^3*b^2*c^3*d*e*g*h + 384*a^3*b^2*c^3*d*e*f*i - 96*a^2*b^4*c^2*d*e*g*h - 64*a^2*b^4*c^2*d*e*f*i - 192*a^2*b^3*c^3*d*e*f*g + 192*a^5*b^2*c*g*h^2*i + 192*a^5*b^2*c*f*h*i^2 - 384*a^5*b*c^2*e*h^2*i - 32*a^4*b^3*c*e*h^2*i + 16*a^3*b^4*c*f^2*g*i + 1536*a^5*b*c^2*e*g*i^2 + 1536*a^4*b*c^3*e^2*g*i - 896*a^5*b*c^2*d*h*i^2 + 96*a^4*b^3*c*d*h*i^2 + 48*a^3*b^4*c*f*g^2*h - 384*a^4*b*c^3*e*f^2*i + 16*a^3*b^4*c*e*g*h^2 - 32*a^3*b^4*c*d*f*i^2 + 24*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d^2*f*h - 1920*a^3*b*c^4*d^2*e*i + 800*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2*f*h - 32*a*b^5*c^2*d^2*e*i - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4*c^3*d^2*e*g - 32*a*b^4*c^3*d*e^2*f + 512*a^5*c^3*e*f*h*i + 1536*a^4*c^4*d*e*f*i + 16*a*b^6*c*d^2*g*i + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^2 + 192*a^4*b^2*c^2*f^2*g*i - 768*a^4*b^2*c^2*e*g^2*i + 64*a^4*b^2*c^2*f*g^2*h + 960*a^3*b^2*c^3*d^2*g*i - 240*a^2*b^4*c^2*d^2*g*i + 192*a^4*b^2*c^2*e*g*h^2 - 32*a^3*b^3*c^2*e*f^2*i - 224*a^3*b^3*c^2*d*g^2*h + 192*a^4*b^2*c^2*d*f*i^2 + 192*a^3*b^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 480*a^2*b^3*c^3*d^2*e*i + 336*a^3*b^3*c^2*d*f*h^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2*e*f^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^3*d*e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*

a^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*c^4*d*e^2*f - 384*a^5*b^2*c*g^2*i^2 - 192
*a^5*b*c^2*f^2*i^2 - 48*a^4*b^3*c*g^2*h^2 - 16*a^4*b^3*c*f^2*i^2 + 80*a^3*b
^3*c^2*f^3*h - 42*a^3*b^4*c*f^2*h^2 - 960*a^4*b*c^3*d^2*i^2 - 192*a^4*b*c^3
*e^2*h^2 - 16*a^2*b^5*c*d^2*i^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h
^3 - 192*a^2*b^2*c^4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2
+ 60*a*b^5*c^2*d^2*g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 96
0*a^2*b*c^5*d^2*e^2 + 240*a*b^3*c^4*d^2*e^2 + 256*a^6*c^2*f*h*i^2 + 16*a^4*
b^4*g*h^2*i + 768*a^5*c^3*d*f*i^2 + 256*a^4*c^4*e^2*f*h - 192*a^6*b*c*h^2*i
^2 - 192*a^4*c^4*d*f^2*h + 128*a^4*b^3*c*g^3*i + 16*b^6*c^2*d^2*e*g + 96*a^
5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h + 80*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h
 + 768*a^3*c^5*d*e^2*f + 512*a^3*b*c^4*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3
*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 + 2016*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^
3*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5*c^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 +
240*a^3*b^3*c^2*d^2*i^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 -
 464*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2
 - 240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f
^2 + 6*b^7*c*d^2*f*h + 512*a^6*b*c*g*i^3 - 2*a*b^7*d*f*h^2 - 16*a^5*b^3*h^2
*i^2 - 1536*a^5*c^3*e^2*i^2 - 32*a^5*c^3*f^2*h^2 - 4*a^3*b^5*g^2*h^2 - 864*
a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d^2*f^2 - 16*b^5*c^3*d^2*
e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 - 1024*a^6*c^2*e*i^3 - 1024*a^
4*c^4*e^3*i - 10*b^6*c^2*d^3*h + 6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192
*a^5*c^3*d*h^3 - 4*b^7*c*d^2*g^2 + 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*
a^5*b^2*c*h^4 - 16*a^3*b^4*c*g^4 + 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a
^4*b^4*h^4 - 16*a^4*c^4*f^4 - 256*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c
^6*d^4 - a^2*b^6*f^2*h^2 - 256*a^7*c*i^4 - b^8*d^2*h^2, z, l)*((32*a*b^5*c^
3*d*e - 512*a^5*c^4*f*i - 512*a^4*c^5*e*f + 1024*a^3*b*c^5*d*e - 16*a*b^6*c
^2*d*g + 1024*a^4*b*c^4*d*i + 512*a^4*b*c^4*e*h + 256*a^4*b*c^4*f*g + 512*a
^5*b*c^3*h*i - 384*a^2*b^3*c^4*d*e + 192*a^2*b^4*c^3*d*g + 32*a^2*b^4*c^3*e
*f - 512*a^3*b^2*c^4*d*g + 32*a^2*b^5*c^2*d*i - 16*a^2*b^5*c^2*f*g - 384*a^
3*b^3*c^3*d*i - 128*a^3*b^3*c^3*e*h + 32*a^3*b^4*c^2*f*i + 64*a^3*b^4*c^2*g
*h - 256*a^4*b^2*c^3*g*h - 128*a^4*b^3*c^2*h*i)/(8*(a^2*b^6 - 64*a^5*c^3 -
12*a^3*b^4*c + 48*a^4*b^2*c^2)) + root(1572864*a^8*b^2*c^6*z^4 - 983040*a^7
*b^4*c^5*z^4 + 327680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b^
10*c^2*z^4 - 256*a^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 32768*a^7*b*c^4*g*i
*z^2 - 512*a^4*b^7*c*g*i*z^2 + 192*a^3*b^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h*
z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^2*b^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2
- 24576*a^6*b^3*c^3*g*i*z^2 + 6144*a^5*b^5*c^2*g*i*z^2 + 49152*a^6*b^2*c^4*
e*i*z^2 - 12288*a^5*b^4*c^3*e*i*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b
^6*c^2*f*h*z^2 + 1024*a^4*b^6*c^2*e*i*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 245
76*a^5*b^3*c^4*e*g*z^2 + 15360*a^4*b^5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z
^2 - 2048*a^3*b^7*c^2*d*h*z^2 - 512*a^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^5
*d*f*z^2 - 3072*a^3*b^6*c^3*d*f*z^2 + 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^
8*c^2*d*f*z^2 + 512*a^5*b^6*c*i^2*z^2 + 12288*a^7*b*c^4*h^2*z^2 + 128*a^3*b
^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*c*f^2*z^2 + 61440*a^5*b
*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2 - 65536*a^7*c^5*e*i*z^2 - 16384*a^7*c^

5*f*h*z^2 - 49152*a^6*c^6*d*f*z^2 + 24576*a^7*b^2*c^3*i^2*z^2 - 6144*a^6*b^
4*c^2*i^2*z^2 - 8192*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8192*
a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z^2 -
8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^5*e^
2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*b^3*
c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 32768*
a^8*c^4*i^2*z^2 - 16*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^
2*z^2 - 192*a^3*b^6*c*d*h*i*z - 6144*a^5*b*c^4*d*g*h*z - 4096*a^5*b*c^4*d*f
*i*z + 96*a^2*b^7*c*d*g*h*z + 64*a^2*b^7*c*d*f*i*z - 4096*a^4*b*c^5*d*e*f*z
+ 64*a*b^7*c^2*d*e*f*z - 32*a*b^8*c*d*f*g*z - 9216*a^5*b^2*c^3*d*h*i*z + 2
304*a^4*b^4*c^2*d*h*i*z + 4608*a^4*b^3*c^3*d*g*h*z + 3072*a^4*b^3*c^3*d*f*i
*z - 1152*a^3*b^5*c^2*d*g*h*z - 768*a^3*b^5*c^2*d*f*i*z - 9216*a^4*b^2*c^4*
d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z + 2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^
4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*f*g*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^
3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^3*d*e*f*z + 384*a^5*b^4*c*h^2*i*z - 1024*
a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h^2*z + 32*a^3*b^6*c*f^2*i*z + 1024*a^5
*b*c^4*f^2*g*z - 32*a^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c
^5*d^2*g*z + 336*a*b^7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^6*c^4*
d*h*i*z + 12288*a^5*c^5*d*e*h*z + 32*a*b^8*c*d^2*i*z - 1536*a^6*b^2*c^2*h^2
*i*z + 1536*a^5*b^2*c^3*f^2*i*z + 768*a^5*b^3*c^2*g*h^2*z - 384*a^4*b^4*c^2
*f^2*i*z - 15872*a^4*b^2*c^4*d^2*i*z + 4992*a^3*b^4*c^3*d^2*i*z - 1536*a^5*
b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^2*g*z - 672*a^2*b^6*c^2*d^2*i*z + 384*a
^4*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 2
496*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*
z + 32*a^2*b^6*c^2*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d
^2*e*z + 2048*a^7*c^3*h^2*i*z - 32*a^4*b^6*h^2*i*z - 2048*a^6*c^4*f^2*i*z +
16*a^3*b^7*g*h^2*z + 18432*a^5*c^5*d^2*i*z + 2048*a^6*c^4*e*h^2*z - 2048*a
^5*c^5*e*f^2*z + 32*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*
g*z - 256*a^5*b*c^2*f*g*h*i - 192*a^4*b^3*c*f*g*h*i - 96*a^3*b^4*c*d*g*h*i
- 1792*a^4*b*c^3*d*e*h*i - 768*a^4*b*c^3*d*f*g*i - 256*a^4*b*c^3*e*f*g*h +
32*a^2*b^5*c*d*f*g*i - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g + 896*a
^4*b^2*c^2*d*g*h*i + 384*a^4*b^2*c^2*e*f*h*i - 192*a^3*b^3*c^2*e*f*g*h - 19
2*a^3*b^3*c^2*d*f*g*i + 192*a^3*b^3*c^2*d*e*h*i + 896*a^3*b^2*c^3*d*e*g*h +
384*a^3*b^2*c^3*d*e*f*i - 96*a^2*b^4*c^2*d*e*g*h - 64*a^2*b^4*c^2*d*e*f*i
- 192*a^2*b^3*c^3*d*e*f*g + 192*a^5*b^2*c*g*h^2*i + 192*a^5*b^2*c*f*h*i^2 -
384*a^5*b*c^2*e*h^2*i - 32*a^4*b^3*c*e*h^2*i + 16*a^3*b^4*c*f^2*g*i + 1536
*a^5*b*c^2*e*g*i^2 + 1536*a^4*b*c^3*e^2*g*i - 896*a^5*b*c^2*d*h*i^2 + 96*a^
4*b^3*c*d*h*i^2 + 48*a^3*b^4*c*f*g^2*h - 384*a^4*b*c^3*e*f^2*i + 16*a^3*b^4
*c*e*g*h^2 - 32*a^3*b^4*c*d*f*i^2 + 24*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d
^2*f*h - 1920*a^3*b*c^4*d^2*e*i + 800*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2
*f*h - 32*a*b^5*c^2*d^2*e*i - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h
- 240*a*b^4*c^3*d^2*e*g - 32*a*b^4*c^3*d*e^2*f + 512*a^5*c^3*e*f*h*i + 1536
*a^4*c^4*d*e*f*i + 16*a*b^6*c*d^2*g*i + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*
g^2 + 192*a^4*b^2*c^2*f^2*g*i - 768*a^4*b^2*c^2*e*g^2*i + 64*a^4*b^2*c^2*f*
g^2*h + 960*a^3*b^2*c^3*d^2*g*i - 240*a^2*b^4*c^2*d^2*g*i + 192*a^4*b^2*c^2

*e*g*h^2 - 32*a^3*b^3*c^2*e*f^2*i - 224*a^3*b^3*c^2*d*g^2*h + 192*a^4*b^2*c
^2*d*f*i^2 + 192*a^3*b^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 480*a^2*b^
3*c^3*d^2*e*i + 336*a^3*b^3*c^2*d*f*h^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2
*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2*e*f^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^
3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^3*d*e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*a
^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*c^4*d*e^2*f - 384*a^5*b^2*c*g^2*i^2 - 192*
a^5*b*c^2*f^2*i^2 - 48*a^4*b^3*c*g^2*h^2 - 16*a^4*b^3*c*f^2*i^2 + 80*a^3*b^
3*c^2*f^3*h - 42*a^3*b^4*c*f^2*h^2 - 960*a^4*b*c^3*d^2*i^2 - 192*a^4*b*c^3*
e^2*h^2 - 16*a^2*b^5*c*d^2*i^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^
3 - 192*a^2*b^2*c^4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 +
 60*a*b^5*c^2*d^2*g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960
*a^2*b*c^5*d^2*e^2 + 240*a*b^3*c^4*d^2*e^2 + 256*a^6*c^2*f*h*i^2 + 16*a^4*b
^4*g*h^2*i + 768*a^5*c^3*d*f*i^2 + 256*a^4*c^4*e^2*f*h - 192*a^6*b*c*h^2*i^
2 - 192*a^4*c^4*d*f^2*h + 128*a^4*b^3*c*g^3*i + 16*b^6*c^2*d^2*e*g + 96*a^5
*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h + 80*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h
+ 768*a^3*c^5*d*e^2*f + 512*a^3*b*c^4*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3*
b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 + 2016*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^3
*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5*c^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 + 2
40*a^3*b^3*c^2*d^2*i^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 -
464*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2
- 240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f^
2 + 6*b^7*c*d^2*f*h + 512*a^6*b*c*g*i^3 - 2*a*b^7*d*f*h^2 - 16*a^5*b^3*h^2*
i^2 - 1536*a^5*c^3*e^2*i^2 - 32*a^5*c^3*f^2*h^2 - 4*a^3*b^5*g^2*h^2 - 864*a
^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d^2*f^2 - 16*b^5*c^3*d^2*e
^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c^2*f^4 - 1024*a^6*c^2*e*i^3 - 1024*a^4
*c^4*e^3*i - 10*b^6*c^2*d^3*h + 6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*
a^5*c^3*d*h^3 - 4*b^7*c*d^2*g^2 + 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a
^5*b^2*c*h^4 - 16*a^3*b^4*c*g^4 + 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^
4*b^4*h^4 - 16*a^4*c^4*f^4 - 256*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^
6*d^4 - a^2*b^6*f^2*h^2 - 256*a^7*c*i^4 - b^8*d^2*h^2, z, l)*((x*(2048*a^5*
c^6*e + 2048*a^6*c^5*i - 32*a^2*b^6*c^3*e + 384*a^3*b^4*c^4*e - 1536*a^4*b^
2*c^5*e + 16*a^2*b^7*c^2*g - 192*a^3*b^5*c^3*g + 768*a^4*b^3*c^4*g - 32*a^3
*b^6*c^2*i + 384*a^4*b^4*c^3*i - 1536*a^5*b^2*c^4*i - 1024*a^5*b*c^5*g))/(4
*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - (6144*a^5*c^6*d
+ 2048*a^6*c^5*h - 288*a^2*b^6*c^3*d + 1920*a^3*b^4*c^4*d - 5632*a^4*b^2*c^
5*d + 16*a^2*b^7*c^2*f - 192*a^3*b^5*c^3*f + 768*a^4*b^3*c^4*f - 32*a^3*b^6
*c^2*h + 384*a^4*b^4*c^3*h - 1536*a^5*b^2*c^4*h + 16*a*b^8*c^2*d - 1024*a^5
*b*c^5*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (roo
t(1572864*a^8*b^2*c^6*z^4 - 983040*a^7*b^4*c^5*z^4 + 327680*a^6*b^6*c^4*z^4
 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b^10*c^2*z^4 - 256*a^3*b^12*c*z^4 - 104
8576*a^9*c^7*z^4 + 32768*a^7*b*c^4*g*i*z^2 - 512*a^4*b^7*c*g*i*z^2 + 192*a^
3*b^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h*z^2 + 32768*a^6*b*c^5*e*g*z^2 + 96*a^
2*b^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2 - 24576*a^6*b^3*c^3*g*i*z^2 + 6144*a^
5*b^5*c^2*g*i*z^2 + 49152*a^6*b^2*c^4*e*i*z^2 - 12288*a^5*b^4*c^3*e*i*z^2 +
 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b^6*c^2*f*h*z^2 + 1024*a^4*b^6*c^2*e*i

*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g*z^2 + 15360*a^4*b^
5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*c^2*d*h*z^2 - 512*a
^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^3*b^6*c^3*d*f*z^2 +
 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 512*a^5*b^6*c*i^2*z^2
 + 12288*a^7*b*c^4*h^2*z^2 + 128*a^3*b^8*c*g^2*z^2 + 12288*a^6*b*c^5*f^2*z^
2 - 16*a^2*b^9*c*f^2*z^2 + 61440*a^5*b*c^6*d^2*z^2 + 432*a*b^9*c^2*d^2*z^2
- 65536*a^7*c^5*e*i*z^2 - 16384*a^7*c^5*f*h*z^2 - 49152*a^6*c^6*d*f*z^2 + 2
4576*a^7*b^2*c^3*i^2*z^2 - 6144*a^6*b^4*c^2*i^2*z^2 - 8192*a^6*b^3*c^3*h^2*
z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8192*a^6*b^2*c^4*g^2*z^2 + 6144*a^5*b^4*c^
3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z^2 - 8192*a^5*b^3*c^4*f^2*z^2 + 1536*a^4*
b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^5*e^2*z^2 - 6144*a^4*b^4*c^4*e^2*z^2 + 51
2*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*b^3*c^5*d^2*z^2 + 24064*a^3*b^5*c^4*d^2*z
^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 32768*a^8*c^4*i^2*z^2 - 16*a^3*b^9*h^2*z^2
- 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2 - 192*a^3*b^6*c*d*h*i*z - 6144*
a^5*b*c^4*d*g*h*z - 4096*a^5*b*c^4*d*f*i*z + 96*a^2*b^7*c*d*g*h*z + 64*a^2*
b^7*c*d*f*i*z - 4096*a^4*b*c^5*d*e*f*z + 64*a*b^7*c^2*d*e*f*z - 32*a*b^8*c*
d*f*g*z - 9216*a^5*b^2*c^3*d*h*i*z + 2304*a^4*b^4*c^2*d*h*i*z + 4608*a^4*b^
3*c^3*d*g*h*z + 3072*a^4*b^3*c^3*d*f*i*z - 1152*a^3*b^5*c^2*d*g*h*z - 768*a
^3*b^5*c^2*d*f*i*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3*b^4*c^3*d*e*h*z +
2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 384*a^2*b^6*c^2*d*f*g
*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z - 768*a^2*b^5*c^3*d
*e*f*z + 384*a^5*b^4*c*h^2*i*z - 1024*a^6*b*c^3*g*h^2*z - 192*a^4*b^5*c*g*h
^2*z + 32*a^3*b^6*c*f^2*i*z + 1024*a^5*b*c^4*f^2*g*z - 32*a^3*b^6*c*e*h^2*z
 - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2*g*z + 336*a*b^7*c^2*d^2*g*z -
672*a*b^6*c^3*d^2*e*z + 12288*a^6*c^4*d*h*i*z + 12288*a^5*c^5*d*e*h*z + 32*
a*b^8*c*d^2*i*z - 1536*a^6*b^2*c^2*h^2*i*z + 1536*a^5*b^2*c^3*f^2*i*z + 768
*a^5*b^3*c^2*g*h^2*z - 384*a^4*b^4*c^2*f^2*i*z - 15872*a^4*b^2*c^4*d^2*i*z
+ 4992*a^3*b^4*c^3*d^2*i*z - 1536*a^5*b^2*c^3*e*h^2*z - 768*a^4*b^3*c^3*f^2
*g*z - 672*a^2*b^6*c^2*d^2*i*z + 384*a^4*b^4*c^2*e*h^2*z + 192*a^3*b^5*c^2*
f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 2496*a^2*b^5*c^3*d^2*g*z + 1536*a^4*b^
2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*z + 32*a^2*b^6*c^2*e*f^2*z - 15872*a^
3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d^2*e*z + 2048*a^7*c^3*h^2*i*z - 32*a^
4*b^6*h^2*i*z - 2048*a^6*c^4*f^2*i*z + 16*a^3*b^7*g*h^2*z + 18432*a^5*c^5*d
^2*i*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*c^5*e*f^2*z + 32*b^8*c^2*d^2*e*z +
 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z - 256*a^5*b*c^2*f*g*h*i - 192*a^4
*b^3*c*f*g*h*i - 96*a^3*b^4*c*d*g*h*i - 1792*a^4*b*c^3*d*e*h*i - 768*a^4*b*
c^3*d*f*g*i - 256*a^4*b*c^3*e*f*g*h + 32*a^2*b^5*c*d*f*g*i - 768*a^3*b*c^4*
d*e*f*g + 32*a*b^5*c^2*d*e*f*g + 896*a^4*b^2*c^2*d*g*h*i + 384*a^4*b^2*c^2*
e*f*h*i - 192*a^3*b^3*c^2*e*f*g*h - 192*a^3*b^3*c^2*d*f*g*i + 192*a^3*b^3*c
^2*d*e*h*i + 896*a^3*b^2*c^3*d*e*g*h + 384*a^3*b^2*c^3*d*e*f*i - 96*a^2*b^4
*c^2*d*e*g*h - 64*a^2*b^4*c^2*d*e*f*i - 192*a^2*b^3*c^3*d*e*f*g + 192*a^5*b
^2*c*g*h^2*i + 192*a^5*b^2*c*f*h*i^2 - 384*a^5*b*c^2*e*h^2*i - 32*a^4*b^3*c
*e*h^2*i + 16*a^3*b^4*c*f^2*g*i + 1536*a^5*b*c^2*e*g*i^2 + 1536*a^4*b*c^3*e
^2*g*i - 896*a^5*b*c^2*d*h*i^2 + 96*a^4*b^3*c*d*h*i^2 + 48*a^3*b^4*c*f*g^2*
h - 384*a^4*b*c^3*e*f^2*i + 16*a^3*b^4*c*e*g*h^2 - 32*a^3*b^4*c*d*f*i^2 + 2

4*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d^2*f*h - 1920*a^3*b*c^4*d^2*e*i + 800
*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2*f*h - 32*a*b^5*c^2*d^2*e*i - 30*a^2*
b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4*c^3*d^2*e*g - 32*a*b^4*c^
3*d*e^2*f + 512*a^5*c^3*e*f*h*i + 1536*a^4*c^4*d*e*f*i + 16*a*b^6*c*d^2*g*i
 + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^2 + 192*a^4*b^2*c^2*f^2*g*i - 768*a
^4*b^2*c^2*e*g^2*i + 64*a^4*b^2*c^2*f*g^2*h + 960*a^3*b^2*c^3*d^2*g*i - 240
*a^2*b^4*c^2*d^2*g*i + 192*a^4*b^2*c^2*e*g*h^2 - 32*a^3*b^3*c^2*e*f^2*i - 2
24*a^3*b^3*c^2*d*g^2*h + 192*a^4*b^2*c^2*d*f*i^2 + 192*a^3*b^2*c^3*e^2*f*h
- 864*a^3*b^2*c^3*d*f^2*h + 480*a^2*b^3*c^3*d^2*e*i + 336*a^3*b^3*c^2*d*f*h
^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h + 16*a^2*b^4*c^2*e*f
^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2 + 96*a^2*b^3*c^3*d*
e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g + 192*a^2*b^2*c^4*
d*e^2*f - 384*a^5*b^2*c*g^2*i^2 - 192*a^5*b*c^2*f^2*i^2 - 48*a^4*b^3*c*g^2*
h^2 - 16*a^4*b^3*c*f^2*i^2 + 80*a^3*b^3*c^2*f^3*h - 42*a^3*b^4*c*f^2*h^2 -
960*a^4*b*c^3*d^2*i^2 - 192*a^4*b*c^3*e^2*h^2 - 16*a^2*b^5*c*d^2*i^2 - 4*a^
2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^3 - 192*a^2*b^2*c^4*d^3*h + 128*a^3*b
^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 + 60*a*b^5*c^2*d^2*g^2 + 198*a*b^4*c^3
*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960*a^2*b*c^5*d^2*e^2 + 240*a*b^3*c^4*d^
2*e^2 + 256*a^6*c^2*f*h*i^2 + 16*a^4*b^4*g*h^2*i + 768*a^5*c^3*d*f*i^2 + 25
6*a^4*c^4*e^2*f*h - 192*a^6*b*c*h^2*i^2 - 192*a^4*c^4*d*f^2*h + 128*a^4*b^3
*c*g^3*i + 16*b^6*c^2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*a^4*b*c^3*f^3*h + 8
0*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^2*f + 512*a^3*b*c^4
*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a*b^6*c*d^2*h^2 + 20
16*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4*d*f^3 - 18*a*b^5*c
^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 + 240*a^3*b^3*c^2*d^2*i^2 - 48*a^3*b^3*c
^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 - 464*a^3*b^2*c^3*d^2*h^2 - 384*a^3*b^2
*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2 - 240*a^2*b^3*c^3*d^2*g^2 - 16*a^2*b^
3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f^2 + 6*b^7*c*d^2*f*h + 512*a^6*b*c*g*i
^3 - 2*a*b^7*d*f*h^2 - 16*a^5*b^3*h^2*i^2 - 1536*a^5*c^3*e^2*i^2 - 32*a^5*c
^3*f^2*h^2 - 4*a^3*b^5*g^2*h^2 - 864*a^4*c^4*d^2*h^2 - 9*b^6*c^2*d^2*f^2 -
288*a^3*c^5*d^2*f^2 - 16*b^5*c^3*d^2*e^2 - 24*a^3*b^2*c^3*f^4 - 9*a^2*b^4*c
^2*f^4 - 1024*a^6*c^2*e*i^3 - 1024*a^4*c^4*e^3*i - 10*b^6*c^2*d^3*h + 6*a^3
*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3 - 4*b^7*c*d^2*g^2 + 30*
b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4 - 16*a^3*b^4*c*g^4 + 360
*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 16*a^4*c^4*f^4 - 256*a^3*
c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b^6*f^2*h^2 - 256*a^7*c*i
^4 - b^8*d^2*h^2, z, l)*x*(8192*a^6*b*c^6 + 32*a^2*b^9*c^2 - 512*a^3*b^7*c^
3 + 3072*a^4*b^5*c^4 - 8192*a^5*b^3*c^5))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3
*b^4*c + 48*a^4*b^2*c^2))) + (x*(2*b^6*c^3*d^2 - 576*a^3*c^6*d^2 + 64*a^4*c
^5*f^2 - 64*a^5*c^4*h^2 - 36*a*b^4*c^4*d^2 + 128*a^3*b*c^5*e^2 + 2*a^2*b^6*
c*h^2 + 128*a^5*b*c^3*i^2 + 256*a^2*b^2*c^5*d^2 - 32*a^2*b^3*c^4*e^2 + 20*a
^2*b^4*c^3*f^2 - 96*a^3*b^2*c^4*f^2 - 8*a^2*b^5*c^2*g^2 + 32*a^3*b^3*c^3*g^
2 - 4*a^3*b^4*c^2*h^2 - 32*a^4*b^3*c^2*i^2 - 384*a^4*c^5*d*h + 4*a*b^5*c^3*
d*f + 320*a^3*b*c^5*d*f + 256*a^4*b*c^4*e*i + 64*a^4*b*c^4*f*h - 96*a^2*b^3
*c^4*d*f + 8*a^2*b^4*c^3*d*h + 32*a^2*b^4*c^3*e*g + 64*a^3*b^2*c^4*d*h - 12

8*a^3*b^2*c^4*e*g - 12*a^2*b^5*c^2*f*h - 64*a^3*b^3*c^3*e*i + 32*a^3*b^3*c^
3*f*h + 32*a^3*b^4*c^2*g*i - 128*a^4*b^2*c^3*g*i))/(4*(a^2*b^6 - 64*a^5*c^3
 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) - (x*(32*a^2*c^5*e^3 + 32*a^5*c^2*i^3 -
 2*b^3*c^4*d^2*e + b^4*c^3*d^2*g + 96*a^3*c^4*e^2*i + 96*a^4*c^3*e*i^2 - 4*
a^2*b^3*c^2*g^3 + 24*a*b*c^5*d^2*e - 48*a^2*c^5*d*e*f - 48*a^3*c^4*d*f*i -
16*a^3*c^4*e*f*h - 16*a^4*c^3*f*h*i - 12*a*b^2*c^4*d^2*g + 16*a^2*b*c^4*e*f
^2 - 48*a^2*b*c^4*e^2*g - 2*a*b^3*c^3*d^2*i + 24*a^2*b*c^4*d^2*i + 8*a^3*b*
c^3*e*h^2 - a^2*b^4*c*g*h^2 + 16*a^3*b*c^3*f^2*i - 48*a^4*b*c^2*g*i^2 + 2*a
^3*b^3*c*h^2*i + 8*a^4*b*c^2*h^2*i + 24*a^2*b^2*c^3*e*g^2 - 8*a^2*b^2*c^3*f
^2*g + 2*a^2*b^3*c^2*e*h^2 - 4*a^3*b^2*c^2*g*h^2 + 24*a^3*b^2*c^2*g^2*i - 4
*a*b^2*c^4*d*e*f + 2*a*b^3*c^3*d*f*g + 32*a^2*b*c^4*d*e*h + 24*a^2*b*c^4*d*
f*g + 32*a^3*b*c^3*d*h*i - 96*a^3*b*c^3*e*g*i + 8*a^3*b*c^3*f*g*h - 4*a^2*b
^2*c^3*d*f*i - 16*a^2*b^2*c^3*d*g*h - 12*a^2*b^2*c^3*e*f*h + 6*a^2*b^3*c^2*
f*g*h - 12*a^3*b^2*c^2*f*h*i))/(4*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48
*a^4*b^2*c^2)))*root(1572864*a^8*b^2*c^6*z^4 - 983040*a^7*b^4*c^5*z^4 + 327
680*a^6*b^6*c^4*z^4 - 61440*a^5*b^8*c^3*z^4 + 6144*a^4*b^10*c^2*z^4 - 256*a
^3*b^12*c*z^4 - 1048576*a^9*c^7*z^4 + 32768*a^7*b*c^4*g*i*z^2 - 512*a^4*b^7
*c*g*i*z^2 + 192*a^3*b^8*c*f*h*z^2 + 57344*a^6*b*c^5*d*h*z^2 + 32768*a^6*b*
c^5*e*g*z^2 + 96*a^2*b^9*c*d*h*z^2 - 32*a*b^10*c*d*f*z^2 - 24576*a^6*b^3*c^
3*g*i*z^2 + 6144*a^5*b^5*c^2*g*i*z^2 + 49152*a^6*b^2*c^4*e*i*z^2 - 12288*a^
5*b^4*c^3*e*i*z^2 + 6144*a^5*b^4*c^3*f*h*z^2 - 2048*a^4*b^6*c^2*f*h*z^2 + 1
024*a^4*b^6*c^2*e*i*z^2 - 49152*a^5*b^3*c^4*d*h*z^2 - 24576*a^5*b^3*c^4*e*g
*z^2 + 15360*a^4*b^5*c^3*d*h*z^2 + 6144*a^4*b^5*c^3*e*g*z^2 - 2048*a^3*b^7*
c^2*d*h*z^2 - 512*a^3*b^7*c^2*e*g*z^2 + 24576*a^5*b^2*c^5*d*f*z^2 - 3072*a^
3*b^6*c^3*d*f*z^2 + 2048*a^4*b^4*c^4*d*f*z^2 + 576*a^2*b^8*c^2*d*f*z^2 + 51
2*a^5*b^6*c*i^2*z^2 + 12288*a^7*b*c^4*h^2*z^2 + 128*a^3*b^8*c*g^2*z^2 + 122
88*a^6*b*c^5*f^2*z^2 - 16*a^2*b^9*c*f^2*z^2 + 61440*a^5*b*c^6*d^2*z^2 + 432
*a*b^9*c^2*d^2*z^2 - 65536*a^7*c^5*e*i*z^2 - 16384*a^7*c^5*f*h*z^2 - 49152*
a^6*c^6*d*f*z^2 + 24576*a^7*b^2*c^3*i^2*z^2 - 6144*a^6*b^4*c^2*i^2*z^2 - 81
92*a^6*b^3*c^3*h^2*z^2 + 1536*a^5*b^5*c^2*h^2*z^2 - 8192*a^6*b^2*c^4*g^2*z^
2 + 6144*a^5*b^4*c^3*g^2*z^2 - 1536*a^4*b^6*c^2*g^2*z^2 - 8192*a^5*b^3*c^4*
f^2*z^2 + 1536*a^4*b^5*c^3*f^2*z^2 + 24576*a^5*b^2*c^5*e^2*z^2 - 6144*a^4*b
^4*c^4*e^2*z^2 + 512*a^3*b^6*c^3*e^2*z^2 - 61440*a^4*b^3*c^5*d^2*z^2 + 2406
4*a^3*b^5*c^4*d^2*z^2 - 4608*a^2*b^7*c^3*d^2*z^2 - 32768*a^8*c^4*i^2*z^2 -
16*a^3*b^9*h^2*z^2 - 32768*a^6*c^6*e^2*z^2 - 16*b^11*c*d^2*z^2 - 192*a^3*b^
6*c*d*h*i*z - 6144*a^5*b*c^4*d*g*h*z - 4096*a^5*b*c^4*d*f*i*z + 96*a^2*b^7*
c*d*g*h*z + 64*a^2*b^7*c*d*f*i*z - 4096*a^4*b*c^5*d*e*f*z + 64*a*b^7*c^2*d*
e*f*z - 32*a*b^8*c*d*f*g*z - 9216*a^5*b^2*c^3*d*h*i*z + 2304*a^4*b^4*c^2*d*
h*i*z + 4608*a^4*b^3*c^3*d*g*h*z + 3072*a^4*b^3*c^3*d*f*i*z - 1152*a^3*b^5*
c^2*d*g*h*z - 768*a^3*b^5*c^2*d*f*i*z - 9216*a^4*b^2*c^4*d*e*h*z + 2304*a^3
*b^4*c^3*d*e*h*z + 2048*a^4*b^2*c^4*d*f*g*z - 1536*a^3*b^4*c^3*d*f*g*z + 38
4*a^2*b^6*c^2*d*f*g*z - 192*a^2*b^6*c^2*d*e*h*z + 3072*a^3*b^3*c^4*d*e*f*z
- 768*a^2*b^5*c^3*d*e*f*z + 384*a^5*b^4*c*h^2*i*z - 1024*a^6*b*c^3*g*h^2*z
- 192*a^4*b^5*c*g*h^2*z + 32*a^3*b^6*c*f^2*i*z + 1024*a^5*b*c^4*f^2*g*z - 3
2*a^3*b^6*c*e*h^2*z - 16*a^2*b^7*c*f^2*g*z - 9216*a^4*b*c^5*d^2*g*z + 336*a

*b^7*c^2*d^2*g*z - 672*a*b^6*c^3*d^2*e*z + 12288*a^6*c^4*d*h*i*z + 12288*a^
5*c^5*d*e*h*z + 32*a*b^8*c*d^2*i*z - 1536*a^6*b^2*c^2*h^2*i*z + 1536*a^5*b^
2*c^3*f^2*i*z + 768*a^5*b^3*c^2*g*h^2*z - 384*a^4*b^4*c^2*f^2*i*z - 15872*a
^4*b^2*c^4*d^2*i*z + 4992*a^3*b^4*c^3*d^2*i*z - 1536*a^5*b^2*c^3*e*h^2*z -
768*a^4*b^3*c^3*f^2*g*z - 672*a^2*b^6*c^2*d^2*i*z + 384*a^4*b^4*c^2*e*h^2*z
 + 192*a^3*b^5*c^2*f^2*g*z + 7936*a^3*b^3*c^4*d^2*g*z - 2496*a^2*b^5*c^3*d^
2*g*z + 1536*a^4*b^2*c^4*e*f^2*z - 384*a^3*b^4*c^3*e*f^2*z + 32*a^2*b^6*c^2
*e*f^2*z - 15872*a^3*b^2*c^5*d^2*e*z + 4992*a^2*b^4*c^4*d^2*e*z + 2048*a^7*
c^3*h^2*i*z - 32*a^4*b^6*h^2*i*z - 2048*a^6*c^4*f^2*i*z + 16*a^3*b^7*g*h^2*
z + 18432*a^5*c^5*d^2*i*z + 2048*a^6*c^4*e*h^2*z - 2048*a^5*c^5*e*f^2*z + 3
2*b^8*c^2*d^2*e*z + 18432*a^4*c^6*d^2*e*z - 16*b^9*c*d^2*g*z - 256*a^5*b*c^
2*f*g*h*i - 192*a^4*b^3*c*f*g*h*i - 96*a^3*b^4*c*d*g*h*i - 1792*a^4*b*c^3*d
*e*h*i - 768*a^4*b*c^3*d*f*g*i - 256*a^4*b*c^3*e*f*g*h + 32*a^2*b^5*c*d*f*g
*i - 768*a^3*b*c^4*d*e*f*g + 32*a*b^5*c^2*d*e*f*g + 896*a^4*b^2*c^2*d*g*h*i
 + 384*a^4*b^2*c^2*e*f*h*i - 192*a^3*b^3*c^2*e*f*g*h - 192*a^3*b^3*c^2*d*f*
g*i + 192*a^3*b^3*c^2*d*e*h*i + 896*a^3*b^2*c^3*d*e*g*h + 384*a^3*b^2*c^3*d
*e*f*i - 96*a^2*b^4*c^2*d*e*g*h - 64*a^2*b^4*c^2*d*e*f*i - 192*a^2*b^3*c^3*
d*e*f*g + 192*a^5*b^2*c*g*h^2*i + 192*a^5*b^2*c*f*h*i^2 - 384*a^5*b*c^2*e*h
^2*i - 32*a^4*b^3*c*e*h^2*i + 16*a^3*b^4*c*f^2*g*i + 1536*a^5*b*c^2*e*g*i^2
 + 1536*a^4*b*c^3*e^2*g*i - 896*a^5*b*c^2*d*h*i^2 + 96*a^4*b^3*c*d*h*i^2 +
48*a^3*b^4*c*f*g^2*h - 384*a^4*b*c^3*e*f^2*i + 16*a^3*b^4*c*e*g*h^2 - 32*a^
3*b^4*c*d*f*i^2 + 24*a^2*b^5*c*d*g^2*h + 2208*a^3*b*c^4*d^2*f*h - 1920*a^3*
b*c^4*d^2*e*i + 800*a^4*b*c^3*d*f*h^2 - 102*a*b^5*c^2*d^2*f*h - 32*a*b^5*c^
2*d^2*e*i - 30*a^2*b^5*c*d*f*h^2 - 896*a^3*b*c^4*d*e^2*h - 240*a*b^4*c^3*d^
2*e*g - 32*a*b^4*c^3*d*e^2*f + 512*a^5*c^3*e*f*h*i + 1536*a^4*c^4*d*e*f*i +
 16*a*b^6*c*d^2*g*i + 12*a*b^6*c*d*f^2*h - 8*a*b^6*c*d*f*g^2 + 192*a^4*b^2*
c^2*f^2*g*i - 768*a^4*b^2*c^2*e*g^2*i + 64*a^4*b^2*c^2*f*g^2*h + 960*a^3*b^
2*c^3*d^2*g*i - 240*a^2*b^4*c^2*d^2*g*i + 192*a^4*b^2*c^2*e*g*h^2 - 32*a^3*
b^3*c^2*e*f^2*i - 224*a^3*b^3*c^2*d*g^2*h + 192*a^4*b^2*c^2*d*f*i^2 + 192*a
^3*b^2*c^3*e^2*f*h - 864*a^3*b^2*c^3*d*f^2*h + 480*a^2*b^3*c^3*d^2*e*i + 33
6*a^3*b^3*c^2*d*f*h^2 + 192*a^3*b^2*c^3*e*f^2*g + 144*a^2*b^3*c^3*d^2*f*h +
 16*a^2*b^4*c^2*e*f^2*g - 12*a^2*b^4*c^2*d*f^2*h + 192*a^3*b^2*c^3*d*f*g^2
+ 96*a^2*b^3*c^3*d*e^2*h + 48*a^2*b^4*c^2*d*f*g^2 + 960*a^2*b^2*c^4*d^2*e*g
 + 192*a^2*b^2*c^4*d*e^2*f - 384*a^5*b^2*c*g^2*i^2 - 192*a^5*b*c^2*f^2*i^2
- 48*a^4*b^3*c*g^2*h^2 - 16*a^4*b^3*c*f^2*i^2 + 80*a^3*b^3*c^2*f^3*h - 42*a
^3*b^4*c*f^2*h^2 - 960*a^4*b*c^3*d^2*i^2 - 192*a^4*b*c^3*e^2*h^2 - 16*a^2*b
^5*c*d^2*i^2 - 4*a^2*b^5*c*f^2*g^2 - 192*a^4*b^2*c^2*d*h^3 - 192*a^2*b^2*c^
4*d^3*h + 128*a^3*b^3*c^2*e*g^3 - 192*a^3*b*c^4*e^2*f^2 + 60*a*b^5*c^2*d^2*
g^2 + 198*a*b^4*c^3*d^2*f^2 + 144*a^2*b^3*c^3*d*f^3 - 960*a^2*b*c^5*d^2*e^2
 + 240*a*b^3*c^4*d^2*e^2 + 256*a^6*c^2*f*h*i^2 + 16*a^4*b^4*g*h^2*i + 768*a
^5*c^3*d*f*i^2 + 256*a^4*c^4*e^2*f*h - 192*a^6*b*c*h^2*i^2 - 192*a^4*c^4*d*
f^2*h + 128*a^4*b^3*c*g^3*i + 16*b^6*c^2*d^2*e*g + 96*a^5*b*c^2*f*h^3 + 96*
a^4*b*c^3*f^3*h + 80*a^4*b^3*c*f*h^3 + 6*a^2*b^5*c*f^3*h + 768*a^3*c^5*d*e^
2*f + 512*a^3*b*c^4*e^3*g + 132*a*b^4*c^3*d^3*h - 28*a^3*b^4*c*d*h^3 + 12*a
*b^6*c*d^2*h^2 + 2016*a^2*b*c^5*d^3*f - 496*a*b^3*c^4*d^3*f + 224*a^3*b*c^4

```
*d*f^3 - 18*a*b^5*c^2*d*f^3 - 192*a^4*b^2*c^2*f^2*h^2 + 240*a^3*b^3*c^2*d^2
*i^2 - 48*a^3*b^3*c^2*f^2*g^2 - 16*a^3*b^3*c^2*e^2*h^2 - 464*a^3*b^2*c^3*d^
2*h^2 - 384*a^3*b^2*c^3*e^2*g^2 + 42*a^2*b^4*c^2*d^2*h^2 - 240*a^2*b^3*c^3*
d^2*g^2 - 16*a^2*b^3*c^3*e^2*f^2 - 960*a^2*b^2*c^4*d^2*f^2 + 6*b^7*c*d^2*f*
h + 512*a^6*b*c*g*i^3 - 2*a*b^7*d*f*h^2 - 16*a^5*b^3*h^2*i^2 - 1536*a^5*c^3
*e^2*i^2 - 32*a^5*c^3*f^2*h^2 - 4*a^3*b^5*g^2*h^2 - 864*a^4*c^4*d^2*h^2 - 9
*b^6*c^2*d^2*f^2 - 288*a^3*c^5*d^2*f^2 - 16*b^5*c^3*d^2*e^2 - 24*a^3*b^2*c^
3*f^4 - 9*a^2*b^4*c^2*f^4 - 1024*a^6*c^2*e*i^3 - 1024*a^4*c^4*e^3*i - 10*b^
6*c^2*d^3*h + 6*a^3*b^5*f*h^3 - 1728*a^3*c^5*d^3*h - 192*a^5*c^3*d*h^3 - 4*
b^7*c*d^2*g^2 + 30*b^5*c^3*d^3*f + 6*a^2*b^6*d*h^3 - 24*a^5*b^2*c*h^4 - 16*
a^3*b^4*c*g^4 + 360*a*b^2*c^5*d^4 - 16*a^6*c^2*h^4 - 9*a^4*b^4*h^4 - 16*a^4
*c^4*f^4 - 256*a^3*c^5*e^4 - 25*b^4*c^4*d^4 - 1296*a^2*c^6*d^4 - a^2*b^6*f^
2*h^2 - 256*a^7*c*i^4 - b^8*d^2*h^2, z, l), l, 1, 4)
```

3.41 $\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2} \, dx$

Optimal result

Integrand size = 55, antiderivative size = 770

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2} \, dx \\
&= \frac{mx}{c^2} - \frac{bc(ce+aj) - ab^2l - 2ac(cg-al) + (2c^3e - c^2(bg+2aj) - b^3l + bc(bj+3al))\, x^2}{2c^2\,(b^2-4ac)\,(a+bx^2+cx^4)} \\
&\quad - \frac{x(abc(cf+ak) - b^2(c^2d+a^2m) + 2ac(c^2d-ach+a^2m) + (ab^2ck + 2ac^2(cf-ak) - ab^3m - bc(c^2d +}{2ac^2\,(b^2-4ac)\,(a+bx^2+cx^4)} \\
&\quad + \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) - \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-ach-19a^2m)}{\sqrt{b^2-4ac}}\right.}{2\sqrt{2}ac^{5/2}\,(b^2-4ac)\,\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) + \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-ach-19a^2m)}{\sqrt{b^2-4ac}}\right.}{2\sqrt{2}ac^{5/2}\,(b^2-4ac)\,\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad + \frac{(4c^3e - c^2(2bg-4aj) + b^3l - 6abcl)\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2\,(b^2-4ac)^{3/2}} + \frac{l\log(a+bx^2+cx^4)}{4c^2}
\end{aligned}
$$

```
[Out] m*x/c^2+1/2*(-b*c*(a*j+c*e)+a*b^2*l+2*a*c*(-a*l+c*g)-(2*c^3*e-c^2*(2*a*j+b*
g)-b^3*l+b*c*(3*a*l+b*j))*x^2)/c^2/(-4*a*c+b^2)/(c*x^4+b*x^2+a)-1/2*x*(a*b*
c*(a*k+c*f)-b^2*(a^2*m+c^2*d)+2*a*c*(a^2*m-a*c*h+c^2*d)+(a*b^2*c*k+2*a*c^2*
(-a*k+c*f)-a*b^3*m-b*c*(-3*a^2*m+a*c*h+c^2*d))*x^2)/a/c^2/(-4*a*c+b^2)/(c*x
^4+b*x^2+a)+1/2*(4*c^3*e-c^2*(-4*a*j+2*b*g)+b^3*l-6*a*b*c*l)*arctanh((2*c*x
^2+b)/(-4*a*c+b^2)^(1/2))/c^2/(-4*a*c+b^2)^(3/2)+1/4*l*ln(c*x^4+b*x^2+a)/c^
2+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*(a*b^2*c*k-2*a
*c^2*(3*a*k+c*f)-3*a*b^3*m+b*c*(13*a^2*m+a*c*h+c^2*d)+(-a*b^3*c*k+4*a*b*c^2
*(2*a*k+c*f)+3*a*b^4*m+b^2*c*(-19*a^2*m-a*c*h+c^2*d)-4*a*c^2*(-5*a^2*m+a*c*
h+3*c^2*d))/(-4*a*c+b^2)^(1/2))/a/c^(5/2)/(-4*a*c+b^2)*2^(1/2)/(b-(-4*a*c+b
```

```
^2)^(1/2))^(1/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))
*(a*b^2*c*k-2*a*c^2*(3*a*k+c*f)-3*a*b^3*m+b*c*(13*a^2*m+a*c*h+c^2*d)+(a*b^3
*c*k-4*a*b*c^2*(2*a*k+c*f)-3*a*b^4*m-b^2*c*(-19*a^2*m-a*c*h+c^2*d)+4*a*c^2*
(-5*a^2*m+a*c*h+3*c^2*d))/(-4*a*c+b^2)^(1/2))/a/c^(5/2)/(-4*a*c+b^2)*2^(1/2
)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 4.43 (sec) , antiderivative size = 770, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 11, $\frac{\text{number of rules}}{\text{integrand size}} = 0.200$, Rules used = {1687, 1692, 1690, 1180, 211, 1677, 1674, 648, 632, 212, 642}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(-\frac{-b^2c(-19a^2m-ach+c^2d)+4ac^2(-5a^2m+ach+3c^2d)-3ab^4m+ab^3ck-4abc^2(2ak+cf)}{\sqrt{b^2-4ac}}+bc(13a^2m+a\right.}{2\sqrt{2}ac^{5/2}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(\frac{-b^2c(-19a^2m-ach+c^2d)+4ac^2(-5a^2m+ach+3c^2d)-3ab^4m+ab^3ck-4abc^2(2ak+cf)}{\sqrt{b^2-4ac}}+bc(13a^2m+c\right.}{2\sqrt{2}ac^{5/2}(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}}$$

$$-\frac{x(-(b^2(a^2m+c^2d))+x^2(-bc(-3a^2m+ach+c^2d)-ab^3m+ab^2ck+2ac^2(cf-ak))+2ac(a^2m-ac}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$+\frac{\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)(-c^2(2bg-4aj)-6abcl+b^3l+4c^3e)}{2c^2(b^2-4ac)^{3/2}}$$

$$-\frac{x^2(-c^2(2aj+bg)+bc(3al+bj)+b^3(-l)+2c^3e)-ab^2l+bc(aj+ce)-2ac(cg-al)}{2c^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$+\frac{l\log(a+bx^2+cx^4)}{4c^2}+\frac{mx}{c^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4)^2,x]
```

```
[Out] (m*x)/c^2 - (b*c*(c*e + a*j) - a*b^2*l - 2*a*c*(c*g - a*l) + (2*c^3*e - c^2
*(b*g + 2*a*j) - b^3*l + b*c*(b*j + 3*a*l))*x^2)/(2*c^2*(b^2 - 4*a*c)*(a +
b*x^2 + c*x^4)) - (x*(a*b*c*(c*f + a*k) - b^2*(c^2*d + a^2*m) + 2*a*c*(c^2*
d - a*c*h + a^2*m) + (a*b^2*c*k + 2*a*c^2*(c*f - a*k) - a*b^3*m - b*c*(c^2*
d + a*c*h - 3*a^2*m))*x^2))/(2*a*c^2*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (
(a*b^2*c*k - 2*a*c^2*(c*f + 3*a*k) - 3*a*b^3*m + b*c*(c^2*d + a*c*h + 13*a^
2*m) - (a*b^3*c*k - 4*a*b*c^2*(c*f + 2*a*k) - 3*a*b^4*m - b^2*c*(c^2*d - a*
c*h - 19*a^2*m) + 4*a*c^2*(3*c^2*d + a*c*h - 5*a^2*m))/Sqrt[b^2 - 4*a*c])*A
rcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*c^(5/2
)*(b^2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((a*b^2*c*k - 2*a*c^2*(c*f +
 3*a*k) - 3*a*b^3*m + b*c*(c^2*d + a*c*h + 13*a^2*m) + (a*b^3*c*k - 4*a*b*c
```

```
^2*(c*f + 2*a*k) - 3*a*b^4*m - b^2*c*(c^2*d - a*c*h - 19*a^2*m) + 4*a*c^2*(
3*c^2*d + a*c*h - 5*a^2*m))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/S
qrt[b + Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*c^(5/2)*(b^2 - 4*a*c)*Sqrt[b + Sq
rt[b^2 - 4*a*c]]) + ((4*c^3*e - c^2*(2*b*g - 4*a*j) + b^3*l - 6*a*b*c*l)*Ar
cTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(2*c^2*(b^2 - 4*a*c)^(3/2)) + (l*Lo
g[a + b*x^2 + c*x^4])/(4*c^2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1690

```
Int[(Pq_)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :> Int[ExpandInte
grand[Pq/(a + b*x^2 + c*x^4), x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{x(e+gx^2+jx^4+lx^6)}{(a+bx^2+cx^4)^2}\,dx + \int \frac{d+fx^2+hx^4+kx^6+mx^8}{(a+bx^2+cx^4)^2}\,dx$$

$$= -\frac{x(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+(ab^2ck+2ac^2(cf-ak)-ab^3m-bc}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$+\frac{1}{2}\text{Subst}\left(\int\frac{e+gx+jx^2+lx^3}{(a+bx+cx^2)^2}\,dx,x,x^2\right)$$

$$-\frac{\int\frac{-\frac{abc(cf+ak)+b^2\left(c^2d-a^2m\right)-2ac\left(3c^2d+ach-a^2m\right)}{c^2}-\frac{\left(ab^2ck-2ac^2(cf+3ak)-ab^3m+bc\left(c^2d+ach+5a^2m\right)\right)x^2}{c^2}+2a\left(4a-\frac{b^2}{c}\right)mx^4}{a+bx^2+cx^4}\,dx}{2a(b^2-4ac)}$$

$$= -\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$-\frac{x(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+(ab^2ck+2ac^2(cf-ak)-ab^3m-bc}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$-\frac{\text{Subst}\left(\int\frac{2ce-bg+2aj-\frac{abl}{c}+\left(4a-\frac{b^2}{c}\right)lx}{a+bx+cx^2}\,dx,x,x^2\right)}{2(b^2-4ac)}$$

$$-\frac{\int\left(-\frac{2a(b^2-4ac)m}{c^2}-\frac{abc(cf+ak)-2ac\left(3c^2d+ach-5a^2m\right)+b^2\left(c^2d-3a^2m\right)+\left(ab^2ck-2ac^2(cf+3ak)-3ab^3m+bc\left(c^2d+ach+13a}{c^2(a+bx^2+cx^4)}\right.}{2a(b^2-4ac)}$$

$$= \frac{mx}{c^2}$$

$$-\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$-\frac{x(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+(ab^2ck+2ac^2(cf-ak)-ab^3m-bc}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$

$$+\frac{\int\frac{abc(cf+ak)-2ac\left(3c^2d+ach-5a^2m\right)+b^2\left(c^2d-3a^2m\right)+\left(ab^2ck-2ac^2(cf+3ak)-3ab^3m+bc\left(c^2d+ach+13a^2m\right)\right)x^2}{a+bx^2+cx^4}\,dx}{2ac^2(b^2-4ac)}$$

$$+\frac{l\text{Subst}\left(\int\frac{b+2cx}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^2}$$

$$-\frac{(4c^3e-c^2(2bg-4aj)+b^3l-6abcl)\,\text{Subst}\left(\int\frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^2(b^2-4ac)}$$

$$= \frac{mx}{c^2} - \frac{bc(ce+aj) - ab^2l - 2ac(cg-al) + (2c^3e - c^2(bg+2aj) - b^3l + bc(bj+3al))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)}$$
$$- \frac{x(abc(cf+ak) - b^2(c^2d+a^2m) + 2ac(c^2d-ach+a^2m) + (ab^2ck + 2ac^2(cf-ak) - ab^3m - \cdots}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$
$$+ \frac{l\log(a+bx^2+cx^4)}{4c^2}$$
$$+ \frac{(4c^3e - c^2(2bg-4aj) + b^3l - 6abcl)\operatorname{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{2c^2(b^2-4ac)}$$
$$+ \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) - \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-\cdots}{\sqrt{b^2-\cdots}}\right.}{4ac^2(b^2-4ac)}$$
$$+ \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) + \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-\cdots}{\sqrt{b^2-\cdots}}\right.}{4ac^2(b^2-4ac)}$$

$$= \frac{mx}{c^2} - \frac{bc(ce+aj) - ab^2l - 2ac(cg-al) + (2c^3e - c^2(bg+2aj) - b^3l + bc(bj+3al))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)}$$
$$- \frac{x(abc(cf+ak) - b^2(c^2d+a^2m) + 2ac(c^2d-ach+a^2m) + (ab^2ck + 2ac^2(cf-ak) - ab^3m - \cdots}{2ac^2(b^2-4ac)(a+bx^2+cx^4)}$$
$$+ \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) - \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-\cdots}{\sqrt{b^2-\cdots}}\right.}{2\sqrt{2}ac^{5/2}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\left(ab^2ck - 2ac^2(cf+3ak) - 3ab^3m + bc(c^2d+ach+13a^2m) + \frac{ab^3ck-4abc^2(cf+2ak)-3ab^4m-b^2c(c^2d-\cdots}{\sqrt{b^2-\cdots}}\right.}{2\sqrt{2}ac^{5/2}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+ \frac{(4c^3e - c^2(2bg-4aj) + b^3l - 6abcl)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2(b^2-4ac)^{3/2}} + \frac{l\log(a+bx^2+cx^4)}{4c^2}$$

Mathematica [A] (verified)

Time = 3.22 (sec) , antiderivative size = 935, normalized size of antiderivative = 1.21

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{4\sqrt{c}mx + \frac{2\sqrt{c}(2a^3c(l+mx)-bc^2dx(b+cx^2)+a(b^2cx^2(j+kx)-b^3x^2(l+mx)+2c^3x(d+x(e+fx))+bc^2(e+x(f-x(g+hx))))-a^2(b^2(l+mx)+2\cdots}{a(-b^2+4ac)(a+bx^2+cx^4)}}{\cdots}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)
/(a + b*x^2 + c*x^4)^2,x]
```

[Out] (4*Sqrt[c]*m*x + (2*Sqrt[c]*(2*a^3*c*(l + m*x) - b*c^2*d*x*(b + c*x^2) + a*
(b^2*c*x^2*(j + k*x) - b^3*x^2*(l + m*x) + 2*c^3*x*(d + x*(e + f*x)) + b*c^
2*(e + x*(f - x*(g + h*x)))) - a^2*(b^2*(l + m*x) + 2*c^2*(g + x*(h + x*(j
+ k*x))) - b*c*(j + x*(k + 3*x*(l + m*x))))))/(a*(-b^2 + 4*a*c)*(a + b*x^2
+ c*x^4)) - (Sqrt[2]*(-3*a*b^4*m + 2*a*c^2*(6*c^2*d + c*Sqrt[b^2 - 4*a*c]*f
 + 2*a*c*h + 3*a*Sqrt[b^2 - 4*a*c]*k - 10*a^2*m) + a*b^3*(c*k + 3*Sqrt[b^2
- 4*a*c]*m) - b*c*(c^2*(Sqrt[b^2 - 4*a*c]*d + 4*a*f) + a*c*(Sqrt[b^2 - 4*a*
c]*h + 8*a*k) + 13*a^2*Sqrt[b^2 - 4*a*c]*m) + b^2*c*(-(c^2*d) + a*c*h + a*(
-(Sqrt[b^2 - 4*a*c]*k) + 19*a*m)))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt
[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) - (Sqr
t[2]*(3*a*b^4*m + 2*a*c^2*(-6*c^2*d + c*Sqrt[b^2 - 4*a*c]*f - 2*a*c*h + 3*a
*Sqrt[b^2 - 4*a*c]*k + 10*a^2*m) + a*b^3*(-(c*k) + 3*Sqrt[b^2 - 4*a*c]*m) -
 b*c*(c^2*(Sqrt[b^2 - 4*a*c]*d - 4*a*f) + a*c*(Sqrt[b^2 - 4*a*c]*h - 8*a*k)
 + 13*a^2*Sqrt[b^2 - 4*a*c]*m) + b^2*c*(c^2*d - a*c*h - a*(Sqrt[b^2 - 4*a*c
]*k + 19*a*m)))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a
*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(-4*c^3*e + 2*
c^2*(b*g - 2*a*j) + b^2*(-b + Sqrt[b^2 - 4*a*c])*l + a*c*(6*b*l - 4*Sqrt[b^
2 - 4*a*c]*l))*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) +
 (Sqrt[c]*(4*c^3*e + c^2*(-2*b*g + 4*a*j) + b^2*(b + Sqrt[b^2 - 4*a*c])*l -
 2*a*c*(3*b + 2*Sqrt[b^2 - 4*a*c])*l)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])
/(b^2 - 4*a*c)^(3/2))/(4*c^(5/2))

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 1.01 (sec) , antiderivative size = 506, normalized size of antiderivative = 0.66

method	result
risch	$\frac{mx}{c^2}+\frac{\frac{\left(3a^2bcm-2a^2c^2k-a\,b^3m+a\,b^2ck-ab\,c^2h+2a\,c^3f-b\,c^3d\right)x^3}{2a\left(4ac-b^2\right)}+\frac{\left(3abcl-2a\,c^2j-b^3l+b^2cj-b\,c^2g+2c^3e\right)x^2}{8ac-2b^2}+\frac{\left(2a^3cm-a^2b^2m+a^2bck-2a^2c^2 \right.}{2a\left(4ac- \right.}}{c^2(c\,x^4+b\,x^2+a)}$
default	Expression too large to display

[In] int((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,m
ethod=_RETURNVERBOSE)

[Out] m*x/c^2+(1/2/a*(3*a^2*b*c*m-2*a^2*c^2*k-a*b^3*m+a*b^2*c*k-a*b*c^2*h+2*a*c^3
*f-b*c^3*d)/(4*a*c-b^2)*x^3+1/2*(3*a*b*c*l-2*a*c^2*j-b^3*l+b^2*c*j-b*c^2*g+
2*c^3*e)/(4*a*c-b^2)*x^2+1/2*(2*a^3*c*m-a^2*b^2*m+a^2*b*c*k-2*a^2*c^2*h+a*b
*c^2*f+2*a*c^3*d-b^2*c^2*d)/a/(4*a*c-b^2)*x+1/2*(2*a^2*c*l-a*b^2*l+a*b*c*j-
2*a*c^2*g+b*c^2*e)/(4*a*c-b^2))/c^2/(c*x^4+b*x^2+a)+1/4/c^2*sum((2*l*c*_R^3
-(13*a^2*b*c*m-6*a^2*c^2*k-3*a*b^3*m+a*b^2*c*k+a*b*c^2*h-2*a*c^3*f+b*c^3*d)
/a/(4*a*c-b^2)*_R^2-2*c*(a*b*l-2*a*c*j+b*c*g-2*c^2*e)/(4*a*c-b^2)*_R-(10*a^

```
3*c*m-3*a^2*b^2*m+a^2*b*c*k-2*a^2*c^2*h+a*b*c^2*f-6*a*c^3*d+b^2*c^2*d)/a/(4
*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((m*x**8+l*x**7+k*x**6+j*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+
b*x**2+a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx$$
$$=\int \frac{mx^8+lx^7+kx^6+jx^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^2,x, algorithm="maxima")
```

```
[Out] -1/2*(a*b*c^2*e - 2*a^2*c^2*g + a^2*b*c*j - (b*c^3*d - 2*a*c^3*f + a*b*c^2*
h - (a*b^2*c - 2*a^2*c^2)*k + (a*b^3 - 3*a^2*b*c)*m)*x^3 + (2*a*c^3*e - a*b
*c^2*g + (a*b^2*c - 2*a^2*c^2)*j - (a*b^3 - 3*a^2*b*c)*l)*x^2 - (a^2*b^2 -
2*a^3*c)*l + (a*b*c^2*f - 2*a^2*c^2*h + a^2*b*c*k - (b^2*c^2 - 2*a*c^3)*d -
 (a^2*b^2 - 2*a^3*c)*m)*x)/(a^2*b^2*c^2 - 4*a^3*c^3 + (a*b^2*c^3 - 4*a^2*c^
4)*x^4 + (a*b^3*c^2 - 4*a^2*b*c^3)*x^2) + m*x/c^2 - 1/2*integrate(-(a*b*c^2
```

```
*f - 2*a^2*c^2*h + a^2*b*c*k + 2*(a*b^2*c - 4*a^2*c^2)*l*x^3 + (b*c^3*d - 2
*a*c^3*f + a*b*c^2*h + (a*b^2*c - 6*a^2*c^2)*k - (3*a*b^3 - 13*a^2*b*c)*m)*
x^2 + (b^2*c^2 - 6*a*c^3)*d - (3*a^2*b^2 - 10*a^3*c)*m - 2*(2*a*c^3*e - a*b
*c^2*g + 2*a^2*c^2*j - a^2*b*c*l)*x)/(c*x^4 + b*x^2 + a), x)/(a*b^2*c^2 - 4
*a^2*c^3)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 20159 vs. 2(718) = 1436.

Time = 3.93 (sec) , antiderivative size = 20159, normalized size of antiderivative = 26.18

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^2,x, algorithm="giac")
```

```
[Out] m*x/c^2 + 1/4*l*log(abs(c*x^4 + b*x^2 + a))/c^2 + 1/16*((a^2*b^4*c^5 - 8*a^
3*b^2*c^6 + 16*a^4*c^7)^2*(2*b^3*c^5 - 8*a*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*b*c^5 - 2*(b^2 - 4*a*c)*b*c^5)*d - 2*(a^2*b^4*c^5 - 8*a^3*
b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^2*c^5 - 8*a^2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a*c^5 - 2*(b^2 - 4*a*c)*a*c^5)*f + (a^2*b^4*c^5 - 8*a^3*
b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^3*c^4 - 8*a^2*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a*b*c^4 - 2*(b^2 - 4*a*c)*a*b*c^4)*h + (a^2*b^4*c^
5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^4*c^3 - 20*a^2*b^2*c^4 + 48*a^3*c^
5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c + 10*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^2 + 2*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^2 - 24*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^3 - 12*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^3 - sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 + 6*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^4 - 2*(b^2 - 4*a*c)*a*b^2*
c^3 + 12*(b^2 - 4*a*c)*a^2*c^4)*k - (a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c
^7)^2*(6*a*b^5*c^2 - 50*a^2*b^3*c^3 + 104*a^3*b*c^4 - 3*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5 + 25*sqrt(2)*sqrt(b^2 - 4*a*c)
```

*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c - 52*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^2 - 26*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^2 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a*b^3*c^2 + 13*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^2*b*c^3 - 6*(b^2 - 4*a*c)*a*b^3*c^2 + 26*(b^2 - 4*a*
c)*a^2*b*c^3)*m + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^8*c^8 -
18*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^9 - 2*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b^7*c^9 - 2*a^2*b^8*c^9 + 120*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 + 28*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^3*b^5*c^10 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c^10 +
 36*a^3*b^6*c^10 - 352*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^11
 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^11 - 14*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^11 - 240*a^4*b^4*c^11 + 384*sqrt(2)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*c^12 + 192*sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^5*b*c^12 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^
2*c^12 + 704*a^5*b^2*c^12 - 96*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*
c^13 - 768*a^6*c^13 + 2*(b^2 - 4*a*c)*a^2*b^6*c^9 - 28*(b^2 - 4*a*c)*a^3*b^
4*c^10 + 128*(b^2 - 4*a*c)*a^4*b^2*c^11 - 192*(b^2 - 4*a*c)*a^5*c^12)*d*abs
(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^3*b^7*c^8 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b
^5*c^9 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^9 - 2*a^3*b^7*
c^9 + 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^10 + 16*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 + sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^3*b^5*c^10 + 24*a^4*b^5*c^10 - 64*sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^6*b*c^11 - 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^
2*c^11 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^11 - 96*a^5*b^
3*c^11 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*c^12 + 128*a^6*b*
c^12 + 2*(b^2 - 4*a*c)*a^3*b^5*c^9 - 16*(b^2 - 4*a*c)*a^4*b^3*c^10 + 32*(b^
2 - 4*a*c)*a^5*b*c^11)*f*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 4*
(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^8 - 12*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^9 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^4*b^5*c^9 - 2*a^4*b^6*c^9 + 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^6*b^2*c^10 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^10 +
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 + 24*a^5*b^4*c^10 - 64
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*c^11 - 32*sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^6*b*c^11 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^5*b^2*c^11 - 96*a^6*b^2*c^11 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^6*c^12 + 128*a^7*c^12 + 2*(b^2 - 4*a*c)*a^4*b^4*c^9 - 16*(b^2 - 4*a*c)*a
^5*b^2*c^10 + 32*(b^2 - 4*a*c)*a^6*c^11)*h*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6
+ 16*a^4*c^7) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^7*c^7 - 12
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^8 - 2*sqrt(2)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^8 - 2*a^4*b^7*c^8 + 48*sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^6*b^3*c^9 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^5*b^4*c^9 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^9 + 24*a^

5*b^5*c^9 - 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b*c^10 - 32*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^10 - 8*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^5*b^3*c^10 - 96*a^6*b^3*c^10 + 16*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^6*b*c^11 + 128*a^7*b*c^11 + 2*(b^2 - 4*a*c)*a^4*b^5*c^
8 - 16*(b^2 - 4*a*c)*a^5*b^3*c^9 + 32*(b^2 - 4*a*c)*a^6*b*c^10)*k*abs(a^2*b
^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 2*(3*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^4*b^8*c^6 - 46*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^6*c
^7 - 6*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^7*c^7 - 6*a^4*b^8*c^7
+ 264*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^4*c^8 + 68*sqrt(2)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^8 + 3*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^4*b^6*c^8 + 92*a^5*b^6*c^8 - 672*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^7*b^2*c^9 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^3*
c^9 - 34*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^9 - 528*a^6*b^4*
c^9 + 640*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*c^10 + 320*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b*c^10 + 128*sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^6*b^2*c^10 + 1344*a^7*b^2*c^10 - 160*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^7*c^11 - 1280*a^8*c^11 + 6*(b^2 - 4*a*c)*a^4*b^6*c^7 - 68
*(b^2 - 4*a*c)*a^5*b^4*c^8 + 256*(b^2 - 4*a*c)*a^6*b^2*c^9 - 320*(b^2 - 4*a
*c)*a^7*c^10)*m*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + (2*a^4*b^11
*c^15 - 56*a^5*b^9*c^16 + 576*a^6*b^7*c^17 - 2816*a^7*b^5*c^18 + 6656*a^8*b
^3*c^19 - 6144*a^9*b*c^20 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^4*b^11*c^13 + 28*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^5*b^9*c^14 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^4*b^10*c^14 - 288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^6*b^7*c^15 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^5*b^8*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^4*b^9*c^15 + 1408*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^7*b^5*c^16 + 384*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^6*b^6*c^16 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^16 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^17 - 1280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^17 - 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^17 + 3072*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b*c^18 + 1536*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^18 + 640*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^3*c^18 - 768*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b*c^19 - 2*(b^2 - 4*a*c)*a^4*b^9*c
^15 + 48*(b^2 - 4*a*c)*a^5*b^7*c^16 - 384*(b^2 - 4*a*c)*a^6*b^5*c^17 + 1280
*(b^2 - 4*a*c)*a^7*b^3*c^18 - 1536*(b^2 - 4*a*c)*a^8*b*c^19)*d + 4*(2*a^5*b
^10*c^15 - 32*a^6*b^8*c^16 + 192*a^7*b^6*c^17 - 512*a^8*b^4*c^18 + 512*a^9*
b^2*c^19 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^
10*c^13 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*
b^8*c^14 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*
b^9*c^14 - 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7
*b^6*c^15 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^

6*b^7*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*
b^8*c^15 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
8*b^4*c^16 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^7*b^5*c^16 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^6*b^6*c^16 - 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^9*b^2*c^17 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^8*b^3*c^17 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^7*b^4*c^17 + 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^8*b^2*c^18 - 2*(b^2 - 4*a*c)*a^5*b^8*c^15 + 24*(b^2 - 4*a*c)*a^6*b^
6*c^16 - 96*(b^2 - 4*a*c)*a^7*b^4*c^17 + 128*(b^2 - 4*a*c)*a^8*b^2*c^18)*f
- (2*a^5*b^11*c^14 - 24*a^6*b^9*c^15 + 64*a^7*b^7*c^16 + 256*a^8*b^5*c^17 -
1536*a^9*b^3*c^18 + 2048*a^10*b*c^19 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^12 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^13 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^13 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^14 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^14 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^9*c^14 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^15 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^7*c^15 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^16 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^16 - 1024*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^17 - 512*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^17 - 128*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^17 + 256*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b*c^18 - 2*(b^2 - 4*a*c)*a^5*b^
9*c^14 + 16*(b^2 - 4*a*c)*a^6*b^7*c^15 - 256*(b^2 - 4*a*c)*a^8*b^3*c^17 + 5
12*(b^2 - 4*a*c)*a^9*b*c^18)*h - (2*a^5*b^12*c^13 - 48*a^6*b^10*c^14 + 448*
a^7*b^8*c^15 - 2048*a^8*b^6*c^16 + 4608*a^9*b^4*c^17 - 4096*a^10*b^2*c^18 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^11 +
24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^12
+ 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^12
- 224*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^
13 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c
^13 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^
13 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6
*c^14 + 288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b
^7*c^14 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*
b^8*c^14 - 2304*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^9*b^4*c^15 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^8*b^5*c^15 - 144*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^7*b^6*c^15 + 2048*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^10*b^2*c^16 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^9*b^3*c^16 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^8*b^4*c^16 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^

2 - 4*a*c)*c)*a^9*b^2*c^17 - 2*(b^2 - 4*a*c)*a^5*b^10*c^13 + 40*(b^2 - 4*a*c)*a^6*b^8*c^14 - 288*(b^2 - 4*a*c)*a^7*b^6*c^15 + 896*(b^2 - 4*a*c)*a^8*b^4*c^16 - 1024*(b^2 - 4*a*c)*a^9*b^2*c^17)*k + (6*a^5*b^13*c^12 - 134*a^6*b^11*c^13 + 1224*a^7*b^9*c^14 - 5824*a^8*b^7*c^15 + 15104*a^9*b^5*c^16 - 19968*a^10*b^3*c^17 + 10240*a^11*b*c^18 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^13*c^10 + 67*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^11 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^11 - 612*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^12 - 110*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^12 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^12 + 2912*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^13 + 784*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^13 + 55*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^13 - 7552*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^14 - 2688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^14 - 392*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^14 + 9984*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^3*c^15 + 4352*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^15 + 1344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^15 - 5120*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b*c^16 - 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^16 - 2176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^16 + 1280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^17 - 6*(b^2 - 4*a*c)*a^5*b^11*c^12 + 110*(b^2 - 4*a*c)*a^6*b^9*c^13 - 784*(b^2 - 4*a*c)*a^7*b^7*c^14 + 2688*(b^2 - 4*a*c)*a^8*b^5*c^15 - 4352*(b^2 - 4*a*c)*a^9*b^3*c^16 + 2560*(b^2 - 4*a*c)*a^10*b*c^17)*m)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7 + sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7)^2 - 4*(a^3*b^4*c^5 - 8*a^4*b^2*c^6 + 16*a^5*c^7)*(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8)))/(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8)))/((a^4*b^8*c^8 - 16*a^5*b^6*c^9 - 2*a^4*b^7*c^9 + 96*a^6*b^4*c^10 + 24*a^5*b^5*c^10 + a^4*b^6*c^10 - 256*a^7*b^2*c^11 - 96*a^6*b^3*c^11 - 12*a^5*b^4*c^11 + 256*a^8*c^12 + 128*a^7*b*c^12 + 48*a^6*b^2*c^12 - 64*a^7*c^13)*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)*abs(c)) - 1/16*((a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)^2*(2*b^3*c^5 - 8*a*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^5 - 2*(b^2 - 4*a*c)*b*c^5)*d - 2*(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^2*c^5 - 8*a^2*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^5 - 2*(b^2 - 4*a*c)*a*c^5)*f + (a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^3*c^4 - 8*a^2*b*c^5 - sqr

t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^2 + 4*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^3 + 2*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3 - sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^4 - 2*(b^2 - 4*a*c)*a*b*c
^4)*h + (a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)^2*(2*a*b^4*c^3 - 20*a^2*
b^2*c^4 + 48*a^3*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^4*c + 10*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^2*b^2*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a*b^3*c^2 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^3*c^3 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b
*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^3
+ 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^4 - 2*(
b^2 - 4*a*c)*a*b^2*c^3 + 12*(b^2 - 4*a*c)*a^2*c^4)*k - (a^2*b^4*c^5 - 8*a^3
*b^2*c^6 + 16*a^4*c^7)^2*(6*a*b^5*c^2 - 50*a^2*b^3*c^3 + 104*a^3*b*c^4 - 3*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5 + 25*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c + 6*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c - 52*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^2 - 26*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^2 - 3*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^2 + 13*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^3 - 6*(b^2 - 4*a*c)*a*b^3*c
^2 + 26*(b^2 - 4*a*c)*a^2*b*c^3)*m - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^8*c^8 - 18*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^9 -
 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7*c^9 + 2*a^2*b^8*c^9 + 12
0*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 + 28*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^10 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^6*c^10 - 36*a^3*b^6*c^10 - 352*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a^5*b^2*c^11 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c
^11 - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^11 + 240*a^4*b^4
*c^11 + 384*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*c^12 + 192*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^12 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^4*b^2*c^12 - 704*a^5*b^2*c^12 - 96*sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^5*c^13 + 768*a^6*c^13 - 2*(b^2 - 4*a*c)*a^2*b^6*c^9 + 28*(
b^2 - 4*a*c)*a^3*b^4*c^10 - 128*(b^2 - 4*a*c)*a^4*b^2*c^11 + 192*(b^2 - 4*a
*c)*a^5*c^12)*d*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 2*(sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^8 - 12*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^4*b^5*c^9 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b
^6*c^9 + 2*a^3*b^7*c^9 + 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3
*c^10 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 + sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^10 - 24*a^4*b^5*c^10 - 64*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^11 - 32*sqrt(2)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^5*b^2*c^11 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b
^3*c^11 + 96*a^5*b^3*c^11 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*
b*c^12 - 128*a^6*b*c^12 - 2*(b^2 - 4*a*c)*a^3*b^5*c^9 + 16*(b^2 - 4*a*c)*a^
4*b^3*c^10 - 32*(b^2 - 4*a*c)*a^5*b*c^11)*f*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6

+ 16*a^4*c^7) + 4*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^8 - 1
2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^9 - 2*sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^9 + 2*a^4*b^6*c^9 + 48*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^10 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^5*b^3*c^10 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^10 - 2
4*a^5*b^4*c^10 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*c^11 - 32*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^11 - 8*sqrt(2)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^5*b^2*c^11 + 96*a^6*b^2*c^11 + 16*sqrt(2)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^6*c^12 - 128*a^7*c^12 - 2*(b^2 - 4*a*c)*a^4*b^4*c^9 +
16*(b^2 - 4*a*c)*a^5*b^2*c^10 - 32*(b^2 - 4*a*c)*a^6*c^11)*h*abs(a^2*b^4*c
^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^4*b^7*c^7 - 12*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^8 - 2
*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^8 + 2*a^4*b^7*c^8 + 48*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^9 + 16*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^9 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^4*b^5*c^9 - 24*a^5*b^5*c^9 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^7*b*c^10 - 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^10 - 8*sqr
t(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^10 + 96*a^6*b^3*c^10 + 16*sq
rt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^11 - 128*a^7*b*c^11 - 2*(b^2
- 4*a*c)*a^4*b^5*c^8 + 16*(b^2 - 4*a*c)*a^5*b^3*c^9 - 32*(b^2 - 4*a*c)*a^6*
b*c^10)*k*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + 2*(3*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^8*c^6 - 46*sqrt(2)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^5*b^6*c^7 - 6*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^7*c
^7 + 6*a^4*b^8*c^7 + 264*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^4*c^
8 + 68*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^8 + 3*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^8 - 92*a^5*b^6*c^8 - 672*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^9 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^6*b^3*c^9 - 34*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^4
*c^9 + 528*a^6*b^4*c^9 + 640*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*c^
10 + 320*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b*c^10 + 128*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^10 - 1344*a^7*b^2*c^10 - 160*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*c^11 + 1280*a^8*c^11 - 6*(b^2 - 4*a*
c)*a^4*b^6*c^7 + 68*(b^2 - 4*a*c)*a^5*b^4*c^8 - 256*(b^2 - 4*a*c)*a^6*b^2*c
^9 + 320*(b^2 - 4*a*c)*a^7*c^10)*m*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4
*c^7) + (2*a^4*b^11*c^15 - 56*a^5*b^9*c^16 + 576*a^6*b^7*c^17 - 2816*a^7*b^
5*c^18 + 6656*a^8*b^3*c^19 - 6144*a^9*b*c^20 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^11*c^13 + 28*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^9*c^14 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^10*c^14 - 288*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^7*c^15 - 48*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^8*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^9*c^15 + 1408*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^5*c^16 + 384*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^6*c^16 + 24*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^16 - 3328*sqrt(2)*sqrt(b^

```
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^17 - 1280*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^17 - 192*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^17 + 3072*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b*c^18 + 1536*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^18 + 640*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^3*c^18 - 768*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b*c^19 - 2*(b^2
 - 4*a*c)*a^4*b^9*c^15 + 48*(b^2 - 4*a*c)*a^5*b^7*c^16 - 384*(b^2 - 4*a*c)*
a^6*b^5*c^17 + 1280*(b^2 - 4*a*c)*a^7*b^3*c^18 - 1536*(b^2 - 4*a*c)*a^8*b*c
^19)*d + 4*(2*a^5*b^10*c^15 - 32*a^6*b^8*c^16 + 192*a^7*b^6*c^17 - 512*a^8*
b^4*c^18 + 512*a^9*b^2*c^19 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^5*b^10*c^13 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^6*b^8*c^14 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^5*b^9*c^14 - 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^7*b^6*c^15 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^6*b^7*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^5*b^8*c^15 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^8*b^4*c^16 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^7*b^5*c^16 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^6*b^6*c^16 - 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^17 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^17 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^17 + 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^18 - 2*(b^2 - 4*a*c)*a^5*b^8*c^15 + 24*(
b^2 - 4*a*c)*a^6*b^6*c^16 - 96*(b^2 - 4*a*c)*a^7*b^4*c^17 + 128*(b^2 - 4*a*
c)*a^8*b^2*c^18)*f - (2*a^5*b^11*c^14 - 24*a^6*b^9*c^15 + 64*a^7*b^7*c^16 +
 256*a^8*b^5*c^17 - 1536*a^9*b^3*c^18 + 2048*a^10*b*c^19 - sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^12 + 12*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^13 + 2*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^13 - 32*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^14 - 16*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^14 - sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^9*c^14 - 128*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^15 + 8*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^7*c^15 + 768*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^16 + 256*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^16 - 1024*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b*c^17 - 512*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^17 - 128*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^17 + 256
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b*c^18 - 2*(
b^2 - 4*a*c)*a^5*b^9*c^14 + 16*(b^2 - 4*a*c)*a^6*b^7*c^15 - 256*(b^2 - 4*a*
c)*a^8*b^3*c^17 + 512*(b^2 - 4*a*c)*a^9*b*c^18)*h - (2*a^5*b^12*c^13 - 48*a
^6*b^10*c^14 + 448*a^7*b^8*c^15 - 2048*a^8*b^6*c^16 + 4608*a^9*b^4*c^17 - 4
096*a^10*b^2*c^18 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
```

c)*a^5*b^12*c^11 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^6*b^10*c^12 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^5*b^11*c^12 - 224*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^7*b^8*c^13 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^6*b^9*c^13 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^5*b^10*c^13 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^8*b^6*c^14 + 288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^7*b^7*c^14 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^6*b^8*c^14 - 2304*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^9*b^4*c^15 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^8*b^5*c^15 - 144*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^15 + 2048*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^16 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^16 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^16 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^17 - 2*(b^2 - 4*a*c)*a^5*b^10*c^
13 + 40*(b^2 - 4*a*c)*a^6*b^8*c^14 - 288*(b^2 - 4*a*c)*a^7*b^6*c^15 + 896*(
b^2 - 4*a*c)*a^8*b^4*c^16 - 1024*(b^2 - 4*a*c)*a^9*b^2*c^17)*k + (6*a^5*b^1
3*c^12 - 134*a^6*b^11*c^13 + 1224*a^7*b^9*c^14 - 5824*a^8*b^7*c^15 + 15104*
a^9*b^5*c^16 - 19968*a^10*b^3*c^17 + 10240*a^11*b*c^18 - 3*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^13*c^10 + 67*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^11 + 6*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^11 - 612*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^12 - 110*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^12 - 3*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^12 + 2912*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^13 + 784*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^13 + 55*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^13 - 75
52*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^14 -
 2688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^1
4 - 392*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c
^14 + 9984*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b
^3*c^15 + 4352*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
9*b^4*c^15 + 1344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^8*b^5*c^15 - 5120*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^11*b*c^16 - 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^10*b^2*c^16 - 2176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^9*b^3*c^16 + 1280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^10*b*c^17 - 6*(b^2 - 4*a*c)*a^5*b^11*c^12 + 110*(b^2 - 4*a*c
)*a^6*b^9*c^13 - 784*(b^2 - 4*a*c)*a^7*b^7*c^14 + 2688*(b^2 - 4*a*c)*a^8*b^
5*c^15 - 4352*(b^2 - 4*a*c)*a^9*b^3*c^16 + 2560*(b^2 - 4*a*c)*a^10*b*c^17)*
m)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7 -
sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7)^2 - 4*(a^3*b^4*c^5 - 8*a^
4*b^2*c^6 + 16*a^5*c^7)*(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8)))/(a^2*b

^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8)))/((a^4*b^8*c^8 - 16*a^5*b^6*c^9 - 2*a^4*b^7*c^9 + 96*a^6*b^4*c^10 + 24*a^5*b^5*c^10 + a^4*b^6*c^10 - 256*a^7*b^2*c^11 - 96*a^6*b^3*c^11 - 12*a^5*b^4*c^11 + 256*a^8*c^12 + 128*a^7*b*c^12 + 48*a^6*b^2*c^12 - 64*a^7*c^13)*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)*abs(c)) - 1/16*(4*(b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5 + (b^2*c^3 - 4*a*c^4 - 2*b*c^4 + c^5)*sqrt(b^2 - 4*a*c))*e*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 2*(b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4 + (b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + 4*(a*b^3*c^2 - 4*a^2*b*c^3 - 2*a*b^2*c^3 + a*b*c^4 + (a*b^2*c^2 - 4*a^2*c^3 - 2*a*b*c^3 + a*c^4)*sqrt(b^2 - 4*a*c))*j*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + (b^6 - 10*a*b^4*c - 2*b^5*c + 24*a^2*b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 6*a*b^2*c^3 + (b^5 - 10*a*b^3*c - 2*b^4*c + 24*a^2*b*c^2 + 12*a*b^2*c^2 + b^3*c^2 - 6*a*b*c^3)*sqrt(b^2 - 4*a*c))*l*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 4*(a^2*b^7*c^8 - 12*a^3*b^5*c^9 - 2*a^2*b^6*c^9 + 48*a^4*b^3*c^10 + 16*a^3*b^4*c^10 + a^2*b^5*c^10 - 64*a^5*b*c^11 - 32*a^4*b^2*c^11 - 8*a^3*b^3*c^11 + 16*a^4*b*c^12 - (a^2*b^6*c^8 - 8*a^3*b^4*c^9 - 2*a^2*b^5*c^9 + 16*a^4*b^2*c^10 + 8*a^3*b^3*c^10 + a^2*b^4*c^10 - 4*a^3*b^2*c^11)*sqrt(b^2 - 4*a*c))*e + 2*(a^2*b^8*c^7 - 12*a^3*b^6*c^8 - 2*a^2*b^7*c^8 + 48*a^4*b^4*c^9 + 16*a^3*b^5*c^9 + a^2*b^6*c^9 - 64*a^5*b^2*c^10 - 32*a^4*b^3*c^10 - 8*a^3*b^4*c^10 + 16*a^4*b^2*c^11 + (a^2*b^7*c^7 - 8*a^3*b^5*c^8 - 2*a^2*b^6*c^8 + 16*a^4*b^3*c^9 + 8*a^3*b^4*c^9 + a^2*b^5*c^9 - 4*a^3*b^3*c^10)*sqrt(b^2 - 4*a*c))*g - 4*(a^3*b^7*c^7 - 12*a^4*b^5*c^8 - 2*a^3*b^6*c^8 + 48*a^5*b^3*c^9 + 16*a^4*b^4*c^9 + a^3*b^5*c^9 - 64*a^6*b*c^10 - 32*a^5*b^2*c^10 - 8*a^4*b^3*c^10 + 16*a^5*b*c^11 - (a^3*b^6*c^7 - 8*a^4*b^4*c^8 - 2*a^3*b^5*c^8 + 16*a^5*b^2*c^9 + 8*a^4*b^3*c^9 + a^3*b^4*c^9 - 4*a^4*b^2*c^10)*sqrt(b^2 - 4*a*c))*j - (a^2*b^10*c^5 - 18*a^3*b^8*c^6 - 2*a^2*b^9*c^6 + 120*a^4*b^6*c^7 + 28*a^3*b^7*c^7 + a^2*b^8*c^7 - 352*a^5*b^4*c^8 - 128*a^4*b^5*c^8 - 14*a^3*b^6*c^8 + 384*a^6*b^2*c^9 + 192*a^5*b^3*c^9 + 64*a^4*b^4*c^9 - 96*a^5*b^2*c^10 - (a^2*b^9*c^5 - 14*a^3*b^7*c^6 - 2*a^2*b^8*c^6 + 64*a^4*b^5*c^7 + 20*a^3*b^6*c^7 + a^2*b^7*c^7 - 96*a^5*b^3*c^8 - 48*a^4*b^4*c^8 - 10*a^3*b^5*c^8 + 24*a^4*b^3*c^9)*sqrt(b^2 - 4*a*c))*l)*log(x^2 + 1/2*(a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7 + sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7)^2 - 4*(a^3*b^4*c^5 - 8*a^4*b^2*c^6 + 16*a^5*c^7)*(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8)))/(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8))/((a*b^4*c - 8*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^2*b*c^3 + a*b^2*c^3 - 4*a^2*c^4)*c^2*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)) - 1/16*(4*(b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5 + (b^2*c^3 - 4*a*c^4 - 2*b*c^4 + c^5)*sqrt(b^2 - 4*a*c))*e*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) - 2*(b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4 - (b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + 4*(a*b^3*c^2 - 4*a^2*b*c^3 - 2*a*b^2*c^3 + a*b*c^4 - (a*b^2*c^2 - 4*a^2*c^3 - 2*a*b*c^3 + a*c^4)*sqrt(b^2 - 4*a*c))*j*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7) + (b^6 - 10*a*b^4*c - 2*b^5*c + 24*a^2*b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 6*a*b^2*c^3 - (b^5 - 10*a*b^3*c - 2*b^4*c + 24*a^2*b*c^2 + 12*a*b^2*c^2 + b^3*c^2 - 6*

```
a*b*c^3)*sqrt(b^2 - 4*a*c))*l*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)
 - 4*(a^2*b^7*c^8 - 12*a^3*b^5*c^9 - 2*a^2*b^6*c^9 + 48*a^4*b^3*c^10 + 16*a
^3*b^4*c^10 + a^2*b^5*c^10 - 64*a^5*b*c^11 - 32*a^4*b^2*c^11 - 8*a^3*b^3*c^
11 + 16*a^4*b*c^12 + (a^2*b^6*c^8 - 8*a^3*b^4*c^9 - 2*a^2*b^5*c^9 + 16*a^4*
b^2*c^10 + 8*a^3*b^3*c^10 + a^2*b^4*c^10 - 4*a^3*b^2*c^11)*sqrt(b^2 - 4*a*c
))*e + 2*(a^2*b^8*c^7 - 12*a^3*b^6*c^8 - 2*a^2*b^7*c^8 + 48*a^4*b^4*c^9 + 1
6*a^3*b^5*c^9 + a^2*b^6*c^9 - 64*a^5*b^2*c^10 - 32*a^4*b^3*c^10 - 8*a^3*b^4
*c^10 + 16*a^4*b^2*c^11 - (a^2*b^7*c^7 - 8*a^3*b^5*c^8 - 2*a^2*b^6*c^8 + 16
*a^4*b^3*c^9 + 8*a^3*b^4*c^9 + a^2*b^5*c^9 - 4*a^3*b^3*c^10)*sqrt(b^2 - 4*a
*c))*g - 4*(a^3*b^7*c^7 - 12*a^4*b^5*c^8 - 2*a^3*b^6*c^8 + 48*a^5*b^3*c^9 +
 16*a^4*b^4*c^9 + a^3*b^5*c^9 - 64*a^6*b*c^10 - 32*a^5*b^2*c^10 - 8*a^4*b^3
*c^10 + 16*a^5*b*c^11 - (a^3*b^6*c^7 - 8*a^4*b^4*c^8 - 2*a^3*b^5*c^8 + 16*a
^5*b^2*c^9 + 8*a^4*b^3*c^9 + a^3*b^4*c^9 - 4*a^4*b^2*c^10)*sqrt(b^2 - 4*a*c
))*j - (a^2*b^10*c^5 - 18*a^3*b^8*c^6 - 2*a^2*b^9*c^6 + 120*a^4*b^6*c^7 + 2
8*a^3*b^7*c^7 + a^2*b^8*c^7 - 352*a^5*b^4*c^8 - 128*a^4*b^5*c^8 - 14*a^3*b^
6*c^8 + 384*a^6*b^2*c^9 + 192*a^5*b^3*c^9 + 64*a^4*b^4*c^9 - 96*a^5*b^2*c^1
0 - (a^2*b^9*c^5 - 14*a^3*b^7*c^6 - 2*a^2*b^8*c^6 + 64*a^4*b^5*c^7 + 20*a^3
*b^6*c^7 + a^2*b^7*c^7 - 96*a^5*b^3*c^8 - 48*a^4*b^4*c^8 - 10*a^3*b^5*c^8 +
 24*a^4*b^3*c^9)*sqrt(b^2 - 4*a*c))*l)*log(x^2 + 1/2*(a^2*b^5*c^5 - 8*a^3*b
^3*c^6 + 16*a^4*b*c^7 - sqrt((a^2*b^5*c^5 - 8*a^3*b^3*c^6 + 16*a^4*b*c^7)^2
 - 4*(a^3*b^4*c^5 - 8*a^4*b^2*c^6 + 16*a^5*c^7)*(a^2*b^4*c^6 - 8*a^3*b^2*c^
7 + 16*a^4*c^8)))/(a^2*b^4*c^6 - 8*a^3*b^2*c^7 + 16*a^4*c^8))/((a*b^4*c - 8
*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^2*b*c^3 + a*b^2*c^3 - 4*a^2*c
^4)*c^2*abs(a^2*b^4*c^5 - 8*a^3*b^2*c^6 + 16*a^4*c^7)) - 1/2*(a*b*c^2*e - 2
*a^2*c^2*g + a^2*b*c*j - a^2*b^2*l + 2*a^3*c*l - (b*c^3*d - 2*a*c^3*f + a*b
*c^2*h - a*b^2*c*k + 2*a^2*c^2*k + a*b^3*m - 3*a^2*b*c*m)*x^3 + (2*a*c^3*e
- a*b*c^2*g + a*b^2*c*j - 2*a^2*c^2*j - a*b^3*l + 3*a^2*b*c*l)*x^2 - (b^2*c
^2*d - 2*a*c^3*d - a*b*c^2*f + 2*a^2*c^2*h - a^2*b*c*k + a^2*b^2*m - 2*a^3*
c*m)*x)/((c*x^4 + b*x^2 + a)*(b^2 - 4*a*c)*a*c^2)
```

Mupad [B] (verification not implemented)

Time = 22.90 (sec) , antiderivative size = 82785, normalized size of antiderivative = 107.51

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4)^2,x)
```

```
[Out] symsum(log(root(1572864*a^8*b^2*c^10*z^4 - 983040*a^7*b^4*c^9*z^4 + 327680*
a^6*b^6*c^8*z^4 - 61440*a^5*b^8*c^7*z^4 + 6144*a^4*b^10*c^6*z^4 - 256*a^3*b
^12*c^5*z^4 - 1048576*a^9*c^11*z^4 - 1572864*a^8*b^2*c^8*l*z^3 + 983040*a^7
*b^4*c^7*l*z^3 - 327680*a^6*b^6*c^6*l*z^3 + 61440*a^5*b^8*c^5*l*z^3 - 6144*
a^4*b^10*c^4*l*z^3 + 256*a^3*b^12*c^3*l*z^3 + 1048576*a^9*c^9*l*z^3 + 96*a^
```

3*b^12*c*k*m*z^2 + 98304*a^8*b*c^7*j*l*z^2 + 24576*a^8*b*c^7*h*m*z^2 + 155648*a^7*b*c^8*d*m*z^2 + 98304*a^7*b*c^8*e*l*z^2 + 57344*a^7*b*c^8*f*k*z^2 + 32768*a^7*b*c^8*g*j*z^2 + 57344*a^6*b*c^9*d*h*z^2 + 32768*a^6*b*c^9*e*g*z^2 - 32*a*b^10*c^5*d*f*z^2 - 491520*a^8*b^2*c^6*k*m*z^2 + 358400*a^7*b^4*c^5*k*m*z^2 - 129024*a^6*b^6*c^4*k*m*z^2 + 24768*a^5*b^8*c^3*k*m*z^2 - 2432*a^4*b^10*c^2*k*m*z^2 - 90112*a^7*b^3*c^6*j*l*z^2 + 30720*a^6*b^5*c^5*j*l*z^2 - 4608*a^5*b^7*c^4*j*l*z^2 + 256*a^4*b^9*c^3*j*l*z^2 - 21504*a^6*b^5*c^5*h*m*z^2 + 9216*a^5*b^7*c^4*h*m*z^2 + 8192*a^7*b^3*c^6*h*m*z^2 - 1568*a^4*b^9*c^3*h*m*z^2 + 96*a^3*b^11*c^2*h*m*z^2 - 172032*a^7*b^2*c^7*f*m*z^2 + 116736*a^6*b^4*c^6*f*m*z^2 - 49152*a^7*b^2*c^7*g*l*z^2 + 45056*a^6*b^4*c^6*g*l*z^2 - 35840*a^5*b^6*c^5*f*m*z^2 + 24576*a^7*b^2*c^7*h*k*z^2 - 15360*a^5*b^6*c^5*g*l*z^2 + 5184*a^4*b^8*c^4*f*m*z^2 - 3072*a^5*b^6*c^5*h*k*z^2 + 2304*a^4*b^8*c^4*g*l*z^2 + 2048*a^6*b^4*c^6*h*k*z^2 + 576*a^4*b^8*c^4*h*k*z^2 - 288*a^3*b^10*c^3*f*m*z^2 - 128*a^3*b^10*c^3*g*l*z^2 - 32*a^3*b^10*c^3*h*k*z^2 - 147456*a^6*b^3*c^7*d*m*z^2 - 90112*a^6*b^3*c^7*e*l*z^2 + 52224*a^5*b^5*c^6*d*m*z^2 - 49152*a^6*b^3*c^7*f*k*z^2 + 30720*a^5*b^5*c^6*e*l*z^2 - 24576*a^6*b^3*c^7*g*j*z^2 + 15360*a^5*b^5*c^6*f*k*z^2 - 8192*a^4*b^7*c^5*d*m*z^2 + 6144*a^5*b^5*c^6*g*j*z^2 - 4608*a^4*b^7*c^5*e*l*z^2 - 2048*a^4*b^7*c^5*f*k*z^2 - 512*a^4*b^7*c^5*g*j*z^2 + 480*a^3*b^9*c^4*d*m*z^2 + 256*a^3*b^9*c^4*e*l*z^2 + 96*a^3*b^9*c^4*f*k*z^2 + 131072*a^6*b^2*c^8*d*k*z^2 + 49152*a^6*b^2*c^8*e*j*z^2 - 43008*a^5*b^4*c^7*d*k*z^2 - 12288*a^5*b^4*c^7*e*j*z^2 + 6144*a^4*b^6*c^6*d*k*z^2 + 1024*a^4*b^6*c^6*e*j*z^2 - 320*a^3*b^8*c^5*d*k*z^2 + 6144*a^5*b^4*c^7*f*h*z^2 - 2048*a^4*b^6*c^6*f*h*z^2 + 192*a^3*b^8*c^5*f*h*z^2 - 49152*a^5*b^3*c^8*d*h*z^2 - 24576*a^5*b^3*c^8*e*g*z^2 + 15360*a^4*b^5*c^7*d*h*z^2 + 6144*a^4*b^5*c^7*e*g*z^2 - 2048*a^3*b^7*c^6*d*h*z^2 - 512*a^3*b^7*c^6*e*g*z^2 + 96*a^2*b^9*c^5*d*h*z^2 + 24576*a^5*b^2*c^9*d*f*z^2 - 3072*a^3*b^6*c^7*d*f*z^2 + 2048*a^4*b^4*c^8*d*f*z^2 + 576*a^2*b^8*c^6*d*f*z^2 - 430080*a^9*b*c^6*m^2*z^2 + 3408*a^4*b^11*c*m^2*z^2 - 64*a^3*b^12*c*l^2*z^2 + 61440*a^8*b*c^7*k^2*z^2 + 12288*a^7*b*c^8*h^2*z^2 + 12288*a^6*b*c^9*f^2*z^2 + 61440*a^5*b*c^10*d^2*z^2 + 432*a*b^9*c^6*d^2*z^2 + 245760*a^9*c^7*k*m*z^2 + 81920*a^8*c^8*f*m*z^2 - 49152*a^8*c^8*h*k*z^2 - 147456*a^7*c^9*d*k*z^2 - 65536*a^7*c^9*e*j*z^2 - 16384*a^7*c^9*f*h*z^2 - 49152*a^6*c^10*d*f*z^2 + 716800*a^8*b^3*c^5*m^2*z^2 - 483840*a^7*b^5*c^4*m^2*z^2 + 170496*a^6*b^7*c^3*m^2*z^2 - 33232*a^5*b^9*c^2*m^2*z^2 + 516096*a^8*b^2*c^6*l^2*z^2 - 288768*a^7*b^4*c^5*l^2*z^2 + 88576*a^6*b^6*c^4*l^2*z^2 - 15744*a^5*b^8*c^3*l^2*z^2 + 1536*a^4*b^10*c^2*l^2*z^2 - 61440*a^7*b^3*c^6*k^2*z^2 + 24064*a^6*b^5*c^5*k^2*z^2 - 4608*a^5*b^7*c^4*k^2*z^2 + 432*a^4*b^9*c^3*k^2*z^2 - 16*a^3*b^11*c^2*k^2*z^2 + 24576*a^7*b^2*c^7*j^2*z^2 - 6144*a^6*b^4*c^6*j^2*z^2 + 512*a^5*b^6*c^5*j^2*z^2 - 8192*a^6*b^3*c^7*h^2*z^2 + 1536*a^5*b^5*c^6*h^2*z^2 - 16*a^3*b^9*c^4*h^2*z^2 - 8192*a^6*b^2*c^8*g^2*z^2 + 6144*a^5*b^4*c^7*g^2*z^2 - 1536*a^4*b^6*c^6*g^2*z^2 + 128*a^3*b^8*c^5*g^2*z^2 - 8192*a^5*b^3*c^8*f^2*z^2 + 1536*a^4*b^5*c^7*f^2*z^2 - 16*a^2*b^9*c^5*f^2*z^2 + 24576*a^5*b^2*c^9*e^2*z^2 - 6144*a^4*b^4*c^8*e^2*z^2 + 512*a^3*b^6*c^7*e^2*z^2 - 61440*a^4*b^3*c^9*d^2*z^2 + 24064*a^3*b^5*c^8*d^2*z^2 - 4608*a^2*b^7*c^7*d^2*z^2 - 393216*a^9*c^7*l^2*z^2 - 144*a^3*b^13*m^2*z^2 - 32768*a^8*c^8*j^2*z

^2 - 32768*a^6*c^10*e^2*z^2 - 16*b^11*c^5*d^2*z^2 + 18432*a^8*b*c^5*h*l*m*z
 - 96*a^3*b^10*c*g*k*m*z + 90112*a^7*b*c^6*e*k*m*z + 36864*a^7*b*c^6*f*j*m*
z - 16384*a^7*b*c^6*g*j*l*z + 14336*a^7*b*c^6*d*l*m*z - 10240*a^7*b*c^6*f*k
*l*z + 4096*a^7*b*c^6*h*j*k*z + 10240*a^7*b*c^6*g*h*m*z - 47104*a^6*b*c^7*d
*h*l*z + 36864*a^6*b*c^7*e*f*m*z + 30720*a^6*b*c^7*d*g*m*z - 16384*a^6*b*c^
7*e*g*l*z + 6144*a^6*b*c^7*f*g*k*z + 4096*a^6*b*c^7*e*h*k*z + 32*a*b^10*c^3
*d*f*l*z - 4096*a^5*b*c^8*d*f*j*z - 6144*a^5*b*c^8*d*g*h*z - 32*a*b^8*c^5*d
*f*g*z - 4096*a^4*b*c^9*d*e*f*z + 64*a*b^7*c^6*d*e*f*z + 110592*a^8*b^2*c^4
*k*l*m*z - 36864*a^7*b^4*c^3*k*l*m*z + 5376*a^6*b^6*c^2*k*l*m*z - 79872*a^7
*b^3*c^4*j*k*m*z + 26112*a^6*b^5*c^3*j*k*m*z - 3712*a^5*b^7*c^2*j*k*m*z - 1
3824*a^7*b^3*c^4*h*l*m*z + 3456*a^6*b^5*c^3*h*l*m*z - 288*a^5*b^7*c^2*h*l*m
*z - 45056*a^7*b^2*c^5*g*k*m*z + 39936*a^6*b^4*c^4*g*k*m*z + 30720*a^7*b^2*
c^5*f*l*m*z - 18432*a^7*b^2*c^5*h*k*l*z - 13056*a^5*b^6*c^3*g*k*m*z - 7680*
a^6*b^4*c^4*f*l*m*z + 5376*a^6*b^4*c^4*h*j*m*z + 4608*a^6*b^4*c^4*h*k*l*z +
 3072*a^7*b^2*c^5*h*j*m*z - 1984*a^5*b^6*c^3*h*j*m*z + 1856*a^4*b^8*c^2*g*k
*m*z + 640*a^5*b^6*c^3*f*l*m*z - 384*a^5*b^6*c^3*h*k*l*z + 192*a^4*b^8*c^2*
h*j*m*z - 79872*a^6*b^3*c^5*e*k*m*z - 27648*a^6*b^3*c^5*f*j*m*z + 26112*a^5
*b^5*c^4*e*k*m*z + 12288*a^6*b^3*c^5*g*j*l*z - 10752*a^6*b^3*c^5*d*l*m*z +
7680*a^6*b^3*c^5*f*k*l*z + 6912*a^5*b^5*c^4*f*j*m*z - 3712*a^4*b^7*c^3*e*k*
m*z - 3072*a^6*b^3*c^5*h*j*k*z - 3072*a^5*b^5*c^4*g*j*l*z + 2688*a^5*b^5*c^
4*d*l*m*z - 1920*a^5*b^5*c^4*f*k*l*z + 768*a^5*b^5*c^4*h*j*k*z - 576*a^4*b^
7*c^3*f*j*m*z + 256*a^4*b^7*c^3*g*j*l*z - 224*a^4*b^7*c^3*d*l*m*z + 192*a^3
*b^9*c^2*e*k*m*z + 160*a^4*b^7*c^3*f*k*l*z - 64*a^4*b^7*c^3*h*j*k*z - 2688*
a^5*b^5*c^4*g*h*m*z - 1536*a^6*b^3*c^5*g*h*m*z + 992*a^4*b^7*c^3*g*h*m*z -
96*a^3*b^9*c^2*g*h*m*z - 65536*a^6*b^2*c^6*d*k*l*z + 46080*a^6*b^2*c^6*d*j*
m*z - 24576*a^6*b^2*c^6*e*j*l*z + 21504*a^5*b^4*c^5*d*k*l*z - 11520*a^5*b^4
*c^5*d*j*m*z + 9216*a^6*b^2*c^6*f*j*k*z + 6144*a^5*b^4*c^5*e*j*l*z - 3072*a
^4*b^6*c^4*d*k*l*z - 2304*a^5*b^4*c^5*f*j*k*z + 960*a^4*b^6*c^4*d*j*m*z - 5
12*a^4*b^6*c^4*e*j*l*z + 192*a^4*b^6*c^4*f*j*k*z + 160*a^3*b^8*c^3*d*k*l*z
- 18432*a^6*b^2*c^6*f*g*m*z + 13824*a^5*b^4*c^5*f*g*m*z + 5376*a^5*b^4*c^5*
e*h*m*z - 3456*a^4*b^6*c^4*f*g*m*z + 3072*a^6*b^2*c^6*e*h*m*z - 3072*a^5*b^
4*c^5*f*h*l*z - 2048*a^6*b^2*c^6*g*h*k*z - 1984*a^4*b^6*c^4*e*h*m*z + 1536*
a^5*b^4*c^5*g*h*k*z + 1024*a^4*b^6*c^4*f*h*l*z - 384*a^4*b^6*c^4*g*h*k*z +
288*a^3*b^8*c^3*f*g*m*z + 192*a^3*b^8*c^3*e*h*m*z - 96*a^3*b^8*c^3*f*h*l*z
+ 32*a^3*b^8*c^3*g*h*k*z + 41472*a^5*b^3*c^6*d*h*l*z - 27648*a^5*b^3*c^6*e*
f*m*z - 23040*a^5*b^3*c^6*d*g*m*z - 13440*a^4*b^5*c^5*d*h*l*z + 12288*a^5*b
^3*c^6*e*g*l*z + 6912*a^4*b^5*c^5*e*f*m*z + 5760*a^4*b^5*c^5*d*g*m*z - 4608
*a^5*b^3*c^6*f*g*k*z - 3072*a^5*b^3*c^6*e*h*k*z - 3072*a^4*b^5*c^5*e*g*l*z
+ 1888*a^3*b^7*c^4*d*h*l*z + 1152*a^4*b^5*c^5*f*g*k*z + 768*a^4*b^5*c^5*e*h
*k*z - 576*a^3*b^7*c^4*e*f*m*z - 480*a^3*b^7*c^4*d*g*m*z + 256*a^3*b^7*c^4*
e*g*l*z - 96*a^3*b^7*c^4*f*g*k*z - 96*a^2*b^9*c^3*d*h*l*z - 64*a^3*b^7*c^4*
e*h*k*z + 46080*a^5*b^2*c^7*d*e*m*z - 11520*a^4*b^4*c^6*d*e*m*z + 9216*a^5*
b^2*c^7*e*f*k*z - 9216*a^5*b^2*c^7*d*h*j*z - 6656*a^4*b^4*c^6*d*f*l*z - 614
4*a^5*b^2*c^7*d*f*l*z + 3456*a^3*b^6*c^5*d*f*l*z - 2304*a^4*b^4*c^6*e*f*k*z
 + 2304*a^4*b^4*c^6*d*h*j*z + 960*a^3*b^6*c^5*d*e*m*z - 576*a^2*b^8*c^4*d*f

*l*z + 192*a^3*b^6*c^5*e*f*k*z - 192*a^3*b^6*c^5*d*h*j*z + 3072*a^4*b^3*c^7
*d*f*j*z - 768*a^3*b^5*c^6*d*f*j*z + 64*a^2*b^7*c^5*d*f*j*z + 4608*a^4*b^3*
c^7*d*g*h*z - 1152*a^3*b^5*c^6*d*g*h*z + 96*a^2*b^7*c^5*d*g*h*z - 9216*a^4*
b^2*c^8*d*e*h*z + 2304*a^3*b^4*c^7*d*e*h*z + 2048*a^4*b^2*c^8*d*f*g*z - 153
6*a^3*b^4*c^7*d*f*g*z + 384*a^2*b^6*c^6*d*f*g*z - 192*a^2*b^6*c^6*d*e*h*z +
 3072*a^3*b^3*c^8*d*e*f*z - 768*a^2*b^5*c^7*d*e*f*z - 288*a^5*b^8*c*k*l*m*z
 + 90112*a^8*b*c^5*j*k*m*z + 192*a^4*b^9*c*j*k*m*z + 138240*a^9*b*c^4*l*m^2
*z - 7344*a^6*b^7*c*l*m^2*z + 5088*a^5*b^8*c*j*m^2*z - 3072*a^8*b*c^5*k^2*l
*z - 49152*a^8*b*c^5*j*l^2*z - 128*a^4*b^9*c*j*l^2*z - 25600*a^8*b*c^5*g*m^
2*z - 9216*a^7*b*c^6*h^2*l*z - 2544*a^4*b^9*c*g*m^2*z + 64*a^3*b^10*c*g*l^2
*z + 9216*a^7*b*c^6*g*k^2*z - 3072*a^6*b*c^7*f^2*l*z - 288*a^3*b^10*c*e*m^2
*z - 49152*a^7*b*c^6*e*l^2*z - 58368*a^5*b*c^8*d^2*l*z - 432*a*b^9*c^4*d^2*
l*z - 1024*a^6*b*c^7*g*h^2*z + 32*a*b^8*c^5*d^2*j*z + 1024*a^5*b*c^8*f^2*g*
z - 9216*a^4*b*c^9*d^2*g*z + 336*a*b^7*c^6*d^2*g*z - 672*a*b^6*c^7*d^2*e*z
- 122880*a^9*c^5*k*l*m*z - 40960*a^8*c^6*f*l*m*z + 24576*a^8*c^6*h*k*l*z -
20480*a^8*c^6*h*j*m*z + 73728*a^7*c^7*d*k*l*z - 61440*a^7*c^7*d*j*m*z + 327
68*a^7*c^7*e*j*l*z - 12288*a^7*c^7*f*j*k*z - 20480*a^7*c^7*e*h*m*z + 8192*a
^7*c^7*f*h*l*z - 61440*a^6*c^8*d*e*m*z + 24576*a^6*c^8*d*f*l*z - 12288*a^6*
c^8*e*f*k*z + 12288*a^6*c^8*d*h*j*z + 12288*a^5*c^9*d*e*h*z - 131328*a^8*b^
3*c^3*l*m^2*z + 46656*a^7*b^5*c^2*l*m^2*z - 142848*a^8*b^2*c^4*j*m^2*z + 10
6368*a^7*b^4*c^3*j*m^2*z - 34208*a^6*b^6*c^2*j*m^2*z + 2304*a^7*b^3*c^4*k^2
*l*z - 576*a^6*b^5*c^3*k^2*l*z + 48*a^5*b^7*c^2*k^2*l*z + 45056*a^7*b^3*c^4
*j*l^2*z - 15360*a^6*b^5*c^3*j*l^2*z - 12288*a^7*b^2*c^5*j^2*l*z + 3072*a^6
*b^4*c^4*j^2*l*z + 2304*a^5*b^7*c^2*j*l^2*z - 256*a^5*b^6*c^3*j^2*l*z + 158
72*a^7*b^2*c^5*j*k^2*z - 4992*a^6*b^4*c^4*j*k^2*z + 672*a^5*b^6*c^3*j*k^2*z
 - 32*a^4*b^8*c^2*j*k^2*z + 71424*a^7*b^3*c^4*g*m^2*z - 53184*a^6*b^5*c^3*g
*m^2*z + 17104*a^5*b^7*c^2*g*m^2*z + 6912*a^6*b^3*c^5*h^2*l*z - 1728*a^5*b^
5*c^4*h^2*l*z + 144*a^4*b^7*c^3*h^2*l*z + 24576*a^7*b^2*c^5*g*l^2*z - 22528
*a^6*b^4*c^4*g*l^2*z + 7680*a^5*b^6*c^3*g*l^2*z + 4096*a^6*b^2*c^6*g^2*l*z
- 3072*a^5*b^4*c^5*g^2*l*z - 1152*a^4*b^8*c^2*g*l^2*z + 768*a^4*b^6*c^4*g^2
*l*z - 64*a^3*b^8*c^3*g^2*l*z - 142848*a^7*b^2*c^5*e*m^2*z + 106368*a^6*b^4
*c^4*e*m^2*z - 34208*a^5*b^6*c^3*e*m^2*z - 7936*a^6*b^3*c^5*g*k^2*z + 5088*
a^4*b^8*c^2*e*m^2*z + 2496*a^5*b^5*c^4*g*k^2*z - 1536*a^6*b^2*c^6*h^2*j*z +
 1280*a^5*b^3*c^6*f^2*l*z + 384*a^5*b^4*c^5*h^2*j*z - 336*a^4*b^7*c^3*g*k^2
*z + 192*a^4*b^5*c^5*f^2*l*z - 144*a^3*b^7*c^4*f^2*l*z - 32*a^4*b^6*c^4*h^2
*j*z + 16*a^3*b^9*c^2*g*k^2*z + 16*a^2*b^9*c^3*f^2*l*z + 45056*a^6*b^3*c^5*
e*l^2*z - 15360*a^5*b^5*c^4*e*l^2*z - 12288*a^5*b^2*c^7*e^2*l*z + 3072*a^4*
b^4*c^6*e^2*l*z + 2304*a^4*b^7*c^3*e*l^2*z - 256*a^3*b^6*c^5*e^2*l*z - 128*
a^3*b^9*c^2*e*l^2*z + 59136*a^4*b^3*c^7*d^2*l*z - 23488*a^3*b^5*c^6*d^2*l*z
 + 15872*a^6*b^2*c^6*e*k^2*z - 4992*a^5*b^4*c^5*e*k^2*z + 4560*a^2*b^7*c^5*
d^2*l*z + 1536*a^5*b^2*c^7*f^2*j*z + 672*a^4*b^6*c^4*e*k^2*z - 384*a^4*b^4*
c^6*f^2*j*z - 32*a^3*b^8*c^3*e*k^2*z + 32*a^3*b^6*c^5*f^2*j*z + 768*a^5*b^3
*c^6*g*h^2*z - 192*a^4*b^5*c^5*g*h^2*z + 16*a^3*b^7*c^4*g*h^2*z - 15872*a^4
*b^2*c^8*d^2*j*z + 4992*a^3*b^4*c^7*d^2*j*z - 672*a^2*b^6*c^6*d^2*j*z - 153
6*a^5*b^2*c^7*e*h^2*z - 768*a^4*b^3*c^7*f^2*g*z + 384*a^4*b^4*c^6*e*h^2*z +

192*a^3*b^5*c^6*f^2*g*z - 32*a^3*b^6*c^5*e*h^2*z - 16*a^2*b^7*c^5*f^2*g*z
+ 7936*a^3*b^3*c^8*d^2*g*z - 2496*a^2*b^5*c^7*d^2*g*z + 1536*a^4*b^2*c^8*e*
f^2*z - 384*a^3*b^4*c^7*e*f^2*z + 32*a^2*b^6*c^6*e*f^2*z - 15872*a^3*b^2*c^
9*d^2*e*z + 4992*a^2*b^4*c^8*d^2*e*z - 61440*a^8*b^2*c^4*l^3*z + 21504*a^7*
b^4*c^3*l^3*z - 3328*a^6*b^6*c^2*l^3*z + 432*a^5*b^9*l*m^2*z + 51200*a^9*c^
5*j*m^2*z + 16384*a^8*c^6*j^2*l*z - 288*a^4*b^10*j*m^2*z - 18432*a^8*c^6*j*
k^2*z + 144*a^3*b^11*g*m^2*z + 51200*a^8*c^6*e*m^2*z + 2048*a^7*c^7*h^2*j*z
+ 16384*a^6*c^8*e^2*l*z + 16*b^11*c^3*d^2*l*z - 18432*a^7*c^7*e*k^2*z - 20
48*a^6*c^8*f^2*j*z + 18432*a^5*c^9*d^2*j*z + 192*a^5*b^8*c*l^3*z + 2048*a^6
*c^8*e*h^2*z - 16*b^9*c^5*d^2*g*z - 2048*a^5*c^9*e*f^2*z + 32*b^8*c^6*d^2*e
*z + 18432*a^4*c^10*d^2*e*z + 65536*a^9*c^5*l^3*z - 11008*a^8*b*c^3*j*k*l*m
- 288*a^6*b^5*c*j*k*l*m + 144*a^5*b^6*c*g*k*l*m - 11008*a^7*b*c^4*e*k*l*m
- 5376*a^7*b*c^4*f*j*l*m + 3840*a^7*b*c^4*g*j*k*m - 3328*a^7*b*c^4*h*j*k*l
- 96*a^4*b^7*c*g*j*k*m - 2560*a^7*b*c^4*g*h*l*m - 36*a^3*b^8*c*f*h*k*m - 69
12*a^6*b*c^5*d*j*k*l - 7872*a^6*b*c^5*d*h*k*m - 7680*a^6*b*c^5*d*g*l*m - 53
76*a^6*b*c^5*e*f*l*m + 3840*a^6*b*c^5*e*g*k*m - 3328*a^6*b*c^5*e*h*k*l - 15
36*a^6*b*c^5*f*g*k*l + 1280*a^6*b*c^5*f*g*j*m - 768*a^6*b*c^5*g*h*j*k - 768
*a^6*b*c^5*f*h*j*l - 768*a^6*b*c^5*e*h*j*m - 36*a^2*b^9*c*d*h*k*m - 6912*a^
5*b*c^6*d*e*k*l - 4864*a^5*b*c^6*d*e*j*m - 2304*a^5*b*c^6*d*g*j*k - 1792*a^
5*b*c^6*e*f*j*k - 1280*a^5*b*c^6*d*f*j*l - 4544*a^5*b*c^6*d*f*h*m + 1536*a^
5*b*c^6*d*g*h*l + 1280*a^5*b*c^6*e*f*g*m - 768*a^5*b*c^6*e*g*h*k - 768*a^5*
b*c^6*e*f*h*l - 256*a^5*b*c^6*f*g*h*j + 12*a*b^9*c^2*d*f*h*m + 16*a*b^8*c^3
*d*f*g*l - 4*a*b^8*c^3*d*f*h*k - 2304*a^4*b*c^7*d*e*g*k - 1792*a^4*b*c^7*d*
e*h*j - 1280*a^4*b*c^7*d*e*f*l - 768*a^4*b*c^7*d*f*g*j - 32*a*b^7*c^4*d*e*f
*l - 256*a^4*b*c^7*e*f*g*h - 768*a^3*b*c^8*d*e*f*g + 32*a*b^5*c^6*d*e*f*g +
12*a*b^10*c*d*f*k*m + 3648*a^7*b^3*c^2*j*k*l*m + 5504*a^7*b^2*c^3*g*k*l*m
- 1824*a^6*b^4*c^2*g*k*l*m + 384*a^7*b^2*c^3*h*j*l*m - 288*a^6*b^4*c^2*h*j*
l*m - 4800*a^6*b^3*c^3*g*j*k*m + 3648*a^6*b^3*c^3*e*k*l*m + 1280*a^5*b^5*c^
2*g*j*k*m + 1088*a^6*b^3*c^3*f*j*l*m + 576*a^6*b^3*c^3*h*j*k*l - 288*a^5*b^
5*c^2*e*k*l*m - 192*a^6*b^3*c^3*g*h*l*m + 144*a^5*b^5*c^2*g*h*l*m + 9600*a^
6*b^2*c^4*e*j*k*m - 4224*a^6*b^2*c^4*d*j*l*m - 2560*a^5*b^4*c^3*e*j*k*m + 3
84*a^6*b^2*c^4*f*j*k*l + 224*a^5*b^4*c^3*d*j*l*m + 192*a^4*b^6*c^2*e*j*k*m
- 160*a^5*b^4*c^3*f*j*k*l - 4608*a^6*b^2*c^4*f*h*k*m + 2688*a^6*b^2*c^4*f*g
*l*m + 1664*a^6*b^2*c^4*g*h*k*l - 744*a^5*b^4*c^3*f*h*k*m - 544*a^5*b^4*c^3
*f*g*l*m + 492*a^4*b^6*c^2*f*h*k*m + 416*a^5*b^4*c^3*g*h*j*m + 384*a^6*b^2*
c^4*g*h*j*m + 384*a^6*b^2*c^4*e*h*l*m - 288*a^5*b^4*c^3*g*h*k*l - 288*a^5*b
^4*c^3*e*h*l*m - 96*a^4*b^6*c^2*g*h*j*m + 2112*a^5*b^3*c^4*d*j*k*l - 160*a^
4*b^5*c^3*d*j*k*l + 16992*a^5*b^3*c^4*d*h*k*m - 6252*a^4*b^5*c^3*d*h*k*m -
4800*a^5*b^3*c^4*e*g*k*m + 2112*a^5*b^3*c^4*d*g*l*m - 1728*a^5*b^3*c^4*f*g*
j*m + 1280*a^4*b^5*c^3*e*g*k*m + 1088*a^5*b^3*c^4*e*f*l*m - 832*a^5*b^3*c^4
*e*h*j*m + 816*a^3*b^7*c^2*d*h*k*m + 576*a^5*b^3*c^4*e*h*k*l - 448*a^5*b^3*
c^4*f*h*j*l + 288*a^4*b^5*c^3*f*g*j*m - 192*a^5*b^3*c^4*g*h*j*k - 192*a^5*b
^3*c^4*f*g*k*l + 192*a^4*b^5*c^3*e*h*j*m - 112*a^4*b^5*c^3*d*g*l*m + 96*a^4
*b^5*c^3*f*h*j*l - 96*a^3*b^7*c^2*e*g*k*m + 80*a^4*b^5*c^3*f*g*k*l + 32*a^4
*b^5*c^3*g*h*j*k - 11456*a^5*b^2*c^5*d*f*k*m + 4992*a^5*b^2*c^5*d*h*j*l - 4

608*a^5*b^2*c^5*e*g*j*l - 4224*a^5*b^2*c^5*d*e*l*m + 3456*a^5*b^2*c^5*e*f*j
*m + 3456*a^5*b^2*c^5*d*g*k*l + 2432*a^5*b^2*c^5*d*g*j*m - 1312*a^4*b^4*c^4
*d*h*j*l + 1272*a^3*b^6*c^3*d*f*k*m - 1056*a^4*b^4*c^4*d*g*k*l + 896*a^5*b^
2*c^5*f*g*j*k + 768*a^4*b^4*c^4*e*g*j*l - 576*a^4*b^4*c^4*e*f*j*m - 480*a^4
*b^4*c^4*d*g*j*m + 384*a^5*b^2*c^5*e*h*j*k + 384*a^5*b^2*c^5*e*f*k*l - 232*
a^2*b^8*c^2*d*f*k*m + 224*a^4*b^4*c^4*d*e*l*m - 160*a^4*b^4*c^4*e*f*k*l - 9
6*a^4*b^4*c^4*f*g*j*k + 96*a^3*b^6*c^3*d*h*j*l + 80*a^3*b^6*c^3*d*g*k*l - 6
4*a^4*b^4*c^4*e*h*j*k - 24*a^4*b^4*c^4*d*f*k*m + 416*a^4*b^4*c^4*e*g*h*m +
384*a^5*b^2*c^5*f*g*h*l + 384*a^5*b^2*c^5*e*g*h*m + 224*a^4*b^4*c^4*f*g*h*l
 - 96*a^3*b^6*c^3*e*g*h*m - 48*a^3*b^6*c^3*f*g*h*l + 2112*a^4*b^3*c^5*d*e*k
*l - 960*a^4*b^3*c^5*d*f*j*l + 960*a^4*b^3*c^5*d*e*j*m + 384*a^3*b^5*c^4*d*
f*j*l + 320*a^4*b^3*c^5*d*g*j*k + 192*a^4*b^3*c^5*e*f*j*k - 160*a^3*b^5*c^4
*d*e*k*l - 32*a^2*b^7*c^3*d*f*j*l + 7392*a^4*b^3*c^5*d*f*h*m - 2496*a^4*b^3
*c^5*d*g*h*l - 1728*a^4*b^3*c^5*e*f*g*m - 1500*a^3*b^5*c^4*d*f*h*m + 656*a^
3*b^5*c^4*d*g*h*l - 448*a^4*b^3*c^5*e*f*h*l + 288*a^3*b^5*c^4*e*f*g*m - 192
*a^4*b^3*c^5*f*g*h*j - 192*a^4*b^3*c^5*e*g*h*k + 96*a^3*b^5*c^4*e*f*h*l - 4
8*a^2*b^7*c^3*d*g*h*l + 32*a^3*b^5*c^4*e*g*h*k - 16*a^2*b^7*c^3*d*f*h*m - 6
40*a^4*b^2*c^6*d*e*j*k + 4992*a^4*b^2*c^6*d*e*h*l - 3584*a^4*b^2*c^6*d*f*h*
k + 2432*a^4*b^2*c^6*d*e*g*m - 1312*a^3*b^4*c^5*d*e*h*l + 896*a^4*b^2*c^6*e
*f*g*k + 896*a^4*b^2*c^6*d*g*h*j + 640*a^4*b^2*c^6*d*f*g*l + 600*a^3*b^4*c^
5*d*f*h*k + 480*a^3*b^4*c^5*d*f*g*l - 480*a^3*b^4*c^5*d*e*g*m + 384*a^4*b^2
*c^6*e*f*h*j - 192*a^2*b^6*c^4*d*f*g*l - 96*a^3*b^4*c^5*e*f*g*k - 96*a^3*b^
4*c^5*d*g*h*j + 96*a^2*b^6*c^4*d*e*h*l + 12*a^2*b^6*c^4*d*f*h*k - 960*a^3*b
^3*c^6*d*e*f*l + 384*a^2*b^5*c^5*d*e*f*l + 320*a^3*b^3*c^6*d*e*g*k - 192*a^
3*b^3*c^6*d*f*g*j + 192*a^3*b^3*c^6*d*e*h*j + 32*a^2*b^5*c^5*d*f*g*j - 192*
a^3*b^3*c^6*e*f*g*h + 384*a^3*b^2*c^7*d*e*f*j - 64*a^2*b^4*c^6*d*e*f*j + 89
6*a^3*b^2*c^7*d*e*g*h - 96*a^2*b^4*c^6*d*e*g*h - 192*a^2*b^3*c^7*d*e*f*g +
496*a^7*b^4*c*k*l^2*m - 4752*a^7*b^4*c*j*l*m^2 + 96*a^5*b^6*c*j^2*k*m - 614
4*a^8*b*c^3*h*l^2*m - 168*a^6*b^5*c*h*l^2*m + 6400*a^8*b*c^3*g*l*m^2 - 2862
*a^6*b^5*c*h*k*m^2 + 2376*a^6*b^5*c*g*l*m^2 - 1632*a^7*b*c^4*h^2*k*m - 480*
a^8*b*c^3*h*k*m^2 - 180*a^5*b^6*c*h*k^2*m + 54*a^4*b^7*c*h^2*k*m - 384*a^7*
b*c^4*h*j^2*m + 120*a^5*b^6*c*h*k*l^2 + 56*a^5*b^6*c*f*l^2*m + 24*a^3*b^8*c
*g^2*k*m + 4512*a^7*b*c^4*f*k^2*m - 2304*a^7*b*c^4*g*k^2*l - 1680*a^5*b^6*c
*g*j*m^2 + 1184*a^6*b*c^5*f^2*k*m + 804*a^5*b^6*c*f*k*m^2 + 432*a^5*b^6*c*e
*l*m^2 + 60*a^4*b^7*c*f*k^2*m + 6*a^2*b^9*c*f^2*k*m - 13312*a^7*b*c^4*d*l^2
*m + 2048*a^7*b*c^4*g*j*l^2 - 1024*a^7*b*c^4*f*k*l^2 + 64*a^4*b^7*c*g*j*l^2
 + 56*a^4*b^7*c*d*l^2*m - 40*a^4*b^7*c*f*k*l^2 + 13440*a^7*b*c^4*e*j*m^2 -
8928*a^5*b*c^6*d^2*k*m - 6240*a^7*b*c^4*d*k*m^2 + 1614*a^4*b^7*c*d*k*m^2 -
288*a^4*b^7*c*e*j*m^2 - 170*a*b^9*c^2*d^2*k*m + 60*a^3*b^8*c*d*k^2*m + 4608
*a^6*b*c^5*e*j^2*l + 4608*a^5*b*c^6*e^2*j*l - 2432*a^6*b*c^5*d*j^2*m + 1440
*a^7*b*c^4*f*h*m^2 - 896*a^6*b*c^5*f*j^2*k - 864*a^6*b*c^5*f*h^2*m - 558*a^
4*b^7*c*f*h*m^2 + 256*a^6*b*c^5*g*h^2*l - 40*a^3*b^8*c*d*k*l^2 - 1920*a^6*b
*c^5*e*j*k^2 - 384*a^5*b*c^6*e^2*h*m + 24*a^3*b^8*c*f*h*l^2 - 16*a*b^8*c^3*
d^2*j*l + 2208*a^6*b*c^5*f*h*k^2 - 1044*a^3*b^8*c*d*h*m^2 + 800*a^5*b*c^6*f
^2*h*k - 256*a^5*b*c^6*f^2*g*l + 144*a^3*b^8*c*e*g*m^2 - 116*a*b^8*c^3*d^2*

h*m + 8192*a^6*b*c^5*d*h*l^2 + 2048*a^6*b*c^5*e*g*l^2 + 24*a^2*b^9*c*d*h*l^
2 - 5856*a^4*b*c^7*d^2*f*m + 4896*a^4*b*c^7*d^2*h*k + 2720*a^6*b*c^5*d*f*m^
2 + 2304*a^4*b*c^7*d^2*g*l + 1824*a^5*b*c^6*d*h^2*k + 438*a*b^7*c^4*d^2*f*m
 - 384*a^5*b*c^6*e*h^2*j + 318*a^2*b^9*c*d*f*m^2 - 168*a*b^7*c^4*d^2*g*l +
42*a*b^7*c^4*d^2*h*k - 36*a*b^8*c^3*d*f^2*m - 2432*a^4*b*c^7*d*e^2*m + 1536
*a^5*b*c^6*e*g*j^2 + 1536*a^4*b*c^7*e^2*g*j - 896*a^5*b*c^6*d*h*j^2 - 896*a
^4*b*c^7*e^2*f*k + 4896*a^5*b*c^6*d*f*k^2 + 1824*a^4*b*c^7*d*f^2*k - 384*a^
4*b*c^7*e*f^2*j + 336*a*b^6*c^5*d^2*e*l - 156*a*b^6*c^5*d^2*f*k + 16*a*b^6*
c^5*d^2*g*j + 12*a*b^7*c^4*d*f^2*k - 2*a*b^9*c^2*d*f*k^2 - 1920*a^3*b*c^8*d
^2*e*j - 32*a*b^5*c^6*d^2*e*j + 2208*a^3*b*c^8*d^2*f*h + 800*a^4*b*c^7*d*f*
h^2 - 102*a*b^5*c^6*d^2*f*h + 12*a*b^6*c^5*d*f^2*h - 2*a*b^7*c^4*d*f*h^2 -
896*a^3*b*c^8*d*e^2*h - 8*a*b^6*c^5*d*f*g^2 - 240*a*b^4*c^7*d^2*e*g - 32*a*
b^4*c^7*d*e^2*f + 5120*a^8*c^4*h*j*l*m + 15360*a^7*c^5*d*j*l*m - 7680*a^7*c
^5*e*j*k*m + 3072*a^7*c^5*f*j*k*l + 5120*a^7*c^5*e*h*l*m + 1920*a^7*c^5*f*h
*k*m + 15360*a^6*c^6*d*e*l*m + 5760*a^6*c^6*d*f*k*m + 3072*a^6*c^6*e*f*k*l
- 3072*a^6*c^6*d*h*j*l - 2560*a^6*c^6*e*f*j*m + 1536*a^6*c^6*e*h*j*k + 4608
*a^5*c^7*d*e*j*k - 3072*a^5*c^7*d*e*h*l - 1152*a^5*c^7*d*f*h*k + 512*a^5*c^
7*e*f*h*j + 1536*a^4*c^8*d*e*f*j - 8*a*b^10*c*d*f*l^2 - 5568*a^8*b^2*c^2*k*
l^2*m + 15552*a^8*b^2*c^2*j*l*m^2 + 4800*a^7*b^2*c^3*j^2*k*m - 1280*a^6*b^4
*c^2*j^2*k*m + 2080*a^7*b^3*c^2*h*l^2*m - 1088*a^7*b^2*c^3*j*k^2*l + 48*a^6
*b^4*c^2*j*k^2*l - 8544*a^7*b^2*c^3*h*k^2*m - 7776*a^7*b^3*c^2*g*l*m^2 + 76
32*a^7*b^3*c^2*h*k*m^2 + 3600*a^6*b^3*c^3*h^2*k*m + 2484*a^6*b^4*c^2*h*k^2*
m - 918*a^5*b^5*c^2*h^2*k*m + 4800*a^7*b^2*c^3*h*k*l^2 - 1424*a^6*b^4*c^2*h
*k*l^2 + 1200*a^5*b^4*c^3*g^2*k*m - 960*a^6*b^2*c^4*g^2*k*m - 528*a^6*b^4*c
^2*f*l^2*m - 416*a^6*b^3*c^3*h*j^2*m - 320*a^4*b^6*c^2*g^2*k*m + 192*a^7*b^
2*c^3*f*l^2*m + 96*a^5*b^5*c^2*h*j^2*m + 15552*a^7*b^2*c^3*e*l*m^2 - 6720*a
^7*b^2*c^3*g*j*m^2 + 6160*a^6*b^4*c^2*g*j*m^2 - 4752*a^6*b^4*c^2*e*l*m^2 -
2016*a^7*b^2*c^3*f*k*m^2 - 1164*a^6*b^4*c^2*f*k*m^2 + 1104*a^5*b^3*c^4*f^2*
k*m + 1008*a^6*b^3*c^3*f*k^2*m + 960*a^6*b^2*c^4*h^2*j*l - 678*a^5*b^5*c^2*
f*k^2*m + 544*a^6*b^3*c^3*g*k^2*l - 144*a^5*b^4*c^3*h^2*j*l - 102*a^4*b^5*c
^3*f^2*k*m - 62*a^3*b^7*c^2*f^2*k*m - 24*a^5*b^5*c^2*g*k^2*l + 6432*a^6*b^3
*c^3*d*l^2*m + 4800*a^5*b^2*c^5*e^2*k*m - 2304*a^6*b^2*c^4*g*j^2*l + 1920*a
^6*b^3*c^3*g*j*l^2 + 1728*a^6*b^2*c^4*f*j^2*m - 1280*a^4*b^4*c^4*e^2*k*m +
1152*a^5*b^3*c^4*g^2*j*l - 1032*a^5*b^5*c^2*d*l^2*m - 864*a^6*b^3*c^3*f*k*l
^2 - 768*a^5*b^5*c^2*g*j*l^2 + 408*a^5*b^5*c^2*f*k*l^2 + 384*a^5*b^4*c^3*g*
j^2*l - 288*a^5*b^4*c^3*f*j^2*m + 192*a^6*b^2*c^4*h*j^2*k - 192*a^4*b^5*c^3
*g^2*j*l + 96*a^3*b^6*c^3*e^2*k*m - 32*a^5*b^4*c^3*h*j^2*k - 21120*a^6*b^2*
c^4*d*k^2*m + 20880*a^6*b^3*c^3*d*k*m^2 + 19760*a^4*b^3*c^5*d^2*k*m - 12320
*a^6*b^3*c^3*e*j*m^2 - 9750*a^5*b^5*c^2*d*k*m^2 - 9390*a^3*b^5*c^4*d^2*k*m
+ 8460*a^5*b^4*c^3*d*k^2*m + 3360*a^5*b^5*c^2*e*j*m^2 + 1860*a^2*b^7*c^3*d^
2*k*m - 1218*a^4*b^6*c^2*d*k^2*m - 1088*a^6*b^2*c^4*e*k^2*l + 960*a^6*b^2*c
^4*g*j*k^2 - 240*a^5*b^4*c^3*g*j*k^2 + 192*a^5*b^2*c^5*f^2*j*l - 104*a^4*b^
5*c^3*g^2*h*m - 96*a^5*b^3*c^4*g^2*h*m + 48*a^5*b^4*c^3*e*k^2*l + 48*a^4*b^
4*c^4*f^2*j*l + 24*a^3*b^7*c^2*g^2*h*m + 16*a^4*b^6*c^2*g*j*k^2 - 16*a^3*b^
6*c^3*f^2*j*l + 13376*a^6*b^2*c^4*d*k*l^2 - 5136*a^5*b^4*c^3*d*k*l^2 - 3840

*a^6*b^2*c^4*e*j*l^2 + 1536*a^5*b^4*c^3*e*j*l^2 + 1392*a^5*b^3*c^4*f*h^2*m
+ 1386*a^5*b^5*c^2*f*h*m^2 - 768*a^5*b^3*c^4*e*j^2*l + 768*a^4*b^6*c^2*d*k*
l^2 - 768*a^4*b^3*c^5*e^2*j*l - 588*a^4*b^4*c^4*f^2*h*m - 480*a^5*b^3*c^4*g
*h^2*l + 480*a^5*b^3*c^4*d*j^2*m - 480*a^5*b^2*c^5*f^2*h*m - 128*a^4*b^6*c^
2*e*j*l^2 + 100*a^3*b^6*c^3*f^2*h*m + 96*a^5*b^3*c^4*f*j^2*k + 72*a^4*b^5*c
^3*g*h^2*l - 54*a^4*b^5*c^3*f*h^2*m - 48*a^6*b^3*c^3*f*h*m^2 - 36*a^3*b^7*c
^2*f*h^2*m + 6*a^2*b^8*c^2*f^2*h*m + 6848*a^4*b^2*c^6*d^2*j*l - 2448*a^3*b^
4*c^5*d^2*j*l + 624*a^5*b^4*c^3*f*h*l^2 + 576*a^6*b^2*c^4*f*h*l^2 + 480*a^5
*b^3*c^4*e*j*k^2 + 432*a^4*b^4*c^4*f*g^2*m - 416*a^4*b^3*c^5*e^2*h*m + 336*
a^2*b^6*c^4*d^2*j*l - 320*a^5*b^2*c^5*f*g^2*m - 256*a^4*b^6*c^2*f*h*l^2 + 1
92*a^5*b^2*c^5*g^2*h*k + 96*a^3*b^5*c^4*e^2*h*m - 72*a^3*b^6*c^3*f*g^2*m +
48*a^4*b^4*c^4*g^2*h*k - 32*a^4*b^5*c^3*e*j*k^2 - 8*a^3*b^6*c^3*g^2*h*k + 2
4768*a^6*b^2*c^4*d*h*m^2 - 21108*a^5*b^4*c^3*d*h*m^2 - 10048*a^4*b^2*c^6*d^
2*h*m + 7218*a^4*b^6*c^2*d*h*m^2 - 6720*a^6*b^2*c^4*e*g*m^2 + 6160*a^5*b^4*
c^3*e*g*m^2 - 2592*a^5*b^2*c^5*d*h^2*m - 1680*a^4*b^6*c^2*e*g*m^2 + 1068*a^
3*b^4*c^5*d^2*h*m + 960*a^5*b^2*c^5*e*h^2*l - 876*a^4*b^4*c^4*d*h^2*m - 864
*a^5*b^2*c^5*f*h^2*k + 546*a^2*b^6*c^4*d^2*h*m + 432*a^3*b^6*c^3*d*h^2*m +
336*a^4*b^3*c^5*f^2*h*k - 320*a^5*b^2*c^5*d*j^2*k + 192*a^5*b^2*c^5*g*h^2*j
 + 144*a^5*b^3*c^4*f*h*k^2 - 144*a^4*b^4*c^4*e*h^2*l - 102*a^4*b^5*c^3*f*h*
k^2 - 96*a^4*b^3*c^5*f^2*g*l - 36*a^2*b^8*c^2*d*h^2*m - 30*a^3*b^5*c^4*f^2*
h*k - 24*a^3*b^5*c^4*f^2*g*l + 16*a^4*b^4*c^4*g*h^2*j - 12*a^4*b^4*c^4*f*h^
2*k + 12*a^3*b^6*c^3*f*h^2*k + 8*a^2*b^7*c^3*f^2*g*l + 6*a^3*b^7*c^2*f*h*k^
2 - 2*a^2*b^7*c^3*f^2*h*k - 9312*a^5*b^3*c^4*d*h*l^2 + 3288*a^4*b^5*c^3*d*h
*l^2 - 2304*a^4*b^2*c^6*e^2*g*l + 1920*a^5*b^3*c^4*e*g*l^2 + 1728*a^4*b^2*c
^6*e^2*f*m + 1152*a^4*b^3*c^5*e*g^2*l - 768*a^4*b^5*c^3*e*g*l^2 - 608*a^4*b
^3*c^5*d*g^2*m - 472*a^3*b^7*c^2*d*h*l^2 + 384*a^3*b^4*c^5*e^2*g*l - 288*a^
3*b^4*c^5*e^2*f*m - 224*a^4*b^3*c^5*f*g^2*k + 192*a^5*b^2*c^5*f*h*j^2 + 192
*a^4*b^2*c^6*e^2*h*k - 192*a^3*b^5*c^4*e*g^2*l + 120*a^3*b^5*c^4*d*g^2*m +
64*a^3*b^7*c^2*e*g*l^2 - 32*a^3*b^4*c^5*e^2*h*k + 24*a^3*b^5*c^4*f*g^2*k +
9936*a^3*b^3*c^6*d^2*f*m + 3786*a^4*b^5*c^3*d*f*m^2 - 3552*a^5*b^2*c^5*d*h*
k^2 - 3486*a^2*b^5*c^5*d^2*f*m - 3424*a^3*b^3*c^6*d^2*g*l - 1868*a^3*b^7*c^
2*d*f*m^2 + 1332*a^4*b^4*c^4*d*h*k^2 - 1296*a^5*b^3*c^4*d*f*m^2 - 1236*a^3*
b^4*c^5*d*f^2*m + 1224*a^2*b^5*c^5*d^2*g*l - 1152*a^4*b^2*c^6*d*f^2*m + 960
*a^5*b^2*c^5*e*g*k^2 - 496*a^3*b^3*c^6*d^2*h*k + 462*a^2*b^6*c^4*d*f^2*m +
432*a^4*b^3*c^5*d*h^2*k - 240*a^4*b^4*c^4*e*g*k^2 - 222*a^2*b^5*c^5*d^2*h*k
 + 192*a^4*b^2*c^6*f^2*g*j + 192*a^4*b^2*c^6*e*f^2*l - 174*a^3*b^5*c^4*d*h^
2*k - 156*a^3*b^6*c^3*d*h*k^2 + 48*a^3*b^4*c^5*e*f^2*l - 32*a^4*b^3*c^5*e*h
^2*j + 16*a^3*b^6*c^3*e*g*k^2 + 16*a^3*b^4*c^5*f^2*g*j - 16*a^2*b^6*c^4*e*f
^2*l + 12*a^2*b^7*c^3*d*h^2*k + 6*a^2*b^8*c^2*d*h*k^2 + 1728*a^5*b^2*c^5*d*
f*l^2 + 1392*a^4*b^4*c^4*d*f*l^2 - 840*a^3*b^6*c^3*d*f*l^2 - 768*a^4*b^2*c^
6*e*g^2*j + 576*a^4*b^2*c^6*d*g^2*k + 480*a^3*b^3*c^6*d*e^2*m + 144*a^2*b^8
*c^2*d*f*l^2 + 96*a^4*b^3*c^5*d*h*j^2 + 96*a^3*b^3*c^6*e^2*f*k - 80*a^3*b^4
*c^5*d*g^2*k + 6848*a^3*b^2*c^7*d^2*e*l - 3552*a^3*b^2*c^7*d^2*f*k - 2448*a
^2*b^4*c^6*d^2*e*l + 1332*a^2*b^4*c^6*d^2*f*k + 960*a^3*b^2*c^7*d^2*g*j - 4
96*a^4*b^3*c^5*d*f*k^2 + 432*a^3*b^3*c^6*d*f^2*k - 240*a^2*b^4*c^6*d^2*g*j

- 222*a^3*b^5*c^4*d*f*k^2 - 174*a^2*b^5*c^5*d*f^2*k + 64*a^4*b^2*c^6*f*g^2*
h + 48*a^3*b^4*c^5*f*g^2*h + 42*a^2*b^7*c^3*d*f*k^2 - 32*a^3*b^3*c^6*e*f^2*
j - 320*a^3*b^2*c^7*d*e^2*k + 192*a^4*b^2*c^6*e*g*h^2 + 192*a^4*b^2*c^6*d*f
*j^2 - 32*a^3*b^4*c^5*d*f*j^2 + 16*a^3*b^4*c^5*e*g*h^2 + 480*a^2*b^3*c^7*d^
2*e*j - 224*a^3*b^3*c^6*d*g^2*h + 192*a^3*b^2*c^7*e^2*f*h + 24*a^2*b^5*c^5*
d*g^2*h - 864*a^3*b^2*c^7*d*f^2*h + 336*a^3*b^3*c^6*d*f*h^2 + 192*a^3*b^2*c
^7*e*f^2*g + 144*a^2*b^3*c^7*d^2*f*h - 30*a^2*b^5*c^5*d*f*h^2 + 16*a^2*b^4*
c^6*e*f^2*g - 12*a^2*b^4*c^6*d*f^2*h + 192*a^3*b^2*c^7*d*f*g^2 + 96*a^2*b^3
*c^7*d*e^2*h + 48*a^2*b^4*c^6*d*f*g^2 + 960*a^2*b^2*c^8*d^2*e*g + 192*a^2*b
^2*c^8*d*e^2*f - 7680*a^9*b*c^2*l^2*m^2 + 3152*a^8*b^3*c*l^2*m^2 + 2070*a^7
*b^4*c*k^2*m^2 - 1840*a^7*b^3*c^2*k^3*m + 6720*a^8*b*c^3*j^2*m^2 - 3072*a^8
*b*c^3*k^2*l^2 + 1680*a^6*b^5*c*j^2*m^2 - 100*a^6*b^5*c*k^2*l^2 - 2176*a^7*
b^3*c^2*j*l^3 - 256*a^6*b^3*c^3*j^3*l - 64*a^5*b^6*c*j^2*l^2 - 12480*a^8*b^
2*c^2*h*m^3 + 972*a^5*b^6*c*h^2*m^2 - 960*a^7*b*c^4*j^2*k^2 - 252*a^5*b^4*c
^3*h^3*m - 192*a^6*b^2*c^4*h^3*m + 54*a^4*b^6*c^2*h^3*m + 1536*a^7*b*c^4*h^
2*l^2 + 420*a^4*b^7*c*g^2*m^2 - 36*a^4*b^7*c*h^2*l^2 - 3072*a^7*b^2*c^3*g*l
^3 + 2096*a^7*b^3*c^2*f*m^3 + 1088*a^6*b^4*c^2*g*l^3 - 496*a^6*b^3*c^3*h*k^
3 - 192*a^4*b^4*c^4*g^3*l + 176*a^4*b^3*c^5*f^3*m + 144*a^5*b^3*c^4*h^3*k +
78*a^3*b^8*c*f^2*m^2 + 54*a^3*b^5*c^4*f^3*m + 32*a^3*b^6*c^3*g^3*l + 30*a^
5*b^5*c^2*h*k^3 - 18*a^4*b^5*c^3*h^3*k - 18*a^2*b^7*c^3*f^3*m - 16*a^3*b^8*
c*g^2*l^2 + 6720*a^6*b*c^5*e^2*m^2 - 192*a^6*b*c^5*h^2*j^2 - 4*a^2*b^9*c*f^
2*l^2 - 35040*a^7*b^2*c^3*d*m^3 + 14300*a^6*b^4*c^2*d*m^3 - 12000*a^3*b^2*c
^7*d^3*m + 4380*a^2*b^4*c^6*d^3*m - 2176*a^6*b^3*c^3*e*l^3 - 256*a^3*b^3*c^
6*e^3*l - 192*a^6*b^2*c^4*f*k^3 + 192*a^5*b^5*c^2*e*l^3 - 192*a^4*b^2*c^6*f
^3*k + 132*a^5*b^4*c^3*f*k^3 + 128*a^4*b^3*c^5*g^3*j - 28*a^3*b^4*c^5*f^3*k
- 10*a^4*b^6*c^2*f*k^3 + 6*a^2*b^6*c^4*f^3*k + 10752*a^5*b*c^6*d^2*l^2 - 9
60*a^5*b*c^6*e^2*k^2 - 192*a^5*b*c^6*f^2*j^2 + 108*a*b^9*c^2*d^2*l^2 - 1680
*a^5*b^3*c^4*d*k^3 - 1680*a^2*b^3*c^7*d^3*k + 222*a^4*b^5*c^3*d*k^3 + 30*a*
b^8*c^3*d^2*k^2 - 10*a^3*b^7*c^2*d*k^3 - 960*a^4*b*c^7*d^2*j^2 + 80*a^4*b^3
*c^5*f*h^3 + 80*a^3*b^3*c^6*f^3*h + 6*a^3*b^5*c^4*f*h^3 + 6*a^2*b^5*c^5*f^3
*h - 192*a^4*b*c^7*e^2*h^2 - 192*a^4*b^2*c^6*d*h^3 - 192*a^2*b^2*c^8*d^3*h
+ 128*a^3*b^3*c^6*e*g^3 - 28*a^3*b^4*c^5*d*h^3 + 12*a*b^6*c^5*d^2*h^2 + 6*a
^2*b^6*c^4*d*h^3 - 192*a^3*b*c^8*e^2*f^2 + 60*a*b^5*c^6*d^2*g^2 + 198*a*b^4
*c^7*d^2*f^2 + 144*a^2*b^3*c^7*d*f^3 - 960*a^2*b*c^9*d^2*e^2 + 240*a*b^3*c^
8*d^2*e^2 + 15360*a^9*c^3*k*l^2*m - 12800*a^9*c^3*j*l*m^2 - 3840*a^8*c^4*j^
2*k*m + 432*a^6*b^6*j*l*m^2 + 4608*a^8*c^4*j*k^2*l + 2880*a^8*c^4*h*k^2*m +
5120*a^8*c^4*f*l^2*m - 3072*a^8*c^4*h*k*l^2 + 270*a^5*b^7*h*k*m^2 - 216*a^
5*b^7*g*l*m^2 - 12800*a^8*c^4*e*l*m^2 - 4800*a^8*c^4*f*k*m^2 - 512*a^7*c^5*
h^2*j*l - 3840*a^6*c^6*e^2*k*m - 1280*a^7*c^5*f*j^2*m + 768*a^7*c^5*h*j^2*k
+ 144*a^4*b^8*g*j*m^2 - 90*a^4*b^8*f*k*m^2 + 8640*a^7*c^5*d*k^2*m + 4608*a
^7*c^5*e*k^2*l + 512*a^6*c^6*f^2*j*l - 9216*a^7*c^5*d*k*l^2 - 4096*a^7*c^5*
e*j*l^2 + 320*a^6*c^6*f^2*h*m - 90*a^3*b^9*d*k*m^2 + 15200*a^9*b*c^2*k*m^3
- 6192*a^8*b^3*c*k*m^3 + 5472*a^8*b*c^3*k^3*m - 4608*a^5*c^7*d^2*j*l - 1024
*a^7*c^5*f*h*l^2 + 150*a^6*b^5*c*k^3*m + 54*a^3*b^9*f*h*m^2 + 6*b^10*c^2*d^
2*h*m - 14400*a^7*c^5*d*h*m^2 + 8640*a^5*c^7*d^2*h*m + 2880*a^6*c^6*d*h^2*m

+ 2304*a^6*c^6*d*j^2*k - 512*a^6*c^6*e*h^2*l - 192*a^6*c^6*f*h^2*k + 6144*
a^8*b*c^3*j*l^3 + 1536*a^7*b*c^4*j^3*l - 1280*a^5*c^7*e^2*f*m + 768*a^5*c^7
*e^2*h*k + 256*a^6*c^6*f*h*j^2 + 192*a^6*b^5*c*j*l^3 + 54*a^2*b^10*d*h*m^2
- 18*b^9*c^3*d^2*f*m + 8*b^9*c^3*d^2*g*l - 2*b^9*c^3*d^2*h*k + 4068*a^7*b^4
*c*h*m^3 - 1728*a^6*c^6*d*h*k^2 + 960*a^5*c^7*d*f^2*m + 512*a^5*c^7*e*f^2*l
- 3072*a^6*c^6*d*f*l^2 - 16*b^8*c^4*d^2*e*l + 6*b^8*c^4*d^2*f*k - 4608*a^4
*c^8*d^2*e*l + 2400*a^8*b*c^3*f*m^3 + 2016*a^7*b*c^4*h*k^3 - 1728*a^4*c^8*d
^2*f*k - 1146*a^6*b^5*c*f*m^3 + 224*a^6*b*c^5*h^3*k - 96*a^5*b^6*c*g*l^3 +
96*a^5*b*c^6*f^3*m + 2304*a^4*c^8*d*e^2*k + 768*a^5*c^7*d*f*j^2 + 6144*a^7*
b*c^4*e*l^3 - 2280*a^5*b^6*c*d*m^3 + 1536*a^4*b*c^7*e^3*l - 616*a*b^6*c^5*d
^3*m + 512*a^6*b*c^5*g*j^3 + 256*a^4*c^8*e^2*f*h + 240*a*b^10*c*d^2*m^2 + 6
*b^7*c^5*d^2*f*h - 192*a^4*c^8*d*f^2*h + 4320*a^6*b*c^5*d*k^3 + 4320*a^3*b*
c^8*d^3*k + 222*a*b^5*c^6*d^3*k + 16*b^6*c^6*d^2*e*g + 96*a^5*b*c^6*f*h^3 +
96*a^4*b*c^7*f^3*h + 768*a^3*c^9*d*e^2*f + 512*a^3*b*c^8*e^3*g + 132*a*b^4
*c^7*d^3*h + 2016*a^2*b*c^9*d^3*f - 496*a*b^3*c^8*d^3*f + 224*a^3*b*c^8*d*f
^3 - 18*a*b^5*c^6*d*f^3 - 3264*a^8*b^2*c^2*k^2*m^2 - 6160*a^7*b^3*c^2*j^2*m
^2 + 1104*a^7*b^3*c^2*k^2*l^2 - 1920*a^7*b^2*c^3*j^2*l^2 + 768*a^6*b^4*c^2*
j^2*l^2 + 3888*a^7*b^2*c^3*h^2*m^2 - 3510*a^6*b^4*c^2*h^2*m^2 + 240*a^6*b^3
*c^3*j^2*k^2 - 16*a^5*b^5*c^2*j^2*k^2 + 1680*a^6*b^3*c^3*g^2*m^2 - 1648*a^6
*b^3*c^3*h^2*l^2 - 1540*a^5*b^5*c^2*g^2*m^2 + 444*a^5*b^5*c^2*h^2*l^2 - 960
*a^6*b^2*c^4*h^2*k^2 - 576*a^6*b^2*c^4*f^2*m^2 - 512*a^6*b^2*c^4*g^2*l^2 -
480*a^5*b^4*c^3*g^2*l^2 + 198*a^5*b^4*c^3*h^2*k^2 + 192*a^4*b^6*c^2*g^2*l^2
- 186*a^5*b^4*c^3*f^2*m^2 - 97*a^4*b^6*c^2*f^2*m^2 - 9*a^4*b^6*c^2*h^2*k^2
- 6160*a^5*b^3*c^4*e^2*m^2 + 1680*a^4*b^5*c^3*e^2*m^2 - 240*a^5*b^3*c^4*g^
2*k^2 - 240*a^5*b^3*c^4*f^2*l^2 - 144*a^3*b^7*c^2*e^2*m^2 + 60*a^4*b^5*c^3*
g^2*k^2 - 36*a^4*b^5*c^3*f^2*l^2 + 36*a^3*b^7*c^2*f^2*l^2 - 16*a^5*b^3*c^4*
h^2*j^2 - 4*a^3*b^7*c^2*g^2*k^2 + 38512*a^5*b^2*c^5*d^2*m^2 - 32310*a^4*b^4
*c^4*d^2*m^2 + 12720*a^3*b^6*c^3*d^2*m^2 - 2500*a^2*b^8*c^2*d^2*m^2 - 1920*
a^5*b^2*c^5*e^2*l^2 + 768*a^4*b^4*c^4*e^2*l^2 - 464*a^5*b^2*c^5*f^2*k^2 - 3
84*a^5*b^2*c^5*g^2*j^2 - 64*a^3*b^6*c^3*e^2*l^2 + 42*a^4*b^4*c^4*f^2*k^2 +
12*a^3*b^6*c^3*f^2*k^2 - 13104*a^4*b^3*c^5*d^2*l^2 + 5628*a^3*b^5*c^4*d^2*l
^2 - 1128*a^2*b^7*c^3*d^2*l^2 + 240*a^4*b^3*c^5*e^2*k^2 - 16*a^4*b^3*c^5*f^
2*j^2 - 16*a^3*b^5*c^4*e^2*k^2 - 2880*a^4*b^2*c^6*d^2*k^2 + 1750*a^3*b^4*c^
5*d^2*k^2 - 345*a^2*b^6*c^4*d^2*k^2 - 48*a^4*b^3*c^5*g^2*h^2 - 4*a^3*b^5*c^
4*g^2*h^2 + 240*a^3*b^3*c^6*d^2*j^2 - 192*a^4*b^2*c^6*f^2*h^2 - 42*a^3*b^4*
c^5*f^2*h^2 - 16*a^2*b^5*c^5*d^2*j^2 - 48*a^3*b^3*c^6*f^2*g^2 - 16*a^3*b^3*
c^6*e^2*h^2 - 4*a^2*b^5*c^5*f^2*g^2 - 464*a^3*b^2*c^7*d^2*h^2 - 384*a^3*b^2
*c^7*e^2*g^2 + 42*a^2*b^4*c^6*d^2*h^2 - 240*a^2*b^3*c^7*d^2*g^2 - 16*a^2*b^
3*c^7*e^2*f^2 - 960*a^2*b^2*c^8*d^2*f^2 + 6*b^11*c*d^2*k*m - 18*a*b^11*d*f*
m^2 - 7200*a^9*c^3*k^2*m^2 - 324*a^7*b^5*l^2*m^2 - 225*a^6*b^6*k^2*m^2 - 20
48*a^8*c^4*j^2*l^2 - 144*a^5*b^7*j^2*m^2 - 2400*a^8*c^4*h^2*m^2 - 81*a^4*b^
8*h^2*m^2 - 800*a^7*c^5*f^2*m^2 - 288*a^7*c^5*h^2*k^2 - 36*a^3*b^9*g^2*m^2
- 9*a^2*b^10*f^2*m^2 - 21600*a^6*c^6*d^2*m^2 - 2048*a^6*c^6*e^2*l^2 - 864*a
^6*c^6*f^2*k^2 - 2592*a^5*c^7*d^2*k^2 - 1536*a^5*c^7*e^2*j^2 + 1536*a^8*b^2
*c^2*l^4 - 32*a^5*c^7*f^2*h^2 + 360*a^7*b^2*c^3*k^4 - 25*a^6*b^4*c^2*k^4 -

864*a^4*c^8*d^2*h^2 - 4*b^7*c^5*d^2*g^2 - 9*b^6*c^6*d^2*f^2 - 288*a^3*c^9*d
^2*f^2 - 24*a^5*b^2*c^5*h^4 - 16*b^5*c^7*d^2*e^2 - 9*a^4*b^4*c^4*h^4 - 16*a
^3*b^4*c^5*g^4 - 24*a^3*b^2*c^7*f^4 - 9*a^2*b^4*c^6*f^4 - a^2*b^8*c^2*f^2*k
^2 - a^2*b^6*c^4*f^2*h^2 + 630*a^7*b^5*k*m^3 + 8000*a^9*c^3*h*m^3 + 320*a^7
*c^5*h^3*m - 378*a^6*b^6*h*m^3 + 126*a^5*b^7*f*m^3 + 30*b^8*c^4*d^3*m + 240
00*a^8*c^4*d*m^3 + 8640*a^4*c^8*d^3*m - 1728*a^7*c^5*f*k^3 - 192*a^5*c^7*f^
3*k - 4*b^11*c*d^2*l^2 + 126*a^4*b^8*d*m^3 - 10*b^7*c^5*d^3*k + 4200*a^9*b^
2*c*m^4 - 1024*a^6*c^6*e*j^3 - 1024*a^4*c^8*e^3*j - 144*a^7*b^4*c*l^4 - 10*
b^6*c^6*d^3*h - 1728*a^3*c^9*d^3*h - 192*a^5*c^7*d*h^3 + 30*b^5*c^7*d^3*f +
 360*a*b^2*c^9*d^4 - 9*b^12*d^2*m^2 - 10000*a^10*c^2*m^4 - 4096*a^9*c^3*l^4
 - 441*a^8*b^4*m^4 - 1296*a^8*c^4*k^4 - 256*a^7*c^5*j^4 - 16*a^6*c^6*h^4 -
16*a^4*c^8*f^4 - 256*a^3*c^9*e^4 - 25*b^4*c^8*d^4 - 1296*a^2*c^10*d^4 - b^1
0*c^2*d^2*k^2 - b^8*c^4*d^2*h^2, z, k1)*((3072*a^5*c^7*d*l - 512*a^4*c^8*e*
f - 1536*a^5*c^7*e*k - 512*a^5*c^7*f*j + 1024*a^6*c^6*h*l - 1536*a^6*c^6*j*
k - 5120*a^7*c^5*l*m + 32*a*b^5*c^6*d*e + 1024*a^3*b*c^8*d*e - 16*a*b^6*c^5
*d*g + 512*a^4*b*c^7*e*h + 256*a^4*b*c^7*f*g + 1024*a^4*b*c^7*d*j + 16*a*b^
8*c^3*d*l + 2048*a^5*b*c^6*e*m + 256*a^5*b*c^6*f*l + 768*a^5*b*c^6*g*k + 51
2*a^5*b*c^6*h*j + 2048*a^6*b*c^5*j*m + 1792*a^6*b*c^5*k*l - 384*a^2*b^3*c^7
*d*e + 192*a^2*b^4*c^6*d*g + 32*a^2*b^4*c^6*e*f - 512*a^3*b^2*c^7*d*g - 16*
a^2*b^5*c^5*f*g - 128*a^3*b^3*c^6*e*h + 32*a^2*b^5*c^5*d*j - 384*a^3*b^3*c^
6*d*j + 64*a^3*b^4*c^5*g*h - 256*a^4*b^2*c^6*g*h - 288*a^2*b^6*c^4*d*l + 17
92*a^3*b^4*c^5*d*l - 32*a^3*b^4*c^5*e*k + 32*a^3*b^4*c^5*f*j - 4352*a^4*b^2
*c^6*d*l + 512*a^4*b^2*c^6*e*k + 16*a^2*b^7*c^3*f*l + 96*a^3*b^5*c^4*e*m -
144*a^3*b^5*c^4*f*l + 16*a^3*b^5*c^4*g*k - 896*a^4*b^3*c^5*e*m + 256*a^4*b^
3*c^5*f*l - 256*a^4*b^3*c^5*g*k - 128*a^4*b^3*c^5*h*j - 48*a^3*b^6*c^3*g*m
- 48*a^3*b^6*c^3*h*l + 448*a^4*b^4*c^4*g*m + 512*a^4*b^4*c^4*h*l - 1024*a^5
*b^2*c^5*g*m - 1536*a^5*b^2*c^5*h*l - 32*a^4*b^4*c^4*j*k + 512*a^5*b^2*c^5*
j*k + 96*a^4*b^5*c^3*j*m + 80*a^4*b^5*c^3*k*l - 896*a^5*b^3*c^4*j*m - 768*a
^5*b^3*c^4*k*l - 256*a^5*b^4*c^3*l*m + 2304*a^6*b^2*c^4*l*m)/(8*(64*a^5*c^6
 - a^2*b^6*c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5)) - root(1572864*a^8*b^2*c
^10*z^4 - 983040*a^7*b^4*c^9*z^4 + 327680*a^6*b^6*c^8*z^4 - 61440*a^5*b^8*c
^7*z^4 + 6144*a^4*b^10*c^6*z^4 - 256*a^3*b^12*c^5*z^4 - 1048576*a^9*c^11*z^
4 - 1572864*a^8*b^2*c^8*l*z^3 + 983040*a^7*b^4*c^7*l*z^3 - 327680*a^6*b^6*c
^6*l*z^3 + 61440*a^5*b^8*c^5*l*z^3 - 6144*a^4*b^10*c^4*l*z^3 + 256*a^3*b^12
*c^3*l*z^3 + 1048576*a^9*c^9*l*z^3 + 96*a^3*b^12*c*k*m*z^2 + 98304*a^8*b*c^
7*j*l*z^2 + 24576*a^8*b*c^7*h*m*z^2 + 155648*a^7*b*c^8*d*m*z^2 + 98304*a^7*
b*c^8*e*l*z^2 + 57344*a^7*b*c^8*f*k*z^2 + 32768*a^7*b*c^8*g*j*z^2 + 57344*a
^6*b*c^9*d*h*z^2 + 32768*a^6*b*c^9*e*g*z^2 - 32*a*b^10*c^5*d*f*z^2 - 491520
*a^8*b^2*c^6*k*m*z^2 + 358400*a^7*b^4*c^5*k*m*z^2 - 129024*a^6*b^6*c^4*k*m*
z^2 + 24768*a^5*b^8*c^3*k*m*z^2 - 2432*a^4*b^10*c^2*k*m*z^2 - 90112*a^7*b^3
*c^6*j*l*z^2 + 30720*a^6*b^5*c^5*j*l*z^2 - 4608*a^5*b^7*c^4*j*l*z^2 + 256*a
^4*b^9*c^3*j*l*z^2 - 21504*a^6*b^5*c^5*h*m*z^2 + 9216*a^5*b^7*c^4*h*m*z^2 +
 8192*a^7*b^3*c^6*h*m*z^2 - 1568*a^4*b^9*c^3*h*m*z^2 + 96*a^3*b^11*c^2*h*m*
z^2 - 172032*a^7*b^2*c^7*f*m*z^2 + 116736*a^6*b^4*c^6*f*m*z^2 - 49152*a^7*b
^2*c^7*g*l*z^2 + 45056*a^6*b^4*c^6*g*l*z^2 - 35840*a^5*b^6*c^5*f*m*z^2 + 24

576*a^7*b^2*c^7*h*k*z^2 - 15360*a^5*b^6*c^5*g*l*z^2 + 5184*a^4*b^8*c^4*f*m*
z^2 - 3072*a^5*b^6*c^5*h*k*z^2 + 2304*a^4*b^8*c^4*g*l*z^2 + 2048*a^6*b^4*c^
6*h*k*z^2 + 576*a^4*b^8*c^4*h*k*z^2 - 288*a^3*b^10*c^3*f*m*z^2 - 128*a^3*b^
10*c^3*g*l*z^2 - 32*a^3*b^10*c^3*h*k*z^2 - 147456*a^6*b^3*c^7*d*m*z^2 - 901
12*a^6*b^3*c^7*e*l*z^2 + 52224*a^5*b^5*c^6*d*m*z^2 - 49152*a^6*b^3*c^7*f*k*
z^2 + 30720*a^5*b^5*c^6*e*l*z^2 - 24576*a^6*b^3*c^7*g*j*z^2 + 15360*a^5*b^5
*c^6*f*k*z^2 - 8192*a^4*b^7*c^5*d*m*z^2 + 6144*a^5*b^5*c^6*g*j*z^2 - 4608*a
^4*b^7*c^5*e*l*z^2 - 2048*a^4*b^7*c^5*f*k*z^2 - 512*a^4*b^7*c^5*g*j*z^2 + 4
80*a^3*b^9*c^4*d*m*z^2 + 256*a^3*b^9*c^4*e*l*z^2 + 96*a^3*b^9*c^4*f*k*z^2 +
 131072*a^6*b^2*c^8*d*k*z^2 + 49152*a^6*b^2*c^8*e*j*z^2 - 43008*a^5*b^4*c^7
*d*k*z^2 - 12288*a^5*b^4*c^7*e*j*z^2 + 6144*a^4*b^6*c^6*d*k*z^2 + 1024*a^4*
b^6*c^6*e*j*z^2 - 320*a^3*b^8*c^5*d*k*z^2 + 6144*a^5*b^4*c^7*f*h*z^2 - 2048
*a^4*b^6*c^6*f*h*z^2 + 192*a^3*b^8*c^5*f*h*z^2 - 49152*a^5*b^3*c^8*d*h*z^2
- 24576*a^5*b^3*c^8*e*g*z^2 + 15360*a^4*b^5*c^7*d*h*z^2 + 6144*a^4*b^5*c^7*
e*g*z^2 - 2048*a^3*b^7*c^6*d*h*z^2 - 512*a^3*b^7*c^6*e*g*z^2 + 96*a^2*b^9*c
^5*d*h*z^2 + 24576*a^5*b^2*c^9*d*f*z^2 - 3072*a^3*b^6*c^7*d*f*z^2 + 2048*a^
4*b^4*c^8*d*f*z^2 + 576*a^2*b^8*c^6*d*f*z^2 - 430080*a^9*b*c^6*m^2*z^2 + 34
08*a^4*b^11*c*m^2*z^2 - 64*a^3*b^12*c*l^2*z^2 + 61440*a^8*b*c^7*k^2*z^2 + 1
2288*a^7*b*c^8*h^2*z^2 + 12288*a^6*b*c^9*f^2*z^2 + 61440*a^5*b*c^10*d^2*z^2
 + 432*a*b^9*c^6*d^2*z^2 + 245760*a^9*c^7*k*m*z^2 + 81920*a^8*c^8*f*m*z^2 -
 49152*a^8*c^8*h*k*z^2 - 147456*a^7*c^9*d*k*z^2 - 65536*a^7*c^9*e*j*z^2 - 1
6384*a^7*c^9*f*h*z^2 - 49152*a^6*c^10*d*f*z^2 + 716800*a^8*b^3*c^5*m^2*z^2
- 483840*a^7*b^5*c^4*m^2*z^2 + 170496*a^6*b^7*c^3*m^2*z^2 - 33232*a^5*b^9*c
^2*m^2*z^2 + 516096*a^8*b^2*c^6*l^2*z^2 - 288768*a^7*b^4*c^5*l^2*z^2 + 8857
6*a^6*b^6*c^4*l^2*z^2 - 15744*a^5*b^8*c^3*l^2*z^2 + 1536*a^4*b^10*c^2*l^2*z
^2 - 61440*a^7*b^3*c^6*k^2*z^2 + 24064*a^6*b^5*c^5*k^2*z^2 - 4608*a^5*b^7*c
^4*k^2*z^2 + 432*a^4*b^9*c^3*k^2*z^2 - 16*a^3*b^11*c^2*k^2*z^2 + 24576*a^7*
b^2*c^7*j^2*z^2 - 6144*a^6*b^4*c^6*j^2*z^2 + 512*a^5*b^6*c^5*j^2*z^2 - 8192
*a^6*b^3*c^7*h^2*z^2 + 1536*a^5*b^5*c^6*h^2*z^2 - 16*a^3*b^9*c^4*h^2*z^2 -
8192*a^6*b^2*c^8*g^2*z^2 + 6144*a^5*b^4*c^7*g^2*z^2 - 1536*a^4*b^6*c^6*g^2*
z^2 + 128*a^3*b^8*c^5*g^2*z^2 - 8192*a^5*b^3*c^8*f^2*z^2 + 1536*a^4*b^5*c^7
*f^2*z^2 - 16*a^2*b^9*c^5*f^2*z^2 + 24576*a^5*b^2*c^9*e^2*z^2 - 6144*a^4*b^
4*c^8*e^2*z^2 + 512*a^3*b^6*c^7*e^2*z^2 - 61440*a^4*b^3*c^9*d^2*z^2 + 24064
*a^3*b^5*c^8*d^2*z^2 - 4608*a^2*b^7*c^7*d^2*z^2 - 393216*a^9*c^7*l^2*z^2 -
144*a^3*b^13*m^2*z^2 - 32768*a^8*c^8*j^2*z^2 - 32768*a^6*c^10*e^2*z^2 - 16*
b^11*c^5*d^2*z^2 + 18432*a^8*b*c^5*h*l*m*z - 96*a^3*b^10*c*g*k*m*z + 90112*
a^7*b*c^6*e*k*m*z + 36864*a^7*b*c^6*f*j*m*z - 16384*a^7*b*c^6*g*j*l*z + 143
36*a^7*b*c^6*d*l*m*z - 10240*a^7*b*c^6*f*k*l*z + 4096*a^7*b*c^6*h*j*k*z + 1
0240*a^7*b*c^6*g*h*m*z - 47104*a^6*b*c^7*d*h*l*z + 36864*a^6*b*c^7*e*f*m*z
+ 30720*a^6*b*c^7*d*g*m*z - 16384*a^6*b*c^7*e*g*l*z + 6144*a^6*b*c^7*f*g*k*
z + 4096*a^6*b*c^7*e*h*k*z + 32*a*b^10*c^3*d*f*l*z - 4096*a^5*b*c^8*d*f*j*z
 - 6144*a^5*b*c^8*d*g*h*z - 32*a*b^8*c^5*d*f*g*z - 4096*a^4*b*c^9*d*e*f*z +
 64*a*b^7*c^6*d*e*f*z + 110592*a^8*b^2*c^4*k*l*m*z - 36864*a^7*b^4*c^3*k*l*
m*z + 5376*a^6*b^6*c^2*k*l*m*z - 79872*a^7*b^3*c^4*j*k*m*z + 26112*a^6*b^5*
c^3*j*k*m*z - 3712*a^5*b^7*c^2*j*k*m*z - 13824*a^7*b^3*c^4*h*l*m*z + 3456*a

^6*b^5*c^3*h*l*m*z - 288*a^5*b^7*c^2*h*l*m*z - 45056*a^7*b^2*c^5*g*k*m*z + 39936*a^6*b^4*c^4*g*k*m*z + 30720*a^7*b^2*c^5*f*l*m*z - 18432*a^7*b^2*c^5*h*k*l*z - 13056*a^5*b^6*c^3*g*k*m*z - 7680*a^6*b^4*c^4*f*l*m*z + 5376*a^6*b^4*c^4*h*j*m*z + 4608*a^6*b^4*c^4*h*k*l*z + 3072*a^7*b^2*c^5*h*j*m*z - 1984*a^5*b^6*c^3*h*j*m*z + 1856*a^4*b^8*c^2*g*k*m*z + 640*a^5*b^6*c^3*f*l*m*z - 384*a^5*b^6*c^3*h*k*l*z + 192*a^4*b^8*c^2*h*j*m*z - 79872*a^6*b^3*c^5*e*k*m*z - 27648*a^6*b^3*c^5*f*j*m*z + 26112*a^5*b^5*c^4*e*k*m*z + 12288*a^6*b^3*c^5*g*j*l*z - 10752*a^6*b^3*c^5*d*l*m*z + 7680*a^6*b^3*c^5*f*k*l*z + 6912*a^5*b^5*c^4*f*j*m*z - 3712*a^4*b^7*c^3*e*k*m*z - 3072*a^6*b^3*c^5*h*j*k*z - 3072*a^5*b^5*c^4*g*j*l*z + 2688*a^5*b^5*c^4*d*l*m*z - 1920*a^5*b^5*c^4*f*k*l*z + 768*a^5*b^5*c^4*h*j*k*z - 576*a^4*b^7*c^3*f*j*m*z + 256*a^4*b^7*c^3*g*j*l*z - 224*a^4*b^7*c^3*d*l*m*z + 192*a^3*b^9*c^2*e*k*m*z + 160*a^4*b^7*c^3*f*k*l*z - 64*a^4*b^7*c^3*h*j*k*z - 2688*a^5*b^5*c^4*g*h*m*z - 1536*a^6*b^3*c^5*g*h*m*z + 992*a^4*b^7*c^3*g*h*m*z - 96*a^3*b^9*c^2*g*h*m*z - 65536*a^6*b^2*c^6*d*k*l*z + 46080*a^6*b^2*c^6*d*j*m*z - 24576*a^6*b^2*c^6*e*j*l*z + 21504*a^5*b^4*c^5*d*k*l*z - 11520*a^5*b^4*c^5*d*j*m*z + 9216*a^6*b^2*c^6*f*j*k*z + 6144*a^5*b^4*c^5*e*j*l*z - 3072*a^4*b^6*c^4*d*k*l*z - 2304*a^5*b^4*c^5*f*j*k*z + 960*a^4*b^6*c^4*d*j*m*z - 512*a^4*b^6*c^4*e*j*l*z + 192*a^4*b^6*c^4*f*j*k*z + 160*a^3*b^8*c^3*d*k*l*z - 18432*a^6*b^2*c^6*f*g*m*z + 13824*a^5*b^4*c^5*f*g*m*z + 5376*a^5*b^4*c^5*e*h*m*z - 3456*a^4*b^6*c^4*f*g*m*z + 3072*a^6*b^2*c^6*e*h*m*z - 3072*a^5*b^4*c^5*f*h*l*z - 2048*a^6*b^2*c^6*g*h*k*z - 1984*a^4*b^6*c^4*e*h*m*z + 1536*a^5*b^4*c^5*g*h*k*z + 1024*a^4*b^6*c^4*f*h*l*z - 384*a^4*b^6*c^4*g*h*k*z + 288*a^3*b^8*c^3*f*g*m*z + 192*a^3*b^8*c^3*e*h*m*z - 96*a^3*b^8*c^3*f*h*l*z + 32*a^3*b^8*c^3*g*h*k*z + 41472*a^5*b^3*c^6*d*h*l*z - 27648*a^5*b^3*c^6*e*f*m*z - 23040*a^5*b^3*c^6*d*g*m*z - 13440*a^4*b^5*c^5*d*h*l*z + 12288*a^5*b^3*c^6*e*g*l*z + 6912*a^4*b^5*c^5*e*f*m*z + 5760*a^4*b^5*c^5*d*g*m*z - 4608*a^5*b^3*c^6*f*g*k*z - 3072*a^5*b^3*c^6*e*h*k*z - 3072*a^4*b^5*c^5*e*g*l*z + 1888*a^3*b^7*c^4*d*h*l*z + 1152*a^4*b^5*c^5*f*g*k*z + 768*a^4*b^5*c^5*e*h*k*z - 576*a^3*b^7*c^4*e*f*m*z - 480*a^3*b^7*c^4*d*g*m*z + 256*a^3*b^7*c^4*e*g*l*z - 96*a^3*b^7*c^4*f*g*k*z - 96*a^2*b^9*c^3*d*h*l*z - 64*a^3*b^7*c^4*e*h*k*z + 46080*a^5*b^2*c^7*d*e*m*z - 11520*a^4*b^4*c^6*d*e*m*z + 9216*a^5*b^2*c^7*e*f*k*z - 9216*a^5*b^2*c^7*d*h*j*z - 6656*a^4*b^4*c^6*d*f*l*z - 6144*a^5*b^2*c^7*d*f*l*z + 3456*a^3*b^6*c^5*d*f*l*z - 2304*a^4*b^4*c^6*e*f*k*z + 2304*a^4*b^4*c^6*d*h*j*z + 960*a^3*b^6*c^5*d*e*m*z - 576*a^2*b^8*c^4*d*f*l*z + 192*a^3*b^6*c^5*e*f*k*z - 192*a^3*b^6*c^5*d*h*j*z + 3072*a^4*b^3*c^7*d*f*j*z - 768*a^3*b^5*c^6*d*f*j*z + 64*a^2*b^7*c^5*d*f*j*z + 4608*a^4*b^3*c^7*d*g*h*z - 1152*a^3*b^5*c^6*d*g*h*z + 96*a^2*b^7*c^5*d*g*h*z - 9216*a^4*b^2*c^8*d*e*h*z + 2304*a^3*b^4*c^7*d*e*h*z + 2048*a^4*b^2*c^8*d*f*g*z - 1536*a^3*b^4*c^7*d*f*g*z + 384*a^2*b^6*c^6*d*f*g*z - 192*a^2*b^6*c^6*d*e*h*z + 3072*a^3*b^3*c^8*d*e*f*z - 768*a^2*b^5*c^7*d*e*f*z - 288*a^5*b^8*c*k*l*m*z + 90112*a^8*b*c^5*j*k*m*z + 192*a^4*b^9*c*j*k*m*z + 138240*a^9*b*c^4*l*m^2*z - 7344*a^6*b^7*c*l*m^2*z + 5088*a^5*b^8*c*j*m^2*z - 3072*a^8*b*c^5*k^2*l*z - 49152*a^8*b*c^5*j*l^2*z - 128*a^4*b^9*c*j*l^2*z - 25600*a^8*b*c^5*g*m^2*z - 9216*a^7*b*c^6*h^2*l*z - 2544*a^4*b^9*c*g*m^2*z + 64*a^3*b^10*c*g*l^2*z + 9216*a^7*b*c^6*g*k^2*z - 307

2*a^6*b*c^7*f^2*l*z - 288*a^3*b^10*c*e*m^2*z - 49152*a^7*b*c^6*e*l^2*z - 58
368*a^5*b*c^8*d^2*l*z - 432*a*b^9*c^4*d^2*l*z - 1024*a^6*b*c^7*g*h^2*z + 32
*a*b^8*c^5*d^2*j*z + 1024*a^5*b*c^8*f^2*g*z - 9216*a^4*b*c^9*d^2*g*z + 336*
a*b^7*c^6*d^2*g*z - 672*a*b^6*c^7*d^2*e*z - 122880*a^9*c^5*k*l*m*z - 40960*
a^8*c^6*f*l*m*z + 24576*a^8*c^6*h*k*l*z - 20480*a^8*c^6*h*j*m*z + 73728*a^7
*c^7*d*k*l*z - 61440*a^7*c^7*d*j*m*z + 32768*a^7*c^7*e*j*l*z - 12288*a^7*c^
7*f*j*k*z - 20480*a^7*c^7*e*h*m*z + 8192*a^7*c^7*f*h*l*z - 61440*a^6*c^8*d*
e*m*z + 24576*a^6*c^8*d*f*l*z - 12288*a^6*c^8*e*f*k*z + 12288*a^6*c^8*d*h*j
*z + 12288*a^5*c^9*d*e*h*z - 131328*a^8*b^3*c^3*l*m^2*z + 46656*a^7*b^5*c^2
*l*m^2*z - 142848*a^8*b^2*c^4*j*m^2*z + 106368*a^7*b^4*c^3*j*m^2*z - 34208*
a^6*b^6*c^2*j*m^2*z + 2304*a^7*b^3*c^4*k^2*l*z - 576*a^6*b^5*c^3*k^2*l*z +
48*a^5*b^7*c^2*k^2*l*z + 45056*a^7*b^3*c^4*j*l^2*z - 15360*a^6*b^5*c^3*j*l^
2*z - 12288*a^7*b^2*c^5*j^2*l*z + 3072*a^6*b^4*c^4*j^2*l*z + 2304*a^5*b^7*c
^2*j*l^2*z - 256*a^5*b^6*c^3*j^2*l*z + 15872*a^7*b^2*c^5*j*k^2*z - 4992*a^6
*b^4*c^4*j*k^2*z + 672*a^5*b^6*c^3*j*k^2*z - 32*a^4*b^8*c^2*j*k^2*z + 71424
*a^7*b^3*c^4*g*m^2*z - 53184*a^6*b^5*c^3*g*m^2*z + 17104*a^5*b^7*c^2*g*m^2*
z + 6912*a^6*b^3*c^5*h^2*l*z - 1728*a^5*b^5*c^4*h^2*l*z + 144*a^4*b^7*c^3*h
^2*l*z + 24576*a^7*b^2*c^5*g*l^2*z - 22528*a^6*b^4*c^4*g*l^2*z + 7680*a^5*b
^6*c^3*g*l^2*z + 4096*a^6*b^2*c^6*g^2*l*z - 3072*a^5*b^4*c^5*g^2*l*z - 1152
*a^4*b^8*c^2*g*l^2*z + 768*a^4*b^6*c^4*g^2*l*z - 64*a^3*b^8*c^3*g^2*l*z - 1
42848*a^7*b^2*c^5*e*m^2*z + 106368*a^6*b^4*c^4*e*m^2*z - 34208*a^5*b^6*c^3*
e*m^2*z - 7936*a^6*b^3*c^5*g*k^2*z + 5088*a^4*b^8*c^2*e*m^2*z + 2496*a^5*b^
5*c^4*g*k^2*z - 1536*a^6*b^2*c^6*h^2*j*z + 1280*a^5*b^3*c^6*f^2*l*z + 384*a
^5*b^4*c^5*h^2*j*z - 336*a^4*b^7*c^3*g*k^2*z + 192*a^4*b^5*c^5*f^2*l*z - 14
4*a^3*b^7*c^4*f^2*l*z - 32*a^4*b^6*c^4*h^2*j*z + 16*a^3*b^9*c^2*g*k^2*z + 1
6*a^2*b^9*c^3*f^2*l*z + 45056*a^6*b^3*c^5*e*l^2*z - 15360*a^5*b^5*c^4*e*l^2
*z - 12288*a^5*b^2*c^7*e^2*l*z + 3072*a^4*b^4*c^6*e^2*l*z + 2304*a^4*b^7*c^
3*e*l^2*z - 256*a^3*b^6*c^5*e^2*l*z - 128*a^3*b^9*c^2*e*l^2*z + 59136*a^4*b
^3*c^7*d^2*l*z - 23488*a^3*b^5*c^6*d^2*l*z + 15872*a^6*b^2*c^6*e*k^2*z - 49
92*a^5*b^4*c^5*e*k^2*z + 4560*a^2*b^7*c^5*d^2*l*z + 1536*a^5*b^2*c^7*f^2*j*
z + 672*a^4*b^6*c^4*e*k^2*z - 384*a^4*b^4*c^6*f^2*j*z - 32*a^3*b^8*c^3*e*k^
2*z + 32*a^3*b^6*c^5*f^2*j*z + 768*a^5*b^3*c^6*g*h^2*z - 192*a^4*b^5*c^5*g*
h^2*z + 16*a^3*b^7*c^4*g*h^2*z - 15872*a^4*b^2*c^8*d^2*j*z + 4992*a^3*b^4*c
^7*d^2*j*z - 672*a^2*b^6*c^6*d^2*j*z - 1536*a^5*b^2*c^7*e*h^2*z - 768*a^4*b
^3*c^7*f^2*g*z + 384*a^4*b^4*c^6*e*h^2*z + 192*a^3*b^5*c^6*f^2*g*z - 32*a^3
*b^6*c^5*e*h^2*z - 16*a^2*b^7*c^5*f^2*g*z + 7936*a^3*b^3*c^8*d^2*g*z - 2496
*a^2*b^5*c^7*d^2*g*z + 1536*a^4*b^2*c^8*e*f^2*z - 384*a^3*b^4*c^7*e*f^2*z +
32*a^2*b^6*c^6*e*f^2*z - 15872*a^3*b^2*c^9*d^2*e*z + 4992*a^2*b^4*c^8*d^2*
e*z - 61440*a^8*b^2*c^4*l^3*z + 21504*a^7*b^4*c^3*l^3*z - 3328*a^6*b^6*c^2*
l^3*z + 432*a^5*b^9*l*m^2*z + 51200*a^9*c^5*j*m^2*z + 16384*a^8*c^6*j^2*l*z
- 288*a^4*b^10*j*m^2*z - 18432*a^8*c^6*j*k^2*z + 144*a^3*b^11*g*m^2*z + 51
200*a^8*c^6*e*m^2*z + 2048*a^7*c^7*h^2*j*z + 16384*a^6*c^8*e^2*l*z + 16*b^1
1*c^3*d^2*l*z - 18432*a^7*c^7*e*k^2*z - 2048*a^6*c^8*f^2*j*z + 18432*a^5*c^
9*d^2*j*z + 192*a^5*b^8*c*l^3*z + 2048*a^6*c^8*e*h^2*z - 16*b^9*c^5*d^2*g*z
- 2048*a^5*c^9*e*f^2*z + 32*b^8*c^6*d^2*e*z + 18432*a^4*c^10*d^2*e*z + 655

36*a^9*c^5*l^3*z - 11008*a^8*b*c^3*j*k*l*m - 288*a^6*b^5*c*j*k*l*m + 144*a^
5*b^6*c*g*k*l*m - 11008*a^7*b*c^4*e*k*l*m - 5376*a^7*b*c^4*f*j*l*m + 3840*a
^7*b*c^4*g*j*k*m - 3328*a^7*b*c^4*h*j*k*l - 96*a^4*b^7*c*g*j*k*m - 2560*a^7
*b*c^4*g*h*l*m - 36*a^3*b^8*c*f*h*k*m - 6912*a^6*b*c^5*d*j*k*l - 7872*a^6*b
*c^5*d*h*k*m - 7680*a^6*b*c^5*d*g*l*m - 5376*a^6*b*c^5*e*f*l*m + 3840*a^6*b
*c^5*e*g*k*m - 3328*a^6*b*c^5*e*h*k*l - 1536*a^6*b*c^5*f*g*k*l + 1280*a^6*b
*c^5*f*g*j*m - 768*a^6*b*c^5*g*h*j*k - 768*a^6*b*c^5*f*h*j*l - 768*a^6*b*c^
5*e*h*j*m - 36*a^2*b^9*c*d*h*k*m - 6912*a^5*b*c^6*d*e*k*l - 4864*a^5*b*c^6*
d*e*j*m - 2304*a^5*b*c^6*d*g*j*k - 1792*a^5*b*c^6*e*f*j*k - 1280*a^5*b*c^6*
d*f*j*l - 4544*a^5*b*c^6*d*f*h*m + 1536*a^5*b*c^6*d*g*h*l + 1280*a^5*b*c^6*
e*f*g*m - 768*a^5*b*c^6*e*g*h*k - 768*a^5*b*c^6*e*f*h*l - 256*a^5*b*c^6*f*g
*h*j + 12*a*b^9*c^2*d*f*h*m + 16*a*b^8*c^3*d*f*g*l - 4*a*b^8*c^3*d*f*h*k -
2304*a^4*b*c^7*d*e*g*k - 1792*a^4*b*c^7*d*e*h*j - 1280*a^4*b*c^7*d*e*f*l -
768*a^4*b*c^7*d*f*g*j - 32*a*b^7*c^4*d*e*f*l - 256*a^4*b*c^7*e*f*g*h - 768*
a^3*b*c^8*d*e*f*g + 32*a*b^5*c^6*d*e*f*g + 12*a*b^10*c*d*f*k*m + 3648*a^7*b
^3*c^2*j*k*l*m + 5504*a^7*b^2*c^3*g*k*l*m - 1824*a^6*b^4*c^2*g*k*l*m + 384*
a^7*b^2*c^3*h*j*l*m - 288*a^6*b^4*c^2*h*j*l*m - 4800*a^6*b^3*c^3*g*j*k*m +
3648*a^6*b^3*c^3*e*k*l*m + 1280*a^5*b^5*c^2*g*j*k*m + 1088*a^6*b^3*c^3*f*j*
l*m + 576*a^6*b^3*c^3*h*j*k*l - 288*a^5*b^5*c^2*e*k*l*m - 192*a^6*b^3*c^3*g
*h*l*m + 144*a^5*b^5*c^2*g*h*l*m + 9600*a^6*b^2*c^4*e*j*k*m - 4224*a^6*b^2*
c^4*d*j*l*m - 2560*a^5*b^4*c^3*e*j*k*m + 384*a^6*b^2*c^4*f*j*k*l + 224*a^5*
b^4*c^3*d*j*l*m + 192*a^4*b^6*c^2*e*j*k*m - 160*a^5*b^4*c^3*f*j*k*l - 4608*
a^6*b^2*c^4*f*h*k*m + 2688*a^6*b^2*c^4*f*g*l*m + 1664*a^6*b^2*c^4*g*h*k*l -
744*a^5*b^4*c^3*f*h*k*m - 544*a^5*b^4*c^3*f*g*l*m + 492*a^4*b^6*c^2*f*h*k*
m + 416*a^5*b^4*c^3*g*h*j*m + 384*a^6*b^2*c^4*g*h*j*m + 384*a^6*b^2*c^4*e*h
*l*m - 288*a^5*b^4*c^3*g*h*k*l - 288*a^5*b^4*c^3*e*h*l*m - 96*a^4*b^6*c^2*g
*h*j*m + 2112*a^5*b^3*c^4*d*j*k*l - 160*a^4*b^5*c^3*d*j*k*l + 16992*a^5*b^3
*c^4*d*h*k*m - 6252*a^4*b^5*c^3*d*h*k*m - 4800*a^5*b^3*c^4*e*g*k*m + 2112*a
^5*b^3*c^4*d*g*l*m - 1728*a^5*b^3*c^4*f*g*j*m + 1280*a^4*b^5*c^3*e*g*k*m +
1088*a^5*b^3*c^4*e*f*l*m - 832*a^5*b^3*c^4*e*h*j*m + 816*a^3*b^7*c^2*d*h*k*
m + 576*a^5*b^3*c^4*e*h*k*l - 448*a^5*b^3*c^4*f*h*j*l + 288*a^4*b^5*c^3*f*g
*j*m - 192*a^5*b^3*c^4*g*h*j*k - 192*a^5*b^3*c^4*f*g*k*l + 192*a^4*b^5*c^3*
e*h*j*m - 112*a^4*b^5*c^3*d*g*l*m + 96*a^4*b^5*c^3*f*h*j*l - 96*a^3*b^7*c^2
*e*g*k*m + 80*a^4*b^5*c^3*f*g*k*l + 32*a^4*b^5*c^3*g*h*j*k - 11456*a^5*b^2*
c^5*d*f*k*m + 4992*a^5*b^2*c^5*d*h*j*l - 4608*a^5*b^2*c^5*e*g*j*l - 4224*a^
5*b^2*c^5*d*e*l*m + 3456*a^5*b^2*c^5*e*f*j*m + 3456*a^5*b^2*c^5*d*g*k*l + 2
432*a^5*b^2*c^5*d*g*j*m - 1312*a^4*b^4*c^4*d*h*j*l + 1272*a^3*b^6*c^3*d*f*k
*m - 1056*a^4*b^4*c^4*d*g*k*l + 896*a^5*b^2*c^5*f*g*j*k + 768*a^4*b^4*c^4*e
*g*j*l - 576*a^4*b^4*c^4*e*f*j*m - 480*a^4*b^4*c^4*d*g*j*m + 384*a^5*b^2*c^
5*e*h*j*k + 384*a^5*b^2*c^5*e*f*k*l - 232*a^2*b^8*c^2*d*f*k*m + 224*a^4*b^4
*c^4*d*e*l*m - 160*a^4*b^4*c^4*e*f*k*l - 96*a^4*b^4*c^4*f*g*j*k + 96*a^3*b^
6*c^3*d*h*j*l + 80*a^3*b^6*c^3*d*g*k*l - 64*a^4*b^4*c^4*e*h*j*k - 24*a^4*b^
4*c^4*d*f*k*m + 416*a^4*b^4*c^4*e*g*h*m + 384*a^5*b^2*c^5*f*g*h*l + 384*a^5
*b^2*c^5*e*g*h*m + 224*a^4*b^4*c^4*f*g*h*l - 96*a^3*b^6*c^3*e*g*h*m - 48*a^
3*b^6*c^3*f*g*h*l + 2112*a^4*b^3*c^5*d*e*k*l - 960*a^4*b^3*c^5*d*f*j*l + 96

0*a^4*b^3*c^5*d*e*j*m + 384*a^3*b^5*c^4*d*f*j*l + 320*a^4*b^3*c^5*d*g*j*k +
192*a^4*b^3*c^5*e*f*j*k - 160*a^3*b^5*c^4*d*e*k*l - 32*a^2*b^7*c^3*d*f*j*l
+ 7392*a^4*b^3*c^5*d*f*h*m - 2496*a^4*b^3*c^5*d*g*h*l - 1728*a^4*b^3*c^5*e
*f*g*m - 1500*a^3*b^5*c^4*d*f*h*m + 656*a^3*b^5*c^4*d*g*h*l - 448*a^4*b^3*c
^5*e*f*h*l + 288*a^3*b^5*c^4*e*f*g*m - 192*a^4*b^3*c^5*f*g*h*j - 192*a^4*b^
3*c^5*e*g*h*k + 96*a^3*b^5*c^4*e*f*h*l - 48*a^2*b^7*c^3*d*g*h*l + 32*a^3*b^
5*c^4*e*g*h*k - 16*a^2*b^7*c^3*d*f*h*m - 640*a^4*b^2*c^6*d*e*j*k + 4992*a^4
*b^2*c^6*d*e*h*l - 3584*a^4*b^2*c^6*d*f*h*k + 2432*a^4*b^2*c^6*d*e*g*m - 13
12*a^3*b^4*c^5*d*e*h*l + 896*a^4*b^2*c^6*e*f*g*k + 896*a^4*b^2*c^6*d*g*h*j
+ 640*a^4*b^2*c^6*d*f*g*l + 600*a^3*b^4*c^5*d*f*h*k + 480*a^3*b^4*c^5*d*f*g
*l - 480*a^3*b^4*c^5*d*e*g*m + 384*a^4*b^2*c^6*e*f*h*j - 192*a^2*b^6*c^4*d*
f*g*l - 96*a^3*b^4*c^5*e*f*g*k - 96*a^3*b^4*c^5*d*g*h*j + 96*a^2*b^6*c^4*d*
e*h*l + 12*a^2*b^6*c^4*d*f*h*k - 960*a^3*b^3*c^6*d*e*f*l + 384*a^2*b^5*c^5*
d*e*f*l + 320*a^3*b^3*c^6*d*e*g*k - 192*a^3*b^3*c^6*d*f*g*j + 192*a^3*b^3*c
^6*d*e*h*j + 32*a^2*b^5*c^5*d*f*g*j - 192*a^3*b^3*c^6*e*f*g*h + 384*a^3*b^2
*c^7*d*e*f*j - 64*a^2*b^4*c^6*d*e*f*j + 896*a^3*b^2*c^7*d*e*g*h - 96*a^2*b^
4*c^6*d*e*g*h - 192*a^2*b^3*c^7*d*e*f*g + 496*a^7*b^4*c*k*l^2*m - 4752*a^7*
b^4*c*j*l*m^2 + 96*a^5*b^6*c*j^2*k*m - 6144*a^8*b*c^3*h*l^2*m - 168*a^6*b^5
*c*h*l^2*m + 6400*a^8*b*c^3*g*l*m^2 - 2862*a^6*b^5*c*h*k*m^2 + 2376*a^6*b^5
*c*g*l*m^2 - 1632*a^7*b*c^4*h^2*k*m - 480*a^8*b*c^3*h*k*m^2 - 180*a^5*b^6*c
*h*k^2*m + 54*a^4*b^7*c*h^2*k*m - 384*a^7*b*c^4*h*j^2*m + 120*a^5*b^6*c*h*k
*l^2 + 56*a^5*b^6*c*f*l^2*m + 24*a^3*b^8*c*g^2*k*m + 4512*a^7*b*c^4*f*k^2*m
- 2304*a^7*b*c^4*g*k^2*l - 1680*a^5*b^6*c*g*j*m^2 + 1184*a^6*b*c^5*f^2*k*m
+ 804*a^5*b^6*c*f*k*m^2 + 432*a^5*b^6*c*e*l*m^2 + 60*a^4*b^7*c*f*k^2*m + 6
*a^2*b^9*c*f^2*k*m - 13312*a^7*b*c^4*d*l^2*m + 2048*a^7*b*c^4*g*j*l^2 - 102
4*a^7*b*c^4*f*k*l^2 + 64*a^4*b^7*c*g*j*l^2 + 56*a^4*b^7*c*d*l^2*m - 40*a^4*
b^7*c*f*k*l^2 + 13440*a^7*b*c^4*e*j*m^2 - 8928*a^5*b*c^6*d^2*k*m - 6240*a^7
*b*c^4*d*k*m^2 + 1614*a^4*b^7*c*d*k*m^2 - 288*a^4*b^7*c*e*j*m^2 - 170*a*b^9
*c^2*d^2*k*m + 60*a^3*b^8*c*d*k^2*m + 4608*a^6*b*c^5*e*j^2*l + 4608*a^5*b*c
^6*e^2*j*l - 2432*a^6*b*c^5*d*j^2*m + 1440*a^7*b*c^4*f*h*m^2 - 896*a^6*b*c^
5*f*j^2*k - 864*a^6*b*c^5*f*h^2*m - 558*a^4*b^7*c*f*h*m^2 + 256*a^6*b*c^5*g
*h^2*l - 40*a^3*b^8*c*d*k*l^2 - 1920*a^6*b*c^5*e*j*k^2 - 384*a^5*b*c^6*e^2*
h*m + 24*a^3*b^8*c*f*h*l^2 - 16*a*b^8*c^3*d^2*j*l + 2208*a^6*b*c^5*f*h*k^2
- 1044*a^3*b^8*c*d*h*m^2 + 800*a^5*b*c^6*f^2*h*k - 256*a^5*b*c^6*f^2*g*l +
144*a^3*b^8*c*e*g*m^2 - 116*a*b^8*c^3*d^2*h*m + 8192*a^6*b*c^5*d*h*l^2 + 20
48*a^6*b*c^5*e*g*l^2 + 24*a^2*b^9*c*d*h*l^2 - 5856*a^4*b*c^7*d^2*f*m + 4896
*a^4*b*c^7*d^2*h*k + 2720*a^6*b*c^5*d*f*m^2 + 2304*a^4*b*c^7*d^2*g*l + 1824
*a^5*b*c^6*d*h^2*k + 438*a*b^7*c^4*d^2*f*m - 384*a^5*b*c^6*e*h^2*j + 318*a^
2*b^9*c*d*f*m^2 - 168*a*b^7*c^4*d^2*g*l + 42*a*b^7*c^4*d^2*h*k - 36*a*b^8*c
^3*d*f^2*m - 2432*a^4*b*c^7*d*e^2*m + 1536*a^5*b*c^6*e*g*j^2 + 1536*a^4*b*c
^7*e^2*g*j - 896*a^5*b*c^6*d*h*j^2 - 896*a^4*b*c^7*e^2*f*k + 4896*a^5*b*c^6
*d*f*k^2 + 1824*a^4*b*c^7*d*f^2*k - 384*a^4*b*c^7*e*f^2*j + 336*a*b^6*c^5*d
^2*e*l - 156*a*b^6*c^5*d^2*f*k + 16*a*b^6*c^5*d^2*g*j + 12*a*b^7*c^4*d*f^2*
k - 2*a*b^9*c^2*d*f*k^2 - 1920*a^3*b*c^8*d^2*e*j - 32*a*b^5*c^6*d^2*e*j + 2
208*a^3*b*c^8*d^2*f*h + 800*a^4*b*c^7*d*f*h^2 - 102*a*b^5*c^6*d^2*f*h + 12*

a*b^6*c^5*d*f^2*h - 2*a*b^7*c^4*d*f*h^2 - 896*a^3*b*c^8*d*e^2*h - 8*a*b^6*c
^5*d*f*g^2 - 240*a*b^4*c^7*d^2*e*g - 32*a*b^4*c^7*d*e^2*f + 5120*a^8*c^4*h*
j*l*m + 15360*a^7*c^5*d*j*l*m - 7680*a^7*c^5*e*j*k*m + 3072*a^7*c^5*f*j*k*l
 + 5120*a^7*c^5*e*h*l*m + 1920*a^7*c^5*f*h*k*m + 15360*a^6*c^6*d*e*l*m + 57
60*a^6*c^6*d*f*k*m + 3072*a^6*c^6*e*f*k*l - 3072*a^6*c^6*d*h*j*l - 2560*a^6
*c^6*e*f*j*m + 1536*a^6*c^6*e*h*j*k + 4608*a^5*c^7*d*e*j*k - 3072*a^5*c^7*d
*e*h*l - 1152*a^5*c^7*d*f*h*k + 512*a^5*c^7*e*f*h*j + 1536*a^4*c^8*d*e*f*j
- 8*a*b^10*c*d*f*l^2 - 5568*a^8*b^2*c^2*k*l^2*m + 15552*a^8*b^2*c^2*j*l*m^2
 + 4800*a^7*b^2*c^3*j^2*k*m - 1280*a^6*b^4*c^2*j^2*k*m + 2080*a^7*b^3*c^2*h
*l^2*m - 1088*a^7*b^2*c^3*j*k^2*l + 48*a^6*b^4*c^2*j*k^2*l - 8544*a^7*b^2*c
^3*h*k^2*m - 7776*a^7*b^3*c^2*g*l*m^2 + 7632*a^7*b^3*c^2*h*k*m^2 + 3600*a^6
*b^3*c^3*h^2*k*m + 2484*a^6*b^4*c^2*h*k^2*m - 918*a^5*b^5*c^2*h^2*k*m + 480
0*a^7*b^2*c^3*h*k*l^2 - 1424*a^6*b^4*c^2*h*k*l^2 + 1200*a^5*b^4*c^3*g^2*k*m
 - 960*a^6*b^2*c^4*g^2*k*m - 528*a^6*b^4*c^2*f*l^2*m - 416*a^6*b^3*c^3*h*j^
2*m - 320*a^4*b^6*c^2*g^2*k*m + 192*a^7*b^2*c^3*f*l^2*m + 96*a^5*b^5*c^2*h*
j^2*m + 15552*a^7*b^2*c^3*e*l*m^2 - 6720*a^7*b^2*c^3*g*j*m^2 + 6160*a^6*b^4
*c^2*g*j*m^2 - 4752*a^6*b^4*c^2*e*l*m^2 - 2016*a^7*b^2*c^3*f*k*m^2 - 1164*a
^6*b^4*c^2*f*k*m^2 + 1104*a^5*b^3*c^4*f^2*k*m + 1008*a^6*b^3*c^3*f*k^2*m +
960*a^6*b^2*c^4*h^2*j*l - 678*a^5*b^5*c^2*f*k^2*m + 544*a^6*b^3*c^3*g*k^2*l
 - 144*a^5*b^4*c^3*h^2*j*l - 102*a^4*b^5*c^3*f^2*k*m - 62*a^3*b^7*c^2*f^2*k
*m - 24*a^5*b^5*c^2*g*k^2*l + 6432*a^6*b^3*c^3*d*l^2*m + 4800*a^5*b^2*c^5*e
^2*k*m - 2304*a^6*b^2*c^4*g*j^2*l + 1920*a^6*b^3*c^3*g*j*l^2 + 1728*a^6*b^2
*c^4*f*j^2*m - 1280*a^4*b^4*c^4*e^2*k*m + 1152*a^5*b^3*c^4*g^2*j*l - 1032*a
^5*b^5*c^2*d*l^2*m - 864*a^6*b^3*c^3*f*k*l^2 - 768*a^5*b^5*c^2*g*j*l^2 + 40
8*a^5*b^5*c^2*f*k*l^2 + 384*a^5*b^4*c^3*g*j^2*l - 288*a^5*b^4*c^3*f*j^2*m +
 192*a^6*b^2*c^4*h*j^2*k - 192*a^4*b^5*c^3*g^2*j*l + 96*a^3*b^6*c^3*e^2*k*m
 - 32*a^5*b^4*c^3*h*j^2*k - 21120*a^6*b^2*c^4*d*k^2*m + 20880*a^6*b^3*c^3*d
*k*m^2 + 19760*a^4*b^3*c^5*d^2*k*m - 12320*a^6*b^3*c^3*e*j*m^2 - 9750*a^5*b
^5*c^2*d*k*m^2 - 9390*a^3*b^5*c^4*d^2*k*m + 8460*a^5*b^4*c^3*d*k^2*m + 3360
*a^5*b^5*c^2*e*j*m^2 + 1860*a^2*b^7*c^3*d^2*k*m - 1218*a^4*b^6*c^2*d*k^2*m
- 1088*a^6*b^2*c^4*e*k^2*l + 960*a^6*b^2*c^4*g*j*k^2 - 240*a^5*b^4*c^3*g*j*
k^2 + 192*a^5*b^2*c^5*f^2*j*l - 104*a^4*b^5*c^3*g^2*h*m - 96*a^5*b^3*c^4*g^
2*h*m + 48*a^5*b^4*c^3*e*k^2*l + 48*a^4*b^4*c^4*f^2*j*l + 24*a^3*b^7*c^2*g^
2*h*m + 16*a^4*b^6*c^2*g*j*k^2 - 16*a^3*b^6*c^3*f^2*j*l + 13376*a^6*b^2*c^4
*d*k*l^2 - 5136*a^5*b^4*c^3*d*k*l^2 - 3840*a^6*b^2*c^4*e*j*l^2 + 1536*a^5*b
^4*c^3*e*j*l^2 + 1392*a^5*b^3*c^4*f*h^2*m + 1386*a^5*b^5*c^2*f*h*m^2 - 768*
a^5*b^3*c^4*e*j^2*l + 768*a^4*b^6*c^2*d*k*l^2 - 768*a^4*b^3*c^5*e^2*j*l - 5
88*a^4*b^4*c^4*f^2*h*m - 480*a^5*b^3*c^4*g*h^2*l + 480*a^5*b^3*c^4*d*j^2*m
- 480*a^5*b^2*c^5*f^2*h*m - 128*a^4*b^6*c^2*e*j*l^2 + 100*a^3*b^6*c^3*f^2*h
*m + 96*a^5*b^3*c^4*f*j^2*k + 72*a^4*b^5*c^3*g*h^2*l - 54*a^4*b^5*c^3*f*h^2
*m - 48*a^6*b^3*c^3*f*h*m^2 - 36*a^3*b^7*c^2*f*h^2*m + 6*a^2*b^8*c^2*f^2*h*
m + 6848*a^4*b^2*c^6*d^2*j*l - 2448*a^3*b^4*c^5*d^2*j*l + 624*a^5*b^4*c^3*f
*h*l^2 + 576*a^6*b^2*c^4*f*h*l^2 + 480*a^5*b^3*c^4*e*j*k^2 + 432*a^4*b^4*c^
4*f*g^2*m - 416*a^4*b^3*c^5*e^2*h*m + 336*a^2*b^6*c^4*d^2*j*l - 320*a^5*b^2
*c^5*f*g^2*m - 256*a^4*b^6*c^2*f*h*l^2 + 192*a^5*b^2*c^5*g^2*h*k + 96*a^3*b

^5*c^4*e^2*h*m - 72*a^3*b^6*c^3*f*g^2*m + 48*a^4*b^4*c^4*g^2*h*k - 32*a^4*b
^5*c^3*e*j*k^2 - 8*a^3*b^6*c^3*g^2*h*k + 24768*a^6*b^2*c^4*d*h*m^2 - 21108*
a^5*b^4*c^3*d*h*m^2 - 10048*a^4*b^2*c^6*d^2*h*m + 7218*a^4*b^6*c^2*d*h*m^2
- 6720*a^6*b^2*c^4*e*g*m^2 + 6160*a^5*b^4*c^3*e*g*m^2 - 2592*a^5*b^2*c^5*d*
h^2*m - 1680*a^4*b^6*c^2*e*g*m^2 + 1068*a^3*b^4*c^5*d^2*h*m + 960*a^5*b^2*c
^5*e*h^2*l - 876*a^4*b^4*c^4*d*h^2*m - 864*a^5*b^2*c^5*f*h^2*k + 546*a^2*b^
6*c^4*d^2*h*m + 432*a^3*b^6*c^3*d*h^2*m + 336*a^4*b^3*c^5*f^2*h*k - 320*a^5
*b^2*c^5*d*j^2*k + 192*a^5*b^2*c^5*g*h^2*j + 144*a^5*b^3*c^4*f*h*k^2 - 144*
a^4*b^4*c^4*e*h^2*l - 102*a^4*b^5*c^3*f*h*k^2 - 96*a^4*b^3*c^5*f^2*g*l - 36
*a^2*b^8*c^2*d*h^2*m - 30*a^3*b^5*c^4*f^2*h*k - 24*a^3*b^5*c^4*f^2*g*l + 16
*a^4*b^4*c^4*g*h^2*j - 12*a^4*b^4*c^4*f*h^2*k + 12*a^3*b^6*c^3*f*h^2*k + 8*
a^2*b^7*c^3*f^2*g*l + 6*a^3*b^7*c^2*f*h*k^2 - 2*a^2*b^7*c^3*f^2*h*k - 9312*
a^5*b^3*c^4*d*h*l^2 + 3288*a^4*b^5*c^3*d*h*l^2 - 2304*a^4*b^2*c^6*e^2*g*l +
1920*a^5*b^3*c^4*e*g*l^2 + 1728*a^4*b^2*c^6*e^2*f*m + 1152*a^4*b^3*c^5*e*g
^2*l - 768*a^4*b^5*c^3*e*g*l^2 - 608*a^4*b^3*c^5*d*g^2*m - 472*a^3*b^7*c^2*
d*h*l^2 + 384*a^3*b^4*c^5*e^2*g*l - 288*a^3*b^4*c^5*e^2*f*m - 224*a^4*b^3*c
^5*f*g^2*k + 192*a^5*b^2*c^5*f*h*j^2 + 192*a^4*b^2*c^6*e^2*h*k - 192*a^3*b^
5*c^4*e*g^2*l + 120*a^3*b^5*c^4*d*g^2*m + 64*a^3*b^7*c^2*e*g*l^2 - 32*a^3*b
^4*c^5*e^2*h*k + 24*a^3*b^5*c^4*f*g^2*k + 9936*a^3*b^3*c^6*d^2*f*m + 3786*a
^4*b^5*c^3*d*f*m^2 - 3552*a^5*b^2*c^5*d*h*k^2 - 3486*a^2*b^5*c^5*d^2*f*m -
3424*a^3*b^3*c^6*d^2*g*l - 1868*a^3*b^7*c^2*d*f*m^2 + 1332*a^4*b^4*c^4*d*h*
k^2 - 1296*a^5*b^3*c^4*d*f*m^2 - 1236*a^3*b^4*c^5*d*f^2*m + 1224*a^2*b^5*c^
5*d^2*g*l - 1152*a^4*b^2*c^6*d*f^2*m + 960*a^5*b^2*c^5*e*g*k^2 - 496*a^3*b^
3*c^6*d^2*h*k + 462*a^2*b^6*c^4*d*f^2*m + 432*a^4*b^3*c^5*d*h^2*k - 240*a^4
*b^4*c^4*e*g*k^2 - 222*a^2*b^5*c^5*d^2*h*k + 192*a^4*b^2*c^6*f^2*g*j + 192*
a^4*b^2*c^6*e*f^2*l - 174*a^3*b^5*c^4*d*h^2*k - 156*a^3*b^6*c^3*d*h*k^2 + 4
8*a^3*b^4*c^5*e*f^2*l - 32*a^4*b^3*c^5*e*h^2*j + 16*a^3*b^6*c^3*e*g*k^2 + 1
6*a^3*b^4*c^5*f^2*g*j - 16*a^2*b^6*c^4*e*f^2*l + 12*a^2*b^7*c^3*d*h^2*k + 6
*a^2*b^8*c^2*d*h*k^2 + 1728*a^5*b^2*c^5*d*f*l^2 + 1392*a^4*b^4*c^4*d*f*l^2
- 840*a^3*b^6*c^3*d*f*l^2 - 768*a^4*b^2*c^6*e*g^2*j + 576*a^4*b^2*c^6*d*g^2
*k + 480*a^3*b^3*c^6*d*e^2*m + 144*a^2*b^8*c^2*d*f*l^2 + 96*a^4*b^3*c^5*d*h
*j^2 + 96*a^3*b^3*c^6*e^2*f*k - 80*a^3*b^4*c^5*d*g^2*k + 6848*a^3*b^2*c^7*d
^2*e*l - 3552*a^3*b^2*c^7*d^2*f*k - 2448*a^2*b^4*c^6*d^2*e*l + 1332*a^2*b^4
*c^6*d^2*f*k + 960*a^3*b^2*c^7*d^2*g*j - 496*a^4*b^3*c^5*d*f*k^2 + 432*a^3*
b^3*c^6*d*f^2*k - 240*a^2*b^4*c^6*d^2*g*j - 222*a^3*b^5*c^4*d*f*k^2 - 174*a
^2*b^5*c^5*d*f^2*k + 64*a^4*b^2*c^6*f*g^2*h + 48*a^3*b^4*c^5*f*g^2*h + 42*a
^2*b^7*c^3*d*f*k^2 - 32*a^3*b^3*c^6*e*f^2*j - 320*a^3*b^2*c^7*d*e^2*k + 192
*a^4*b^2*c^6*e*g*h^2 + 192*a^4*b^2*c^6*d*f*j^2 - 32*a^3*b^4*c^5*d*f*j^2 + 1
6*a^3*b^4*c^5*e*g*h^2 + 480*a^2*b^3*c^7*d^2*e*j - 224*a^3*b^3*c^6*d*g^2*h +
192*a^3*b^2*c^7*e^2*f*h + 24*a^2*b^5*c^5*d*g^2*h - 864*a^3*b^2*c^7*d*f^2*h
+ 336*a^3*b^3*c^6*d*f*h^2 + 192*a^3*b^2*c^7*e*f^2*g + 144*a^2*b^3*c^7*d^2*
f*h - 30*a^2*b^5*c^5*d*f*h^2 + 16*a^2*b^4*c^6*e*f^2*g - 12*a^2*b^4*c^6*d*f^
2*h + 192*a^3*b^2*c^7*d*f*g^2 + 96*a^2*b^3*c^7*d*e^2*h + 48*a^2*b^4*c^6*d*f
*g^2 + 960*a^2*b^2*c^8*d^2*e*g + 192*a^2*b^2*c^8*d*e^2*f - 7680*a^9*b*c^2*l
^2*m^2 + 3152*a^8*b^3*c*l^2*m^2 + 2070*a^7*b^4*c*k^2*m^2 - 1840*a^7*b^3*c^2

*k^3*m + 6720*a^8*b*c^3*j^2*m^2 - 3072*a^8*b*c^3*k^2*l^2 + 1680*a^6*b^5*c*j
^2*m^2 - 100*a^6*b^5*c*k^2*l^2 - 2176*a^7*b^3*c^2*j*l^3 - 256*a^6*b^3*c^3*j
^3*l - 64*a^5*b^6*c*j^2*l^2 - 12480*a^8*b^2*c^2*h*m^3 + 972*a^5*b^6*c*h^2*m
^2 - 960*a^7*b*c^4*j^2*k^2 - 252*a^5*b^4*c^3*h^3*m - 192*a^6*b^2*c^4*h^3*m
+ 54*a^4*b^6*c^2*h^3*m + 1536*a^7*b*c^4*h^2*l^2 + 420*a^4*b^7*c*g^2*m^2 - 3
6*a^4*b^7*c*h^2*l^2 - 3072*a^7*b^2*c^3*g*l^3 + 2096*a^7*b^3*c^2*f*m^3 + 108
8*a^6*b^4*c^2*g*l^3 - 496*a^6*b^3*c^3*h*k^3 - 192*a^4*b^4*c^4*g^3*l + 176*a
^4*b^3*c^5*f^3*m + 144*a^5*b^3*c^4*h^3*k + 78*a^3*b^8*c*f^2*m^2 + 54*a^3*b^
5*c^4*f^3*m + 32*a^3*b^6*c^3*g^3*l + 30*a^5*b^5*c^2*h*k^3 - 18*a^4*b^5*c^3*
h^3*k - 18*a^2*b^7*c^3*f^3*m - 16*a^3*b^8*c*g^2*l^2 + 6720*a^6*b*c^5*e^2*m^
2 - 192*a^6*b*c^5*h^2*j^2 - 4*a^2*b^9*c*f^2*l^2 - 35040*a^7*b^2*c^3*d*m^3 +
 14300*a^6*b^4*c^2*d*m^3 - 12000*a^3*b^2*c^7*d^3*m + 4380*a^2*b^4*c^6*d^3*m
 - 2176*a^6*b^3*c^3*e*l^3 - 256*a^3*b^3*c^6*e^3*l - 192*a^6*b^2*c^4*f*k^3 +
 192*a^5*b^5*c^2*e*l^3 - 192*a^4*b^2*c^6*f^3*k + 132*a^5*b^4*c^3*f*k^3 + 12
8*a^4*b^3*c^5*g^3*j - 28*a^3*b^4*c^5*f^3*k - 10*a^4*b^6*c^2*f*k^3 + 6*a^2*b
^6*c^4*f^3*k + 10752*a^5*b*c^6*d^2*l^2 - 960*a^5*b*c^6*e^2*k^2 - 192*a^5*b*
c^6*f^2*j^2 + 108*a*b^9*c^2*d^2*l^2 - 1680*a^5*b^3*c^4*d*k^3 - 1680*a^2*b^3
*c^7*d^3*k + 222*a^4*b^5*c^3*d*k^3 + 30*a*b^8*c^3*d^2*k^2 - 10*a^3*b^7*c^2*
d*k^3 - 960*a^4*b*c^7*d^2*j^2 + 80*a^4*b^3*c^5*f*h^3 + 80*a^3*b^3*c^6*f^3*h
 + 6*a^3*b^5*c^4*f*h^3 + 6*a^2*b^5*c^5*f^3*h - 192*a^4*b*c^7*e^2*h^2 - 192*
a^4*b^2*c^6*d*h^3 - 192*a^2*b^2*c^8*d^3*h + 128*a^3*b^3*c^6*e*g^3 - 28*a^3*
b^4*c^5*d*h^3 + 12*a*b^6*c^5*d^2*h^2 + 6*a^2*b^6*c^4*d*h^3 - 192*a^3*b*c^8*
e^2*f^2 + 60*a*b^5*c^6*d^2*g^2 + 198*a*b^4*c^7*d^2*f^2 + 144*a^2*b^3*c^7*d*
f^3 - 960*a^2*b*c^9*d^2*e^2 + 240*a*b^3*c^8*d^2*e^2 + 15360*a^9*c^3*k*l^2*m
 - 12800*a^9*c^3*j*l*m^2 - 3840*a^8*c^4*j^2*k*m + 432*a^6*b^6*j*l*m^2 + 460
8*a^8*c^4*j*k^2*l + 2880*a^8*c^4*h*k^2*m + 5120*a^8*c^4*f*l^2*m - 3072*a^8*
c^4*h*k*l^2 + 270*a^5*b^7*h*k*m^2 - 216*a^5*b^7*g*l*m^2 - 12800*a^8*c^4*e*l
*m^2 - 4800*a^8*c^4*f*k*m^2 - 512*a^7*c^5*h^2*j*l - 3840*a^6*c^6*e^2*k*m -
1280*a^7*c^5*f*j^2*m + 768*a^7*c^5*h*j^2*k + 144*a^4*b^8*g*j*m^2 - 90*a^4*b
^8*f*k*m^2 + 8640*a^7*c^5*d*k^2*m + 4608*a^7*c^5*e*k^2*l + 512*a^6*c^6*f^2*
j*l - 9216*a^7*c^5*d*k*l^2 - 4096*a^7*c^5*e*j*l^2 + 320*a^6*c^6*f^2*h*m - 9
0*a^3*b^9*d*k*m^2 + 15200*a^9*b*c^2*k*m^3 - 6192*a^8*b^3*c*k*m^3 + 5472*a^8
*b*c^3*k^3*m - 4608*a^5*c^7*d^2*j*l - 1024*a^7*c^5*f*h*l^2 + 150*a^6*b^5*c*
k^3*m + 54*a^3*b^9*f*h*m^2 + 6*b^10*c^2*d^2*h*m - 14400*a^7*c^5*d*h*m^2 + 8
640*a^5*c^7*d^2*h*m + 2880*a^6*c^6*d*h^2*m + 2304*a^6*c^6*d*j^2*k - 512*a^6
*c^6*e*h^2*l - 192*a^6*c^6*f*h^2*k + 6144*a^8*b*c^3*j*l^3 + 1536*a^7*b*c^4*
j^3*l - 1280*a^5*c^7*e^2*f*m + 768*a^5*c^7*e^2*h*k + 256*a^6*c^6*f*h*j^2 +
192*a^6*b^5*c*j*l^3 + 54*a^2*b^10*d*h*m^2 - 18*b^9*c^3*d^2*f*m + 8*b^9*c^3*
d^2*g*l - 2*b^9*c^3*d^2*h*k + 4068*a^7*b^4*c*h*m^3 - 1728*a^6*c^6*d*h*k^2 +
 960*a^5*c^7*d*f^2*m + 512*a^5*c^7*e*f^2*l - 3072*a^6*c^6*d*f*l^2 - 16*b^8*
c^4*d^2*e*l + 6*b^8*c^4*d^2*f*k - 4608*a^4*c^8*d^2*e*l + 2400*a^8*b*c^3*f*m
^3 + 2016*a^7*b*c^4*h*k^3 - 1728*a^4*c^8*d^2*f*k - 1146*a^6*b^5*c*f*m^3 + 2
24*a^6*b*c^5*h^3*k - 96*a^5*b^6*c*g*l^3 + 96*a^5*b*c^6*f^3*m + 2304*a^4*c^8
*d*e^2*k + 768*a^5*c^7*d*f*j^2 + 6144*a^7*b*c^4*e*l^3 - 2280*a^5*b^6*c*d*m^
3 + 1536*a^4*b*c^7*e^3*l - 616*a*b^6*c^5*d^3*m + 512*a^6*b*c^5*g*j^3 + 256*

a^4*c^8*e^2*f*h + 240*a*b^10*c*d^2*m^2 + 6*b^7*c^5*d^2*f*h - 192*a^4*c^8*d*
f^2*h + 4320*a^6*b*c^5*d*k^3 + 4320*a^3*b*c^8*d^3*k + 222*a*b^5*c^6*d^3*k +
16*b^6*c^6*d^2*e*g + 96*a^5*b*c^6*f*h^3 + 96*a^4*b*c^7*f^3*h + 768*a^3*c^9
*d*e^2*f + 512*a^3*b*c^8*e^3*g + 132*a*b^4*c^7*d^3*h + 2016*a^2*b*c^9*d^3*f
- 496*a*b^3*c^8*d^3*f + 224*a^3*b*c^8*d*f^3 - 18*a*b^5*c^6*d*f^3 - 3264*a^
8*b^2*c^2*k^2*m^2 - 6160*a^7*b^3*c^2*j^2*m^2 + 1104*a^7*b^3*c^2*k^2*l^2 - 1
920*a^7*b^2*c^3*j^2*l^2 + 768*a^6*b^4*c^2*j^2*l^2 + 3888*a^7*b^2*c^3*h^2*m^
2 - 3510*a^6*b^4*c^2*h^2*m^2 + 240*a^6*b^3*c^3*j^2*k^2 - 16*a^5*b^5*c^2*j^2
*k^2 + 1680*a^6*b^3*c^3*g^2*m^2 - 1648*a^6*b^3*c^3*h^2*l^2 - 1540*a^5*b^5*c
^2*g^2*m^2 + 444*a^5*b^5*c^2*h^2*l^2 - 960*a^6*b^2*c^4*h^2*k^2 - 576*a^6*b^
2*c^4*f^2*m^2 - 512*a^6*b^2*c^4*g^2*l^2 - 480*a^5*b^4*c^3*g^2*l^2 + 198*a^5
*b^4*c^3*h^2*k^2 + 192*a^4*b^6*c^2*g^2*l^2 - 186*a^5*b^4*c^3*f^2*m^2 - 97*a
^4*b^6*c^2*f^2*m^2 - 9*a^4*b^6*c^2*h^2*k^2 - 6160*a^5*b^3*c^4*e^2*m^2 + 168
0*a^4*b^5*c^3*e^2*m^2 - 240*a^5*b^3*c^4*g^2*k^2 - 240*a^5*b^3*c^4*f^2*l^2 -
144*a^3*b^7*c^2*e^2*m^2 + 60*a^4*b^5*c^3*g^2*k^2 - 36*a^4*b^5*c^3*f^2*l^2
+ 36*a^3*b^7*c^2*f^2*l^2 - 16*a^5*b^3*c^4*h^2*j^2 - 4*a^3*b^7*c^2*g^2*k^2 +
38512*a^5*b^2*c^5*d^2*m^2 - 32310*a^4*b^4*c^4*d^2*m^2 + 12720*a^3*b^6*c^3*
d^2*m^2 - 2500*a^2*b^8*c^2*d^2*m^2 - 1920*a^5*b^2*c^5*e^2*l^2 + 768*a^4*b^4
*c^4*e^2*l^2 - 464*a^5*b^2*c^5*f^2*k^2 - 384*a^5*b^2*c^5*g^2*j^2 - 64*a^3*b
^6*c^3*e^2*l^2 + 42*a^4*b^4*c^4*f^2*k^2 + 12*a^3*b^6*c^3*f^2*k^2 - 13104*a^
4*b^3*c^5*d^2*l^2 + 5628*a^3*b^5*c^4*d^2*l^2 - 1128*a^2*b^7*c^3*d^2*l^2 + 2
40*a^4*b^3*c^5*e^2*k^2 - 16*a^4*b^3*c^5*f^2*j^2 - 16*a^3*b^5*c^4*e^2*k^2 -
2880*a^4*b^2*c^6*d^2*k^2 + 1750*a^3*b^4*c^5*d^2*k^2 - 345*a^2*b^6*c^4*d^2*k
^2 - 48*a^4*b^3*c^5*g^2*h^2 - 4*a^3*b^5*c^4*g^2*h^2 + 240*a^3*b^3*c^6*d^2*j
^2 - 192*a^4*b^2*c^6*f^2*h^2 - 42*a^3*b^4*c^5*f^2*h^2 - 16*a^2*b^5*c^5*d^2*
j^2 - 48*a^3*b^3*c^6*f^2*g^2 - 16*a^3*b^3*c^6*e^2*h^2 - 4*a^2*b^5*c^5*f^2*g
^2 - 464*a^3*b^2*c^7*d^2*h^2 - 384*a^3*b^2*c^7*e^2*g^2 + 42*a^2*b^4*c^6*d^2
*h^2 - 240*a^2*b^3*c^7*d^2*g^2 - 16*a^2*b^3*c^7*e^2*f^2 - 960*a^2*b^2*c^8*d
^2*f^2 + 6*b^11*c*d^2*k*m - 18*a*b^11*d*f*m^2 - 7200*a^9*c^3*k^2*m^2 - 324*
a^7*b^5*l^2*m^2 - 225*a^6*b^6*k^2*m^2 - 2048*a^8*c^4*j^2*l^2 - 144*a^5*b^7*
j^2*m^2 - 2400*a^8*c^4*h^2*m^2 - 81*a^4*b^8*h^2*m^2 - 800*a^7*c^5*f^2*m^2 -
288*a^7*c^5*h^2*k^2 - 36*a^3*b^9*g^2*m^2 - 9*a^2*b^10*f^2*m^2 - 21600*a^6*
c^6*d^2*m^2 - 2048*a^6*c^6*e^2*l^2 - 864*a^6*c^6*f^2*k^2 - 2592*a^5*c^7*d^2
*k^2 - 1536*a^5*c^7*e^2*j^2 + 1536*a^8*b^2*c^2*l^4 - 32*a^5*c^7*f^2*h^2 + 3
60*a^7*b^2*c^3*k^4 - 25*a^6*b^4*c^2*k^4 - 864*a^4*c^8*d^2*h^2 - 4*b^7*c^5*d
^2*g^2 - 9*b^6*c^6*d^2*f^2 - 288*a^3*c^9*d^2*f^2 - 24*a^5*b^2*c^5*h^4 - 16*
b^5*c^7*d^2*e^2 - 9*a^4*b^4*c^4*h^4 - 16*a^3*b^4*c^5*g^4 - 24*a^3*b^2*c^7*f
^4 - 9*a^2*b^4*c^6*f^4 - a^2*b^8*c^2*f^2*k^2 - a^2*b^6*c^4*f^2*h^2 + 630*a^
7*b^5*k*m^3 + 8000*a^9*c^3*h*m^3 + 320*a^7*c^5*h^3*m - 378*a^6*b^6*h*m^3 +
126*a^5*b^7*f*m^3 + 30*b^8*c^4*d^3*m + 24000*a^8*c^4*d*m^3 + 8640*a^4*c^8*d
^3*m - 1728*a^7*c^5*f*k^3 - 192*a^5*c^7*f^3*k - 4*b^11*c*d^2*l^2 + 126*a^4*
b^8*d*m^3 - 10*b^7*c^5*d^3*k + 4200*a^9*b^2*c*m^4 - 1024*a^6*c^6*e*j^3 - 10
24*a^4*c^8*e^3*j - 144*a^7*b^4*c*l^4 - 10*b^6*c^6*d^3*h - 1728*a^3*c^9*d^3*
h - 192*a^5*c^7*d*h^3 + 30*b^5*c^7*d^3*f + 360*a*b^2*c^9*d^4 - 9*b^12*d^2*m
^2 - 10000*a^10*c^2*m^4 - 4096*a^9*c^3*l^4 - 441*a^8*b^4*m^4 - 1296*a^8*c^4

*k^4 - 256*a^7*c^5*j^4 - 16*a^6*c^6*h^4 - 16*a^4*c^8*f^4 - 256*a^3*c^9*e^4
- 25*b^4*c^8*d^4 - 1296*a^2*c^10*d^4 - b^10*c^2*d^2*k^2 - b^8*c^4*d^2*h^2,
z, k1)*((6144*a^5*c^9*d + 2048*a^6*c^8*h - 10240*a^7*c^7*m - 288*a^2*b^6*c^
6*d + 1920*a^3*b^4*c^7*d - 5632*a^4*b^2*c^8*d + 16*a^2*b^7*c^5*f - 192*a^3*
b^5*c^6*f + 768*a^4*b^3*c^7*f - 32*a^3*b^6*c^5*h + 384*a^4*b^4*c^6*h - 1536
*a^5*b^2*c^7*h + 16*a^3*b^7*c^4*k - 192*a^4*b^5*c^5*k + 768*a^5*b^3*c^6*k -
 48*a^3*b^8*c^3*m + 736*a^4*b^6*c^4*m - 4224*a^5*b^4*c^5*m + 10752*a^6*b^2*
c^6*m + 16*a*b^8*c^5*d - 1024*a^5*b*c^8*f - 1024*a^6*b*c^7*k)/(8*(64*a^5*c^
6 - a^2*b^6*c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5)) + (x*(32*a^2*b^6*c^6*e
- 2048*a^6*c^8*j - 2048*a^5*c^9*e - 384*a^3*b^4*c^7*e + 1536*a^4*b^2*c^8*e
- 16*a^2*b^7*c^5*g + 192*a^3*b^5*c^6*g - 768*a^4*b^3*c^7*g + 32*a^3*b^6*c^5
*j - 384*a^4*b^4*c^6*j + 1536*a^5*b^2*c^7*j + 32*a^2*b^9*c^3*l - 528*a^3*b^
7*c^4*l + 3264*a^4*b^5*c^5*l - 8960*a^5*b^3*c^6*l + 1024*a^5*b*c^8*g + 9216
*a^6*b*c^7*l))/(4*(64*a^5*c^6 - a^2*b^6*c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c
^5)) - (root(1572864*a^8*b^2*c^10*z^4 - 983040*a^7*b^4*c^9*z^4 + 327680*a^6
*b^6*c^8*z^4 - 61440*a^5*b^8*c^7*z^4 + 6144*a^4*b^10*c^6*z^4 - 256*a^3*b^12
*c^5*z^4 - 1048576*a^9*c^11*z^4 - 1572864*a^8*b^2*c^8*l*z^3 + 983040*a^7*b^
4*c^7*l*z^3 - 327680*a^6*b^6*c^6*l*z^3 + 61440*a^5*b^8*c^5*l*z^3 - 6144*a^4
*b^10*c^4*l*z^3 + 256*a^3*b^12*c^3*l*z^3 + 1048576*a^9*c^9*l*z^3 + 96*a^3*b
^12*c*k*m*z^2 + 98304*a^8*b*c^7*j*l*z^2 + 24576*a^8*b*c^7*h*m*z^2 + 155648*
a^7*b*c^8*d*m*z^2 + 98304*a^7*b*c^8*e*l*z^2 + 57344*a^7*b*c^8*f*k*z^2 + 327
68*a^7*b*c^8*g*j*z^2 + 57344*a^6*b*c^9*d*h*z^2 + 32768*a^6*b*c^9*e*g*z^2 -
32*a*b^10*c^5*d*f*z^2 - 491520*a^8*b^2*c^6*k*m*z^2 + 358400*a^7*b^4*c^5*k*m
*z^2 - 129024*a^6*b^6*c^4*k*m*z^2 + 24768*a^5*b^8*c^3*k*m*z^2 - 2432*a^4*b^
10*c^2*k*m*z^2 - 90112*a^7*b^3*c^6*j*l*z^2 + 30720*a^6*b^5*c^5*j*l*z^2 - 46
08*a^5*b^7*c^4*j*l*z^2 + 256*a^4*b^9*c^3*j*l*z^2 - 21504*a^6*b^5*c^5*h*m*z^
2 + 9216*a^5*b^7*c^4*h*m*z^2 + 8192*a^7*b^3*c^6*h*m*z^2 - 1568*a^4*b^9*c^3*
h*m*z^2 + 96*a^3*b^11*c^2*h*m*z^2 - 172032*a^7*b^2*c^7*f*m*z^2 + 116736*a^6
*b^4*c^6*f*m*z^2 - 49152*a^7*b^2*c^7*g*l*z^2 + 45056*a^6*b^4*c^6*g*l*z^2 -
35840*a^5*b^6*c^5*f*m*z^2 + 24576*a^7*b^2*c^7*h*k*z^2 - 15360*a^5*b^6*c^5*g
*l*z^2 + 5184*a^4*b^8*c^4*f*m*z^2 - 3072*a^5*b^6*c^5*h*k*z^2 + 2304*a^4*b^8
*c^4*g*l*z^2 + 2048*a^6*b^4*c^6*h*k*z^2 + 576*a^4*b^8*c^4*h*k*z^2 - 288*a^3
*b^10*c^3*f*m*z^2 - 128*a^3*b^10*c^3*g*l*z^2 - 32*a^3*b^10*c^3*h*k*z^2 - 14
7456*a^6*b^3*c^7*d*m*z^2 - 90112*a^6*b^3*c^7*e*l*z^2 + 52224*a^5*b^5*c^6*d*
m*z^2 - 49152*a^6*b^3*c^7*f*k*z^2 + 30720*a^5*b^5*c^6*e*l*z^2 - 24576*a^6*b
^3*c^7*g*j*z^2 + 15360*a^5*b^5*c^6*f*k*z^2 - 8192*a^4*b^7*c^5*d*m*z^2 + 614
4*a^5*b^5*c^6*g*j*z^2 - 4608*a^4*b^7*c^5*e*l*z^2 - 2048*a^4*b^7*c^5*f*k*z^2
 - 512*a^4*b^7*c^5*g*j*z^2 + 480*a^3*b^9*c^4*d*m*z^2 + 256*a^3*b^9*c^4*e*l*
z^2 + 96*a^3*b^9*c^4*f*k*z^2 + 131072*a^6*b^2*c^8*d*k*z^2 + 49152*a^6*b^2*c
^8*e*j*z^2 - 43008*a^5*b^4*c^7*d*k*z^2 - 12288*a^5*b^4*c^7*e*j*z^2 + 6144*a
^4*b^6*c^6*d*k*z^2 + 1024*a^4*b^6*c^6*e*j*z^2 - 320*a^3*b^8*c^5*d*k*z^2 + 6
144*a^5*b^4*c^7*f*h*z^2 - 2048*a^4*b^6*c^6*f*h*z^2 + 192*a^3*b^8*c^5*f*h*z^
2 - 49152*a^5*b^3*c^8*d*h*z^2 - 24576*a^5*b^3*c^8*e*g*z^2 + 15360*a^4*b^5*c
^7*d*h*z^2 + 6144*a^4*b^5*c^7*e*g*z^2 - 2048*a^3*b^7*c^6*d*h*z^2 - 512*a^3*
b^7*c^6*e*g*z^2 + 96*a^2*b^9*c^5*d*h*z^2 + 24576*a^5*b^2*c^9*d*f*z^2 - 3072

*a^3*b^6*c^7*d*f*z^2 + 2048*a^4*b^4*c^8*d*f*z^2 + 576*a^2*b^8*c^6*d*f*z^2 -
430080*a^9*b*c^6*m^2*z^2 + 3408*a^4*b^11*c*m^2*z^2 - 64*a^3*b^12*c*l^2*z^2
+ 61440*a^8*b*c^7*k^2*z^2 + 12288*a^7*b*c^8*h^2*z^2 + 12288*a^6*b*c^9*f^2*
z^2 + 61440*a^5*b*c^10*d^2*z^2 + 432*a*b^9*c^6*d^2*z^2 + 245760*a^9*c^7*k*m
*z^2 + 81920*a^8*c^8*f*m*z^2 - 49152*a^8*c^8*h*k*z^2 - 147456*a^7*c^9*d*k*z
^2 - 65536*a^7*c^9*e*j*z^2 - 16384*a^7*c^9*f*h*z^2 - 49152*a^6*c^10*d*f*z^2
+ 716800*a^8*b^3*c^5*m^2*z^2 - 483840*a^7*b^5*c^4*m^2*z^2 + 170496*a^6*b^7
*c^3*m^2*z^2 - 33232*a^5*b^9*c^2*m^2*z^2 + 516096*a^8*b^2*c^6*l^2*z^2 - 288
768*a^7*b^4*c^5*l^2*z^2 + 88576*a^6*b^6*c^4*l^2*z^2 - 15744*a^5*b^8*c^3*l^2
*z^2 + 1536*a^4*b^10*c^2*l^2*z^2 - 61440*a^7*b^3*c^6*k^2*z^2 + 24064*a^6*b^
5*c^5*k^2*z^2 - 4608*a^5*b^7*c^4*k^2*z^2 + 432*a^4*b^9*c^3*k^2*z^2 - 16*a^3
*b^11*c^2*k^2*z^2 + 24576*a^7*b^2*c^7*j^2*z^2 - 6144*a^6*b^4*c^6*j^2*z^2 +
512*a^5*b^6*c^5*j^2*z^2 - 8192*a^6*b^3*c^7*h^2*z^2 + 1536*a^5*b^5*c^6*h^2*z
^2 - 16*a^3*b^9*c^4*h^2*z^2 - 8192*a^6*b^2*c^8*g^2*z^2 + 6144*a^5*b^4*c^7*g
^2*z^2 - 1536*a^4*b^6*c^6*g^2*z^2 + 128*a^3*b^8*c^5*g^2*z^2 - 8192*a^5*b^3*
c^8*f^2*z^2 + 1536*a^4*b^5*c^7*f^2*z^2 - 16*a^2*b^9*c^5*f^2*z^2 + 24576*a^5
*b^2*c^9*e^2*z^2 - 6144*a^4*b^4*c^8*e^2*z^2 + 512*a^3*b^6*c^7*e^2*z^2 - 614
40*a^4*b^3*c^9*d^2*z^2 + 24064*a^3*b^5*c^8*d^2*z^2 - 4608*a^2*b^7*c^7*d^2*z
^2 - 393216*a^9*c^7*l^2*z^2 - 144*a^3*b^13*m^2*z^2 - 32768*a^8*c^8*j^2*z^2
- 32768*a^6*c^10*e^2*z^2 - 16*b^11*c^5*d^2*z^2 + 18432*a^8*b*c^5*h*l*m*z -
96*a^3*b^10*c*g*k*m*z + 90112*a^7*b*c^6*e*k*m*z + 36864*a^7*b*c^6*f*j*m*z -
16384*a^7*b*c^6*g*j*l*z + 14336*a^7*b*c^6*d*l*m*z - 10240*a^7*b*c^6*f*k*l*
z + 4096*a^7*b*c^6*h*j*k*z + 10240*a^7*b*c^6*g*h*m*z - 47104*a^6*b*c^7*d*h*
l*z + 36864*a^6*b*c^7*e*f*m*z + 30720*a^6*b*c^7*d*g*m*z - 16384*a^6*b*c^7*e
*g*l*z + 6144*a^6*b*c^7*f*g*k*z + 4096*a^6*b*c^7*e*h*k*z + 32*a*b^10*c^3*d*
f*l*z - 4096*a^5*b*c^8*d*f*j*z - 6144*a^5*b*c^8*d*g*h*z - 32*a*b^8*c^5*d*f*
g*z - 4096*a^4*b*c^9*d*e*f*z + 64*a*b^7*c^6*d*e*f*z + 110592*a^8*b^2*c^4*k*
l*m*z - 36864*a^7*b^4*c^3*k*l*m*z + 5376*a^6*b^6*c^2*k*l*m*z - 79872*a^7*b^
3*c^4*j*k*m*z + 26112*a^6*b^5*c^3*j*k*m*z - 3712*a^5*b^7*c^2*j*k*m*z - 1382
4*a^7*b^3*c^4*h*l*m*z + 3456*a^6*b^5*c^3*h*l*m*z - 288*a^5*b^7*c^2*h*l*m*z
- 45056*a^7*b^2*c^5*g*k*m*z + 39936*a^6*b^4*c^4*g*k*m*z + 30720*a^7*b^2*c^5
*f*l*m*z - 18432*a^7*b^2*c^5*h*k*l*z - 13056*a^5*b^6*c^3*g*k*m*z - 7680*a^6
*b^4*c^4*f*l*m*z + 5376*a^6*b^4*c^4*h*j*m*z + 4608*a^6*b^4*c^4*h*k*l*z + 30
72*a^7*b^2*c^5*h*j*m*z - 1984*a^5*b^6*c^3*h*j*m*z + 1856*a^4*b^8*c^2*g*k*m*
z + 640*a^5*b^6*c^3*f*l*m*z - 384*a^5*b^6*c^3*h*k*l*z + 192*a^4*b^8*c^2*h*j
*m*z - 79872*a^6*b^3*c^5*e*k*m*z - 27648*a^6*b^3*c^5*f*j*m*z + 26112*a^5*b^
5*c^4*e*k*m*z + 12288*a^6*b^3*c^5*g*j*l*z - 10752*a^6*b^3*c^5*d*l*m*z + 768
0*a^6*b^3*c^5*f*k*l*z + 6912*a^5*b^5*c^4*f*j*m*z - 3712*a^4*b^7*c^3*e*k*m*z
- 3072*a^6*b^3*c^5*h*j*k*z - 3072*a^5*b^5*c^4*g*j*l*z + 2688*a^5*b^5*c^4*d
*l*m*z - 1920*a^5*b^5*c^4*f*k*l*z + 768*a^5*b^5*c^4*h*j*k*z - 576*a^4*b^7*c
^3*f*j*m*z + 256*a^4*b^7*c^3*g*j*l*z - 224*a^4*b^7*c^3*d*l*m*z + 192*a^3*b^
9*c^2*e*k*m*z + 160*a^4*b^7*c^3*f*k*l*z - 64*a^4*b^7*c^3*h*j*k*z - 2688*a^5
*b^5*c^4*g*h*m*z - 1536*a^6*b^3*c^5*g*h*m*z + 992*a^4*b^7*c^3*g*h*m*z - 96*
a^3*b^9*c^2*g*h*m*z - 65536*a^6*b^2*c^6*d*k*l*z + 46080*a^6*b^2*c^6*d*j*m*z
- 24576*a^6*b^2*c^6*e*j*l*z + 21504*a^5*b^4*c^5*d*k*l*z - 11520*a^5*b^4*c^

5*d*j*m*z + 9216*a^6*b^2*c^6*f*j*k*z + 6144*a^5*b^4*c^5*e*j*l*z - 3072*a^4*
b^6*c^4*d*k*l*z - 2304*a^5*b^4*c^5*f*j*k*z + 960*a^4*b^6*c^4*d*j*m*z - 512*
a^4*b^6*c^4*e*j*l*z + 192*a^4*b^6*c^4*f*j*k*z + 160*a^3*b^8*c^3*d*k*l*z - 1
8432*a^6*b^2*c^6*f*g*m*z + 13824*a^5*b^4*c^5*f*g*m*z + 5376*a^5*b^4*c^5*e*h
*m*z - 3456*a^4*b^6*c^4*f*g*m*z + 3072*a^6*b^2*c^6*e*h*m*z - 3072*a^5*b^4*c
^5*f*h*l*z - 2048*a^6*b^2*c^6*g*h*k*z - 1984*a^4*b^6*c^4*e*h*m*z + 1536*a^5
*b^4*c^5*g*h*k*z + 1024*a^4*b^6*c^4*f*h*l*z - 384*a^4*b^6*c^4*g*h*k*z + 288
*a^3*b^8*c^3*f*g*m*z + 192*a^3*b^8*c^3*e*h*m*z - 96*a^3*b^8*c^3*f*h*l*z + 3
2*a^3*b^8*c^3*g*h*k*z + 41472*a^5*b^3*c^6*d*h*l*z - 27648*a^5*b^3*c^6*e*f*m
*z - 23040*a^5*b^3*c^6*d*g*m*z - 13440*a^4*b^5*c^5*d*h*l*z + 12288*a^5*b^3*
c^6*e*g*l*z + 6912*a^4*b^5*c^5*e*f*m*z + 5760*a^4*b^5*c^5*d*g*m*z - 4608*a^
5*b^3*c^6*f*g*k*z - 3072*a^5*b^3*c^6*e*h*k*z - 3072*a^4*b^5*c^5*e*g*l*z + 1
888*a^3*b^7*c^4*d*h*l*z + 1152*a^4*b^5*c^5*f*g*k*z + 768*a^4*b^5*c^5*e*h*k*
z - 576*a^3*b^7*c^4*e*f*m*z - 480*a^3*b^7*c^4*d*g*m*z + 256*a^3*b^7*c^4*e*g
*l*z - 96*a^3*b^7*c^4*f*g*k*z - 96*a^2*b^9*c^3*d*h*l*z - 64*a^3*b^7*c^4*e*h
*k*z + 46080*a^5*b^2*c^7*d*e*m*z - 11520*a^4*b^4*c^6*d*e*m*z + 9216*a^5*b^2
*c^7*e*f*k*z - 9216*a^5*b^2*c^7*d*h*j*z - 6656*a^4*b^4*c^6*d*f*l*z - 6144*a
^5*b^2*c^7*d*f*l*z + 3456*a^3*b^6*c^5*d*f*l*z - 2304*a^4*b^4*c^6*e*f*k*z +
2304*a^4*b^4*c^6*d*h*j*z + 960*a^3*b^6*c^5*d*e*m*z - 576*a^2*b^8*c^4*d*f*l*
z + 192*a^3*b^6*c^5*e*f*k*z - 192*a^3*b^6*c^5*d*h*j*z + 3072*a^4*b^3*c^7*d*
f*j*z - 768*a^3*b^5*c^6*d*f*j*z + 64*a^2*b^7*c^5*d*f*j*z + 4608*a^4*b^3*c^7
*d*g*h*z - 1152*a^3*b^5*c^6*d*g*h*z + 96*a^2*b^7*c^5*d*g*h*z - 9216*a^4*b^2
*c^8*d*e*h*z + 2304*a^3*b^4*c^7*d*e*h*z + 2048*a^4*b^2*c^8*d*f*g*z - 1536*a
^3*b^4*c^7*d*f*g*z + 384*a^2*b^6*c^6*d*f*g*z - 192*a^2*b^6*c^6*d*e*h*z + 30
72*a^3*b^3*c^8*d*e*f*z - 768*a^2*b^5*c^7*d*e*f*z - 288*a^5*b^8*c*k*l*m*z +
90112*a^8*b*c^5*j*k*m*z + 192*a^4*b^9*c*j*k*m*z + 138240*a^9*b*c^4*l*m^2*z
- 7344*a^6*b^7*c*l*m^2*z + 5088*a^5*b^8*c*j*m^2*z - 3072*a^8*b*c^5*k^2*l*z
- 49152*a^8*b*c^5*j*l^2*z - 128*a^4*b^9*c*j*l^2*z - 25600*a^8*b*c^5*g*m^2*z
- 9216*a^7*b*c^6*h^2*l*z - 2544*a^4*b^9*c*g*m^2*z + 64*a^3*b^10*c*g*l^2*z
+ 9216*a^7*b*c^6*g*k^2*z - 3072*a^6*b*c^7*f^2*l*z - 288*a^3*b^10*c*e*m^2*z
- 49152*a^7*b*c^6*e*l^2*z - 58368*a^5*b*c^8*d^2*l*z - 432*a*b^9*c^4*d^2*l*z
- 1024*a^6*b*c^7*g*h^2*z + 32*a*b^8*c^5*d^2*j*z + 1024*a^5*b*c^8*f^2*g*z -
9216*a^4*b*c^9*d^2*g*z + 336*a*b^7*c^6*d^2*g*z - 672*a*b^6*c^7*d^2*e*z - 1
22880*a^9*c^5*k*l*m*z - 40960*a^8*c^6*f*l*m*z + 24576*a^8*c^6*h*k*l*z - 204
80*a^8*c^6*h*j*m*z + 73728*a^7*c^7*d*k*l*z - 61440*a^7*c^7*d*j*m*z + 32768*
a^7*c^7*e*j*l*z - 12288*a^7*c^7*f*j*k*z - 20480*a^7*c^7*e*h*m*z + 8192*a^7*
c^7*f*h*l*z - 61440*a^6*c^8*d*e*m*z + 24576*a^6*c^8*d*f*l*z - 12288*a^6*c^8
*e*f*k*z + 12288*a^6*c^8*d*h*j*z + 12288*a^5*c^9*d*e*h*z - 131328*a^8*b^3*c
^3*l*m^2*z + 46656*a^7*b^5*c^2*l*m^2*z - 142848*a^8*b^2*c^4*j*m^2*z + 10636
8*a^7*b^4*c^3*j*m^2*z - 34208*a^6*b^6*c^2*j*m^2*z + 2304*a^7*b^3*c^4*k^2*l*
z - 576*a^6*b^5*c^3*k^2*l*z + 48*a^5*b^7*c^2*k^2*l*z + 45056*a^7*b^3*c^4*j*
l^2*z - 15360*a^6*b^5*c^3*j*l^2*z - 12288*a^7*b^2*c^5*j^2*l*z + 3072*a^6*b^
4*c^4*j^2*l*z + 2304*a^5*b^7*c^2*j*l^2*z - 256*a^5*b^6*c^3*j^2*l*z + 15872*
a^7*b^2*c^5*j*k^2*z - 4992*a^6*b^4*c^4*j*k^2*z + 672*a^5*b^6*c^3*j*k^2*z -
32*a^4*b^8*c^2*j*k^2*z + 71424*a^7*b^3*c^4*g*m^2*z - 53184*a^6*b^5*c^3*g*m^

2*z + 17104*a^5*b^7*c^2*g*m^2*z + 6912*a^6*b^3*c^5*h^2*l*z - 1728*a^5*b^5*c
^4*h^2*l*z + 144*a^4*b^7*c^3*h^2*l*z + 24576*a^7*b^2*c^5*g*l^2*z - 22528*a^
6*b^4*c^4*g*l^2*z + 7680*a^5*b^6*c^3*g*l^2*z + 4096*a^6*b^2*c^6*g^2*l*z - 3
072*a^5*b^4*c^5*g^2*l*z - 1152*a^4*b^8*c^2*g*l^2*z + 768*a^4*b^6*c^4*g^2*l*
z - 64*a^3*b^8*c^3*g^2*l*z - 142848*a^7*b^2*c^5*e*m^2*z + 106368*a^6*b^4*c^
4*e*m^2*z - 34208*a^5*b^6*c^3*e*m^2*z - 7936*a^6*b^3*c^5*g*k^2*z + 5088*a^4
*b^8*c^2*e*m^2*z + 2496*a^5*b^5*c^4*g*k^2*z - 1536*a^6*b^2*c^6*h^2*j*z + 12
80*a^5*b^3*c^6*f^2*l*z + 384*a^5*b^4*c^5*h^2*j*z - 336*a^4*b^7*c^3*g*k^2*z
+ 192*a^4*b^5*c^5*f^2*l*z - 144*a^3*b^7*c^4*f^2*l*z - 32*a^4*b^6*c^4*h^2*j*
z + 16*a^3*b^9*c^2*g*k^2*z + 16*a^2*b^9*c^3*f^2*l*z + 45056*a^6*b^3*c^5*e*l
^2*z - 15360*a^5*b^5*c^4*e*l^2*z - 12288*a^5*b^2*c^7*e^2*l*z + 3072*a^4*b^4
*c^6*e^2*l*z + 2304*a^4*b^7*c^3*e*l^2*z - 256*a^3*b^6*c^5*e^2*l*z - 128*a^3
*b^9*c^2*e*l^2*z + 59136*a^4*b^3*c^7*d^2*l*z - 23488*a^3*b^5*c^6*d^2*l*z +
15872*a^6*b^2*c^6*e*k^2*z - 4992*a^5*b^4*c^5*e*k^2*z + 4560*a^2*b^7*c^5*d^2
*l*z + 1536*a^5*b^2*c^7*f^2*j*z + 672*a^4*b^6*c^4*e*k^2*z - 384*a^4*b^4*c^6
*f^2*j*z - 32*a^3*b^8*c^3*e*k^2*z + 32*a^3*b^6*c^5*f^2*j*z + 768*a^5*b^3*c^
6*g*h^2*z - 192*a^4*b^5*c^5*g*h^2*z + 16*a^3*b^7*c^4*g*h^2*z - 15872*a^4*b^
2*c^8*d^2*j*z + 4992*a^3*b^4*c^7*d^2*j*z - 672*a^2*b^6*c^6*d^2*j*z - 1536*a
^5*b^2*c^7*e*h^2*z - 768*a^4*b^3*c^7*f^2*g*z + 384*a^4*b^4*c^6*e*h^2*z + 19
2*a^3*b^5*c^6*f^2*g*z - 32*a^3*b^6*c^5*e*h^2*z - 16*a^2*b^7*c^5*f^2*g*z + 7
936*a^3*b^3*c^8*d^2*g*z - 2496*a^2*b^5*c^7*d^2*g*z + 1536*a^4*b^2*c^8*e*f^2
*z - 384*a^3*b^4*c^7*e*f^2*z + 32*a^2*b^6*c^6*e*f^2*z - 15872*a^3*b^2*c^9*d
^2*e*z + 4992*a^2*b^4*c^8*d^2*e*z - 61440*a^8*b^2*c^4*l^3*z + 21504*a^7*b^4
*c^3*l^3*z - 3328*a^6*b^6*c^2*l^3*z + 432*a^5*b^9*l*m^2*z + 51200*a^9*c^5*j
*m^2*z + 16384*a^8*c^6*j^2*l*z - 288*a^4*b^10*j*m^2*z - 18432*a^8*c^6*j*k^2
*z + 144*a^3*b^11*g*m^2*z + 51200*a^8*c^6*e*m^2*z + 2048*a^7*c^7*h^2*j*z +
16384*a^6*c^8*e^2*l*z + 16*b^11*c^3*d^2*l*z - 18432*a^7*c^7*e*k^2*z - 2048*
a^6*c^8*f^2*j*z + 18432*a^5*c^9*d^2*j*z + 192*a^5*b^8*c*l^3*z + 2048*a^6*c^
8*e*h^2*z - 16*b^9*c^5*d^2*g*z - 2048*a^5*c^9*e*f^2*z + 32*b^8*c^6*d^2*e*z
+ 18432*a^4*c^10*d^2*e*z + 65536*a^9*c^5*l^3*z - 11008*a^8*b*c^3*j*k*l*m -
288*a^6*b^5*c*j*k*l*m + 144*a^5*b^6*c*g*k*l*m - 11008*a^7*b*c^4*e*k*l*m - 5
376*a^7*b*c^4*f*j*l*m + 3840*a^7*b*c^4*g*j*k*m - 3328*a^7*b*c^4*h*j*k*l - 9
6*a^4*b^7*c*g*j*k*m - 2560*a^7*b*c^4*g*h*l*m - 36*a^3*b^8*c*f*h*k*m - 6912*
a^6*b*c^5*d*j*k*l - 7872*a^6*b*c^5*d*h*k*m - 7680*a^6*b*c^5*d*g*l*m - 5376*
a^6*b*c^5*e*f*l*m + 3840*a^6*b*c^5*e*g*k*m - 3328*a^6*b*c^5*e*h*k*l - 1536*
a^6*b*c^5*f*g*k*l + 1280*a^6*b*c^5*f*g*j*m - 768*a^6*b*c^5*g*h*j*k - 768*a^
6*b*c^5*f*h*j*l - 768*a^6*b*c^5*e*h*j*m - 36*a^2*b^9*c*d*h*k*m - 6912*a^5*b
*c^6*d*e*k*l - 4864*a^5*b*c^6*d*e*j*m - 2304*a^5*b*c^6*d*g*j*k - 1792*a^5*b
*c^6*e*f*j*k - 1280*a^5*b*c^6*d*f*j*l - 4544*a^5*b*c^6*d*f*h*m + 1536*a^5*b
*c^6*d*g*h*l + 1280*a^5*b*c^6*e*f*g*m - 768*a^5*b*c^6*e*g*h*k - 768*a^5*b*c
^6*e*f*h*l - 256*a^5*b*c^6*f*g*h*j + 12*a*b^9*c^2*d*f*h*m + 16*a*b^8*c^3*d*
f*g*l - 4*a*b^8*c^3*d*f*h*k - 2304*a^4*b*c^7*d*e*g*k - 1792*a^4*b*c^7*d*e*h
*j - 1280*a^4*b*c^7*d*e*f*l - 768*a^4*b*c^7*d*f*g*j - 32*a*b^7*c^4*d*e*f*l
- 256*a^4*b*c^7*e*f*g*h - 768*a^3*b*c^8*d*e*f*g + 32*a*b^5*c^6*d*e*f*g + 12
*a*b^10*c*d*f*k*m + 3648*a^7*b^3*c^2*j*k*l*m + 5504*a^7*b^2*c^3*g*k*l*m - 1

824*a^6*b^4*c^2*g*k*l*m + 384*a^7*b^2*c^3*h*j*l*m - 288*a^6*b^4*c^2*h*j*l*m - 4800*a^6*b^3*c^3*g*j*k*m + 3648*a^6*b^3*c^3*e*k*l*m + 1280*a^5*b^5*c^2*g*j*k*m + 1088*a^6*b^3*c^3*f*j*l*m + 576*a^6*b^3*c^3*h*j*k*l - 288*a^5*b^5*c^2*e*k*l*m - 192*a^6*b^3*c^3*g*h*l*m + 144*a^5*b^5*c^2*g*h*l*m + 9600*a^6*b^2*c^4*e*j*k*m - 4224*a^6*b^2*c^4*d*j*l*m - 2560*a^5*b^4*c^3*e*j*k*m + 384*a^6*b^2*c^4*f*j*k*l + 224*a^5*b^4*c^3*d*j*l*m + 192*a^4*b^6*c^2*e*j*k*m - 160*a^5*b^4*c^3*f*j*k*l - 4608*a^6*b^2*c^4*f*h*k*m + 2688*a^6*b^2*c^4*f*g*l*m + 1664*a^6*b^2*c^4*g*h*k*l - 744*a^5*b^4*c^3*f*h*k*m - 544*a^5*b^4*c^3*f*g*l*m + 492*a^4*b^6*c^2*f*h*k*m + 416*a^5*b^4*c^3*g*h*j*m + 384*a^6*b^2*c^4*g*h*j*m + 384*a^6*b^2*c^4*e*h*l*m - 288*a^5*b^4*c^3*g*h*k*l - 288*a^5*b^4*c^3*e*h*l*m - 96*a^4*b^6*c^2*g*h*j*m + 2112*a^5*b^3*c^4*d*j*k*l - 160*a^4*b^5*c^3*d*j*k*l + 16992*a^5*b^3*c^4*d*h*k*m - 6252*a^4*b^5*c^3*d*h*k*m - 4800*a^5*b^3*c^4*e*g*k*m + 2112*a^5*b^3*c^4*d*g*l*m - 1728*a^5*b^3*c^4*f*g*j*m + 1280*a^4*b^5*c^3*e*g*k*m + 1088*a^5*b^3*c^4*e*f*l*m - 832*a^5*b^3*c^4*e*h*j*m + 816*a^3*b^7*c^2*d*h*k*m + 576*a^5*b^3*c^4*e*h*k*l - 448*a^5*b^3*c^4*f*h*j*l + 288*a^4*b^5*c^3*f*g*j*m - 192*a^5*b^3*c^4*g*h*j*k - 192*a^5*b^3*c^4*f*g*k*l + 192*a^4*b^5*c^3*e*h*j*m - 112*a^4*b^5*c^3*d*g*l*m + 96*a^4*b^5*c^3*f*h*j*l - 96*a^3*b^7*c^2*e*g*k*m + 80*a^4*b^5*c^3*f*g*k*l + 32*a^4*b^5*c^3*g*h*j*k - 11456*a^5*b^2*c^5*d*f*k*m + 4992*a^5*b^2*c^5*d*h*j*l - 4608*a^5*b^2*c^5*e*g*j*l - 4224*a^5*b^2*c^5*d*e*l*m + 3456*a^5*b^2*c^5*e*f*j*m + 3456*a^5*b^2*c^5*d*g*k*l + 2432*a^5*b^2*c^5*d*g*j*m - 1312*a^4*b^4*c^4*d*h*j*l + 1272*a^3*b^6*c^3*d*f*k*m - 1056*a^4*b^4*c^4*d*g*k*l + 896*a^5*b^2*c^5*f*g*j*k + 768*a^4*b^4*c^4*e*g*j*l - 576*a^4*b^4*c^4*e*f*j*m - 480*a^4*b^4*c^4*d*g*j*m + 384*a^5*b^2*c^5*e*h*j*k + 384*a^5*b^2*c^5*e*f*k*l - 232*a^2*b^8*c^2*d*f*k*m + 224*a^4*b^4*c^4*d*e*l*m - 160*a^4*b^4*c^4*e*f*k*l - 96*a^4*b^4*c^4*f*g*j*k + 96*a^3*b^6*c^3*d*h*j*l + 80*a^3*b^6*c^3*d*g*k*l - 64*a^4*b^4*c^4*e*h*j*k - 24*a^4*b^4*c^4*d*f*k*m + 416*a^4*b^4*c^4*e*g*h*m + 384*a^5*b^2*c^5*f*g*h*l + 384*a^5*b^2*c^5*e*g*h*m + 224*a^4*b^4*c^4*f*g*h*l - 96*a^3*b^6*c^3*e*g*h*m - 48*a^3*b^6*c^3*f*g*h*l + 2112*a^4*b^3*c^5*d*e*k*l - 960*a^4*b^3*c^5*d*f*j*l + 960*a^4*b^3*c^5*d*e*j*m + 384*a^3*b^5*c^4*d*f*j*l + 320*a^4*b^3*c^5*d*g*j*k + 192*a^4*b^3*c^5*e*f*j*k - 160*a^3*b^5*c^4*d*e*k*l - 32*a^2*b^7*c^3*d*f*j*l + 7392*a^4*b^3*c^5*d*f*h*m - 2496*a^4*b^3*c^5*d*g*h*l - 1728*a^4*b^3*c^5*e*f*g*m - 1500*a^3*b^5*c^4*d*f*h*m + 656*a^3*b^5*c^4*d*g*h*l - 448*a^4*b^3*c^5*e*f*h*l + 288*a^3*b^5*c^4*e*f*g*m - 192*a^4*b^3*c^5*f*g*h*j - 192*a^4*b^3*c^5*e*g*h*k + 96*a^3*b^5*c^4*e*f*h*l - 48*a^2*b^7*c^3*d*g*h*l + 32*a^3*b^5*c^4*e*g*h*k - 16*a^2*b^7*c^3*d*f*h*m - 640*a^4*b^2*c^6*d*e*j*k + 4992*a^4*b^2*c^6*d*e*h*l - 3584*a^4*b^2*c^6*d*f*h*k + 2432*a^4*b^2*c^6*d*e*g*m - 1312*a^3*b^4*c^5*d*e*h*l + 896*a^4*b^2*c^6*e*f*g*k + 896*a^4*b^2*c^6*d*g*h*j + 640*a^4*b^2*c^6*d*f*g*l + 600*a^3*b^4*c^5*d*f*h*k + 480*a^3*b^4*c^5*d*f*g*l - 480*a^3*b^4*c^5*d*e*g*m + 384*a^4*b^2*c^6*e*f*h*j - 192*a^2*b^6*c^4*d*f*g*l - 96*a^3*b^4*c^5*e*f*g*k - 96*a^3*b^4*c^5*d*g*h*j + 96*a^2*b^6*c^4*d*e*h*l + 12*a^2*b^6*c^4*d*f*h*k - 960*a^3*b^3*c^6*d*e*f*l + 384*a^2*b^5*c^5*d*e*f*l + 320*a^3*b^3*c^6*d*e*g*k - 192*a^3*b^3*c^6*d*f*g*j + 192*a^3*b^3*c^6*d*e*h*j + 32*a^2*b^5*c^5*d*f*g*j - 192*a^3*b^3*c^6*e*f*g*h + 384*a^3*b^2*c^7*d*e*f*j - 64*a^2*b^4*c^6*d*e*f*j + 896*a

^3*b^2*c^7*d*e*g*h - 96*a^2*b^4*c^6*d*e*g*h - 192*a^2*b^3*c^7*d*e*f*g + 496
*a^7*b^4*c*k*l^2*m - 4752*a^7*b^4*c*j*l*m^2 + 96*a^5*b^6*c*j^2*k*m - 6144*a
^8*b*c^3*h*l^2*m - 168*a^6*b^5*c*h*l^2*m + 6400*a^8*b*c^3*g*l*m^2 - 2862*a^
6*b^5*c*h*k*m^2 + 2376*a^6*b^5*c*g*l*m^2 - 1632*a^7*b*c^4*h^2*k*m - 480*a^8
*b*c^3*h*k*m^2 - 180*a^5*b^6*c*h*k^2*m + 54*a^4*b^7*c*h^2*k*m - 384*a^7*b*c
^4*h*j^2*m + 120*a^5*b^6*c*h*k*l^2 + 56*a^5*b^6*c*f*l^2*m + 24*a^3*b^8*c*g^
2*k*m + 4512*a^7*b*c^4*f*k^2*m - 2304*a^7*b*c^4*g*k^2*l - 1680*a^5*b^6*c*g*
j*m^2 + 1184*a^6*b*c^5*f^2*k*m + 804*a^5*b^6*c*f*k*m^2 + 432*a^5*b^6*c*e*l*
m^2 + 60*a^4*b^7*c*f*k^2*m + 6*a^2*b^9*c*f^2*k*m - 13312*a^7*b*c^4*d*l^2*m
+ 2048*a^7*b*c^4*g*j*l^2 - 1024*a^7*b*c^4*f*k*l^2 + 64*a^4*b^7*c*g*j*l^2 +
56*a^4*b^7*c*d*l^2*m - 40*a^4*b^7*c*f*k*l^2 + 13440*a^7*b*c^4*e*j*m^2 - 892
8*a^5*b*c^6*d^2*k*m - 6240*a^7*b*c^4*d*k*m^2 + 1614*a^4*b^7*c*d*k*m^2 - 288
*a^4*b^7*c*e*j*m^2 - 170*a*b^9*c^2*d^2*k*m + 60*a^3*b^8*c*d*k^2*m + 4608*a^
6*b*c^5*e*j^2*l + 4608*a^5*b*c^6*e^2*j*l - 2432*a^6*b*c^5*d*j^2*m + 1440*a^
7*b*c^4*f*h*m^2 - 896*a^6*b*c^5*f*j^2*k - 864*a^6*b*c^5*f*h^2*m - 558*a^4*b
^7*c*f*h*m^2 + 256*a^6*b*c^5*g*h^2*l - 40*a^3*b^8*c*d*k*l^2 - 1920*a^6*b*c^
5*e*j*k^2 - 384*a^5*b*c^6*e^2*h*m + 24*a^3*b^8*c*f*h*l^2 - 16*a*b^8*c^3*d^2
*j*l + 2208*a^6*b*c^5*f*h*k^2 - 1044*a^3*b^8*c*d*h*m^2 + 800*a^5*b*c^6*f^2*
h*k - 256*a^5*b*c^6*f^2*g*l + 144*a^3*b^8*c*e*g*m^2 - 116*a*b^8*c^3*d^2*h*m
+ 8192*a^6*b*c^5*d*h*l^2 + 2048*a^6*b*c^5*e*g*l^2 + 24*a^2*b^9*c*d*h*l^2 -
5856*a^4*b*c^7*d^2*f*m + 4896*a^4*b*c^7*d^2*h*k + 2720*a^6*b*c^5*d*f*m^2 +
2304*a^4*b*c^7*d^2*g*l + 1824*a^5*b*c^6*d*h^2*k + 438*a*b^7*c^4*d^2*f*m -
384*a^5*b*c^6*e*h^2*j + 318*a^2*b^9*c*d*f*m^2 - 168*a*b^7*c^4*d^2*g*l + 42*
a*b^7*c^4*d^2*h*k - 36*a*b^8*c^3*d*f^2*m - 2432*a^4*b*c^7*d*e^2*m + 1536*a^
5*b*c^6*e*g*j^2 + 1536*a^4*b*c^7*e^2*g*j - 896*a^5*b*c^6*d*h*j^2 - 896*a^4*
b*c^7*e^2*f*k + 4896*a^5*b*c^6*d*f*k^2 + 1824*a^4*b*c^7*d*f^2*k - 384*a^4*b
*c^7*e*f^2*j + 336*a*b^6*c^5*d^2*e*l - 156*a*b^6*c^5*d^2*f*k + 16*a*b^6*c^5
*d^2*g*j + 12*a*b^7*c^4*d*f^2*k - 2*a*b^9*c^2*d*f*k^2 - 1920*a^3*b*c^8*d^2*
e*j - 32*a*b^5*c^6*d^2*e*j + 2208*a^3*b*c^8*d^2*f*h + 800*a^4*b*c^7*d*f*h^2
- 102*a*b^5*c^6*d^2*f*h + 12*a*b^6*c^5*d*f^2*h - 2*a*b^7*c^4*d*f*h^2 - 896
*a^3*b*c^8*d*e^2*h - 8*a*b^6*c^5*d*f*g^2 - 240*a*b^4*c^7*d^2*e*g - 32*a*b^4
*c^7*d*e^2*f + 5120*a^8*c^4*h*j*l*m + 15360*a^7*c^5*d*j*l*m - 7680*a^7*c^5*
e*j*k*m + 3072*a^7*c^5*f*j*k*l + 5120*a^7*c^5*e*h*l*m + 1920*a^7*c^5*f*h*k*
m + 15360*a^6*c^6*d*e*l*m + 5760*a^6*c^6*d*f*k*m + 3072*a^6*c^6*e*f*k*l - 3
072*a^6*c^6*d*h*j*l - 2560*a^6*c^6*e*f*j*m + 1536*a^6*c^6*e*h*j*k + 4608*a^
5*c^7*d*e*j*k - 3072*a^5*c^7*d*e*h*l - 1152*a^5*c^7*d*f*h*k + 512*a^5*c^7*e
*f*h*j + 1536*a^4*c^8*d*e*f*j - 8*a*b^10*c*d*f*l^2 - 5568*a^8*b^2*c^2*k*l^2
*m + 15552*a^8*b^2*c^2*j*l*m^2 + 4800*a^7*b^2*c^3*j^2*k*m - 1280*a^6*b^4*c^
2*j^2*k*m + 2080*a^7*b^3*c^2*h*l^2*m - 1088*a^7*b^2*c^3*j*k^2*l + 48*a^6*b^
4*c^2*j*k^2*l - 8544*a^7*b^2*c^3*h*k^2*m - 7776*a^7*b^3*c^2*g*l*m^2 + 7632*
a^7*b^3*c^2*h*k*m^2 + 3600*a^6*b^3*c^3*h^2*k*m + 2484*a^6*b^4*c^2*h*k^2*m -
918*a^5*b^5*c^2*h^2*k*m + 4800*a^7*b^2*c^3*h*k*l^2 - 1424*a^6*b^4*c^2*h*k*
l^2 + 1200*a^5*b^4*c^3*g^2*k*m - 960*a^6*b^2*c^4*g^2*k*m - 528*a^6*b^4*c^2*
f*l^2*m - 416*a^6*b^3*c^3*h*j^2*m - 320*a^4*b^6*c^2*g^2*k*m + 192*a^7*b^2*c
^3*f*l^2*m + 96*a^5*b^5*c^2*h*j^2*m + 15552*a^7*b^2*c^3*e*l*m^2 - 6720*a^7*

b^2*c^3*g*j*m^2 + 6160*a^6*b^4*c^2*g*j*m^2 - 4752*a^6*b^4*c^2*e*l*m^2 - 201
6*a^7*b^2*c^3*f*k*m^2 - 1164*a^6*b^4*c^2*f*k*m^2 + 1104*a^5*b^3*c^4*f^2*k*m
 + 1008*a^6*b^3*c^3*f*k^2*m + 960*a^6*b^2*c^4*h^2*j*l - 678*a^5*b^5*c^2*f*k
^2*m + 544*a^6*b^3*c^3*g*k^2*l - 144*a^5*b^4*c^3*h^2*j*l - 102*a^4*b^5*c^3*
f^2*k*m - 62*a^3*b^7*c^2*f^2*k*m - 24*a^5*b^5*c^2*g*k^2*l + 6432*a^6*b^3*c^
3*d*l^2*m + 4800*a^5*b^2*c^5*e^2*k*m - 2304*a^6*b^2*c^4*g*j^2*l + 1920*a^6*
b^3*c^3*g*j*l^2 + 1728*a^6*b^2*c^4*f*j^2*m - 1280*a^4*b^4*c^4*e^2*k*m + 115
2*a^5*b^3*c^4*g^2*j*l - 1032*a^5*b^5*c^2*d*l^2*m - 864*a^6*b^3*c^3*f*k*l^2
 - 768*a^5*b^5*c^2*g*j*l^2 + 408*a^5*b^5*c^2*f*k*l^2 + 384*a^5*b^4*c^3*g*j^2
*l - 288*a^5*b^4*c^3*f*j^2*m + 192*a^6*b^2*c^4*h*j^2*k - 192*a^4*b^5*c^3*g^
2*j*l + 96*a^3*b^6*c^3*e^2*k*m - 32*a^5*b^4*c^3*h*j^2*k - 21120*a^6*b^2*c^4
*d*k^2*m + 20880*a^6*b^3*c^3*d*k*m^2 + 19760*a^4*b^3*c^5*d^2*k*m - 12320*a^
6*b^3*c^3*e*j*m^2 - 9750*a^5*b^5*c^2*d*k*m^2 - 9390*a^3*b^5*c^4*d^2*k*m + 8
460*a^5*b^4*c^3*d*k^2*m + 3360*a^5*b^5*c^2*e*j*m^2 + 1860*a^2*b^7*c^3*d^2*k
*m - 1218*a^4*b^6*c^2*d*k^2*m - 1088*a^6*b^2*c^4*e*k^2*l + 960*a^6*b^2*c^4*
g*j*k^2 - 240*a^5*b^4*c^3*g*j*k^2 + 192*a^5*b^2*c^5*f^2*j*l - 104*a^4*b^5*c
^3*g^2*h*m - 96*a^5*b^3*c^4*g^2*h*m + 48*a^5*b^4*c^3*e*k^2*l + 48*a^4*b^4*c
^4*f^2*j*l + 24*a^3*b^7*c^2*g^2*h*m + 16*a^4*b^6*c^2*g*j*k^2 - 16*a^3*b^6*c
^3*f^2*j*l + 13376*a^6*b^2*c^4*d*k*l^2 - 5136*a^5*b^4*c^3*d*k*l^2 - 3840*a^
6*b^2*c^4*e*j*l^2 + 1536*a^5*b^4*c^3*e*j*l^2 + 1392*a^5*b^3*c^4*f*h^2*m + 1
386*a^5*b^5*c^2*f*h*m^2 - 768*a^5*b^3*c^4*e*j^2*l + 768*a^4*b^6*c^2*d*k*l^2
 - 768*a^4*b^3*c^5*e^2*j*l - 588*a^4*b^4*c^4*f^2*h*m - 480*a^5*b^3*c^4*g*h^
2*l + 480*a^5*b^3*c^4*d*j^2*m - 480*a^5*b^2*c^5*f^2*h*m - 128*a^4*b^6*c^2*e
*j*l^2 + 100*a^3*b^6*c^3*f^2*h*m + 96*a^5*b^3*c^4*f*j^2*k + 72*a^4*b^5*c^3*
g*h^2*l - 54*a^4*b^5*c^3*f*h^2*m - 48*a^6*b^3*c^3*f*h*m^2 - 36*a^3*b^7*c^2*
f*h^2*m + 6*a^2*b^8*c^2*f^2*h*m + 6848*a^4*b^2*c^6*d^2*j*l - 2448*a^3*b^4*c
^5*d^2*j*l + 624*a^5*b^4*c^3*f*h*l^2 + 576*a^6*b^2*c^4*f*h*l^2 + 480*a^5*b^
3*c^4*e*j*k^2 + 432*a^4*b^4*c^4*f*g^2*m - 416*a^4*b^3*c^5*e^2*h*m + 336*a^2
*b^6*c^4*d^2*j*l - 320*a^5*b^2*c^5*f*g^2*m - 256*a^4*b^6*c^2*f*h*l^2 + 192*
a^5*b^2*c^5*g^2*h*k + 96*a^3*b^5*c^4*e^2*h*m - 72*a^3*b^6*c^3*f*g^2*m + 48*
a^4*b^4*c^4*g^2*h*k - 32*a^4*b^5*c^3*e*j*k^2 - 8*a^3*b^6*c^3*g^2*h*k + 2476
8*a^6*b^2*c^4*d*h*m^2 - 21108*a^5*b^4*c^3*d*h*m^2 - 10048*a^4*b^2*c^6*d^2*h
*m + 7218*a^4*b^6*c^2*d*h*m^2 - 6720*a^6*b^2*c^4*e*g*m^2 + 6160*a^5*b^4*c^3
*e*g*m^2 - 2592*a^5*b^2*c^5*d*h^2*m - 1680*a^4*b^6*c^2*e*g*m^2 + 1068*a^3*b
^4*c^5*d^2*h*m + 960*a^5*b^2*c^5*e*h^2*l - 876*a^4*b^4*c^4*d*h^2*m - 864*a^
5*b^2*c^5*f*h^2*k + 546*a^2*b^6*c^4*d^2*h*m + 432*a^3*b^6*c^3*d*h^2*m + 336
*a^4*b^3*c^5*f^2*h*k - 320*a^5*b^2*c^5*d*j^2*k + 192*a^5*b^2*c^5*g*h^2*j +
144*a^5*b^3*c^4*f*h*k^2 - 144*a^4*b^4*c^4*e*h^2*l - 102*a^4*b^5*c^3*f*h*k^2
 - 96*a^4*b^3*c^5*f^2*g*l - 36*a^2*b^8*c^2*d*h^2*m - 30*a^3*b^5*c^4*f^2*h*k
 - 24*a^3*b^5*c^4*f^2*g*l + 16*a^4*b^4*c^4*g*h^2*j - 12*a^4*b^4*c^4*f*h^2*k
 + 12*a^3*b^6*c^3*f*h^2*k + 8*a^2*b^7*c^3*f^2*g*l + 6*a^3*b^7*c^2*f*h*k^2 -
 2*a^2*b^7*c^3*f^2*h*k - 9312*a^5*b^3*c^4*d*h*l^2 + 3288*a^4*b^5*c^3*d*h*l^
2 - 2304*a^4*b^2*c^6*e^2*g*l + 1920*a^5*b^3*c^4*e*g*l^2 + 1728*a^4*b^2*c^6*
e^2*f*m + 1152*a^4*b^3*c^5*e*g^2*l - 768*a^4*b^5*c^3*e*g*l^2 - 608*a^4*b^3*
c^5*d*g^2*m - 472*a^3*b^7*c^2*d*h*l^2 + 384*a^3*b^4*c^5*e^2*g*l - 288*a^3*b

^4*c^5*e^2*f*m - 224*a^4*b^3*c^5*f*g^2*k + 192*a^5*b^2*c^5*f*h*j^2 + 192*a^
4*b^2*c^6*e^2*h*k - 192*a^3*b^5*c^4*e*g^2*l + 120*a^3*b^5*c^4*d*g^2*m + 64*
a^3*b^7*c^2*e*g*l^2 - 32*a^3*b^4*c^5*e^2*h*k + 24*a^3*b^5*c^4*f*g^2*k + 993
6*a^3*b^3*c^6*d^2*f*m + 3786*a^4*b^5*c^3*d*f*m^2 - 3552*a^5*b^2*c^5*d*h*k^2
 - 3486*a^2*b^5*c^5*d^2*f*m - 3424*a^3*b^3*c^6*d^2*g*l - 1868*a^3*b^7*c^2*d
*f*m^2 + 1332*a^4*b^4*c^4*d*h*k^2 - 1296*a^5*b^3*c^4*d*f*m^2 - 1236*a^3*b^4
*c^5*d*f^2*m + 1224*a^2*b^5*c^5*d^2*g*l - 1152*a^4*b^2*c^6*d*f^2*m + 960*a^
5*b^2*c^5*e*g*k^2 - 496*a^3*b^3*c^6*d^2*h*k + 462*a^2*b^6*c^4*d*f^2*m + 432
*a^4*b^3*c^5*d*h^2*k - 240*a^4*b^4*c^4*e*g*k^2 - 222*a^2*b^5*c^5*d^2*h*k +
192*a^4*b^2*c^6*f^2*g*j + 192*a^4*b^2*c^6*e*f^2*l - 174*a^3*b^5*c^4*d*h^2*k
 - 156*a^3*b^6*c^3*d*h*k^2 + 48*a^3*b^4*c^5*e*f^2*l - 32*a^4*b^3*c^5*e*h^2*
j + 16*a^3*b^6*c^3*e*g*k^2 + 16*a^3*b^4*c^5*f^2*g*j - 16*a^2*b^6*c^4*e*f^2*
l + 12*a^2*b^7*c^3*d*h^2*k + 6*a^2*b^8*c^2*d*h*k^2 + 1728*a^5*b^2*c^5*d*f*l
^2 + 1392*a^4*b^4*c^4*d*f*l^2 - 840*a^3*b^6*c^3*d*f*l^2 - 768*a^4*b^2*c^6*e
*g^2*j + 576*a^4*b^2*c^6*d*g^2*k + 480*a^3*b^3*c^6*d*e^2*m + 144*a^2*b^8*c^
2*d*f*l^2 + 96*a^4*b^3*c^5*d*h*j^2 + 96*a^3*b^3*c^6*e^2*f*k - 80*a^3*b^4*c^
5*d*g^2*k + 6848*a^3*b^2*c^7*d^2*e*l - 3552*a^3*b^2*c^7*d^2*f*k - 2448*a^2*
b^4*c^6*d^2*e*l + 1332*a^2*b^4*c^6*d^2*f*k + 960*a^3*b^2*c^7*d^2*g*j - 496*
a^4*b^3*c^5*d*f*k^2 + 432*a^3*b^3*c^6*d*f^2*k - 240*a^2*b^4*c^6*d^2*g*j - 2
22*a^3*b^5*c^4*d*f*k^2 - 174*a^2*b^5*c^5*d*f^2*k + 64*a^4*b^2*c^6*f*g^2*h +
 48*a^3*b^4*c^5*f*g^2*h + 42*a^2*b^7*c^3*d*f*k^2 - 32*a^3*b^3*c^6*e*f^2*j -
 320*a^3*b^2*c^7*d*e^2*k + 192*a^4*b^2*c^6*e*g*h^2 + 192*a^4*b^2*c^6*d*f*j^
2 - 32*a^3*b^4*c^5*d*f*j^2 + 16*a^3*b^4*c^5*e*g*h^2 + 480*a^2*b^3*c^7*d^2*e
*j - 224*a^3*b^3*c^6*d*g^2*h + 192*a^3*b^2*c^7*e^2*f*h + 24*a^2*b^5*c^5*d*g
^2*h - 864*a^3*b^2*c^7*d*f^2*h + 336*a^3*b^3*c^6*d*f*h^2 + 192*a^3*b^2*c^7*
e*f^2*g + 144*a^2*b^3*c^7*d^2*f*h - 30*a^2*b^5*c^5*d*f*h^2 + 16*a^2*b^4*c^6
*e*f^2*g - 12*a^2*b^4*c^6*d*f^2*h + 192*a^3*b^2*c^7*d*f*g^2 + 96*a^2*b^3*c^
7*d*e^2*h + 48*a^2*b^4*c^6*d*f*g^2 + 960*a^2*b^2*c^8*d^2*e*g + 192*a^2*b^2*
c^8*d*e^2*f - 7680*a^9*b*c^2*l^2*m^2 + 3152*a^8*b^3*c*l^2*m^2 + 2070*a^7*b^
4*c*k^2*m^2 - 1840*a^7*b^3*c^2*k^3*m + 6720*a^8*b*c^3*j^2*m^2 - 3072*a^8*b*
c^3*k^2*l^2 + 1680*a^6*b^5*c*j^2*m^2 - 100*a^6*b^5*c*k^2*l^2 - 2176*a^7*b^3
*c^2*j*l^3 - 256*a^6*b^3*c^3*j^3*l - 64*a^5*b^6*c*j^2*l^2 - 12480*a^8*b^2*c
^2*h*m^3 + 972*a^5*b^6*c*h^2*m^2 - 960*a^7*b*c^4*j^2*k^2 - 252*a^5*b^4*c^3*
h^3*m - 192*a^6*b^2*c^4*h^3*m + 54*a^4*b^6*c^2*h^3*m + 1536*a^7*b*c^4*h^2*l
^2 + 420*a^4*b^7*c*g^2*m^2 - 36*a^4*b^7*c*h^2*l^2 - 3072*a^7*b^2*c^3*g*l^3
+ 2096*a^7*b^3*c^2*f*m^3 + 1088*a^6*b^4*c^2*g*l^3 - 496*a^6*b^3*c^3*h*k^3 -
 192*a^4*b^4*c^4*g^3*l + 176*a^4*b^3*c^5*f^3*m + 144*a^5*b^3*c^4*h^3*k + 78
*a^3*b^8*c*f^2*m^2 + 54*a^3*b^5*c^4*f^3*m + 32*a^3*b^6*c^3*g^3*l + 30*a^5*b
^5*c^2*h*k^3 - 18*a^4*b^5*c^3*h^3*k - 18*a^2*b^7*c^3*f^3*m - 16*a^3*b^8*c*g
^2*l^2 + 6720*a^6*b*c^5*e^2*m^2 - 192*a^6*b*c^5*h^2*j^2 - 4*a^2*b^9*c*f^2*l
^2 - 35040*a^7*b^2*c^3*d*m^3 + 14300*a^6*b^4*c^2*d*m^3 - 12000*a^3*b^2*c^7*
d^3*m + 4380*a^2*b^4*c^6*d^3*m - 2176*a^6*b^3*c^3*e*l^3 - 256*a^3*b^3*c^6*e
^3*l - 192*a^6*b^2*c^4*f*k^3 + 192*a^5*b^5*c^2*e*l^3 - 192*a^4*b^2*c^6*f^3*
k + 132*a^5*b^4*c^3*f*k^3 + 128*a^4*b^3*c^5*g^3*j - 28*a^3*b^4*c^5*f^3*k -
10*a^4*b^6*c^2*f*k^3 + 6*a^2*b^6*c^4*f^3*k + 10752*a^5*b*c^6*d^2*l^2 - 960*

a^5*b*c^6*e^2*k^2 - 192*a^5*b*c^6*f^2*j^2 + 108*a*b^9*c^2*d^2*l^2 - 1680*a^
5*b^3*c^4*d*k^3 - 1680*a^2*b^3*c^7*d^3*k + 222*a^4*b^5*c^3*d*k^3 + 30*a*b^8
*c^3*d^2*k^2 - 10*a^3*b^7*c^2*d*k^3 - 960*a^4*b*c^7*d^2*j^2 + 80*a^4*b^3*c^
5*f*h^3 + 80*a^3*b^3*c^6*f^3*h + 6*a^3*b^5*c^4*f*h^3 + 6*a^2*b^5*c^5*f^3*h
- 192*a^4*b*c^7*e^2*h^2 - 192*a^4*b^2*c^6*d*h^3 - 192*a^2*b^2*c^8*d^3*h + 1
28*a^3*b^3*c^6*e*g^3 - 28*a^3*b^4*c^5*d*h^3 + 12*a*b^6*c^5*d^2*h^2 + 6*a^2*
b^6*c^4*d*h^3 - 192*a^3*b*c^8*e^2*f^2 + 60*a*b^5*c^6*d^2*g^2 + 198*a*b^4*c^
7*d^2*f^2 + 144*a^2*b^3*c^7*d*f^3 - 960*a^2*b*c^9*d^2*e^2 + 240*a*b^3*c^8*d
^2*e^2 + 15360*a^9*c^3*k*l^2*m - 12800*a^9*c^3*j*l*m^2 - 3840*a^8*c^4*j^2*k
*m + 432*a^6*b^6*j*l*m^2 + 4608*a^8*c^4*j*k^2*l + 2880*a^8*c^4*h*k^2*m + 51
20*a^8*c^4*f*l^2*m - 3072*a^8*c^4*h*k*l^2 + 270*a^5*b^7*h*k*m^2 - 216*a^5*b
^7*g*l*m^2 - 12800*a^8*c^4*e*l*m^2 - 4800*a^8*c^4*f*k*m^2 - 512*a^7*c^5*h^2
*j*l - 3840*a^6*c^6*e^2*k*m - 1280*a^7*c^5*f*j^2*m + 768*a^7*c^5*h*j^2*k +
144*a^4*b^8*g*j*m^2 - 90*a^4*b^8*f*k*m^2 + 8640*a^7*c^5*d*k^2*m + 4608*a^7*
c^5*e*k^2*l + 512*a^6*c^6*f^2*j*l - 9216*a^7*c^5*d*k*l^2 - 4096*a^7*c^5*e*j
*l^2 + 320*a^6*c^6*f^2*h*m - 90*a^3*b^9*d*k*m^2 + 15200*a^9*b*c^2*k*m^3 - 6
192*a^8*b^3*c*k*m^3 + 5472*a^8*b*c^3*k^3*m - 4608*a^5*c^7*d^2*j*l - 1024*a^
7*c^5*f*h*l^2 + 150*a^6*b^5*c*k^3*m + 54*a^3*b^9*f*h*m^2 + 6*b^10*c^2*d^2*h
*m - 14400*a^7*c^5*d*h*m^2 + 8640*a^5*c^7*d^2*h*m + 2880*a^6*c^6*d*h^2*m +
2304*a^6*c^6*d*j^2*k - 512*a^6*c^6*e*h^2*l - 192*a^6*c^6*f*h^2*k + 6144*a^8
*b*c^3*j*l^3 + 1536*a^7*b*c^4*j^3*l - 1280*a^5*c^7*e^2*f*m + 768*a^5*c^7*e^
2*h*k + 256*a^6*c^6*f*h*j^2 + 192*a^6*b^5*c*j*l^3 + 54*a^2*b^10*d*h*m^2 - 1
8*b^9*c^3*d^2*f*m + 8*b^9*c^3*d^2*g*l - 2*b^9*c^3*d^2*h*k + 4068*a^7*b^4*c*
h*m^3 - 1728*a^6*c^6*d*h*k^2 + 960*a^5*c^7*d*f^2*m + 512*a^5*c^7*e*f^2*l -
3072*a^6*c^6*d*f*l^2 - 16*b^8*c^4*d^2*e*l + 6*b^8*c^4*d^2*f*k - 4608*a^4*c^
8*d^2*e*l + 2400*a^8*b*c^3*f*m^3 + 2016*a^7*b*c^4*h*k^3 - 1728*a^4*c^8*d^2*
f*k - 1146*a^6*b^5*c*f*m^3 + 224*a^6*b*c^5*h^3*k - 96*a^5*b^6*c*g*l^3 + 96*
a^5*b*c^6*f^3*m + 2304*a^4*c^8*d*e^2*k + 768*a^5*c^7*d*f*j^2 + 6144*a^7*b*c
^4*e*l^3 - 2280*a^5*b^6*c*d*m^3 + 1536*a^4*b*c^7*e^3*l - 616*a*b^6*c^5*d^3*
m + 512*a^6*b*c^5*g*j^3 + 256*a^4*c^8*e^2*f*h + 240*a*b^10*c*d^2*m^2 + 6*b^
7*c^5*d^2*f*h - 192*a^4*c^8*d*f^2*h + 4320*a^6*b*c^5*d*k^3 + 4320*a^3*b*c^8
*d^3*k + 222*a*b^5*c^6*d^3*k + 16*b^6*c^6*d^2*e*g + 96*a^5*b*c^6*f*h^3 + 96
*a^4*b*c^7*f^3*h + 768*a^3*c^9*d*e^2*f + 512*a^3*b*c^8*e^3*g + 132*a*b^4*c^
7*d^3*h + 2016*a^2*b*c^9*d^3*f - 496*a*b^3*c^8*d^3*f + 224*a^3*b*c^8*d*f^3
- 18*a*b^5*c^6*d*f^3 - 3264*a^8*b^2*c^2*k^2*m^2 - 6160*a^7*b^3*c^2*j^2*m^2
+ 1104*a^7*b^3*c^2*k^2*l^2 - 1920*a^7*b^2*c^3*j^2*l^2 + 768*a^6*b^4*c^2*j^2
*l^2 + 3888*a^7*b^2*c^3*h^2*m^2 - 3510*a^6*b^4*c^2*h^2*m^2 + 240*a^6*b^3*c^
3*j^2*k^2 - 16*a^5*b^5*c^2*j^2*k^2 + 1680*a^6*b^3*c^3*g^2*m^2 - 1648*a^6*b^
3*c^3*h^2*l^2 - 1540*a^5*b^5*c^2*g^2*m^2 + 444*a^5*b^5*c^2*h^2*l^2 - 960*a^
6*b^2*c^4*h^2*k^2 - 576*a^6*b^2*c^4*f^2*m^2 - 512*a^6*b^2*c^4*g^2*l^2 - 480
*a^5*b^4*c^3*g^2*l^2 + 198*a^5*b^4*c^3*h^2*k^2 + 192*a^4*b^6*c^2*g^2*l^2 -
186*a^5*b^4*c^3*f^2*m^2 - 97*a^4*b^6*c^2*f^2*m^2 - 9*a^4*b^6*c^2*h^2*k^2 -
6160*a^5*b^3*c^4*e^2*m^2 + 1680*a^4*b^5*c^3*e^2*m^2 - 240*a^5*b^3*c^4*g^2*k
^2 - 240*a^5*b^3*c^4*f^2*l^2 - 144*a^3*b^7*c^2*e^2*m^2 + 60*a^4*b^5*c^3*g^2
*k^2 - 36*a^4*b^5*c^3*f^2*l^2 + 36*a^3*b^7*c^2*f^2*l^2 - 16*a^5*b^3*c^4*h^2

*j^2 - 4*a^3*b^7*c^2*g^2*k^2 + 38512*a^5*b^2*c^5*d^2*m^2 - 32310*a^4*b^4*c^
4*d^2*m^2 + 12720*a^3*b^6*c^3*d^2*m^2 - 2500*a^2*b^8*c^2*d^2*m^2 - 1920*a^5
*b^2*c^5*e^2*l^2 + 768*a^4*b^4*c^4*e^2*l^2 - 464*a^5*b^2*c^5*f^2*k^2 - 384*
a^5*b^2*c^5*g^2*j^2 - 64*a^3*b^6*c^3*e^2*l^2 + 42*a^4*b^4*c^4*f^2*k^2 + 12*
a^3*b^6*c^3*f^2*k^2 - 13104*a^4*b^3*c^5*d^2*l^2 + 5628*a^3*b^5*c^4*d^2*l^2
- 1128*a^2*b^7*c^3*d^2*l^2 + 240*a^4*b^3*c^5*e^2*k^2 - 16*a^4*b^3*c^5*f^2*j
^2 - 16*a^3*b^5*c^4*e^2*k^2 - 2880*a^4*b^2*c^6*d^2*k^2 + 1750*a^3*b^4*c^5*d
^2*k^2 - 345*a^2*b^6*c^4*d^2*k^2 - 48*a^4*b^3*c^5*g^2*h^2 - 4*a^3*b^5*c^4*g
^2*h^2 + 240*a^3*b^3*c^6*d^2*j^2 - 192*a^4*b^2*c^6*f^2*h^2 - 42*a^3*b^4*c^5
*f^2*h^2 - 16*a^2*b^5*c^5*d^2*j^2 - 48*a^3*b^3*c^6*f^2*g^2 - 16*a^3*b^3*c^6
*e^2*h^2 - 4*a^2*b^5*c^5*f^2*g^2 - 464*a^3*b^2*c^7*d^2*h^2 - 384*a^3*b^2*c^
7*e^2*g^2 + 42*a^2*b^4*c^6*d^2*h^2 - 240*a^2*b^3*c^7*d^2*g^2 - 16*a^2*b^3*c
^7*e^2*f^2 - 960*a^2*b^2*c^8*d^2*f^2 + 6*b^11*c*d^2*k*m - 18*a*b^11*d*f*m^2
 - 7200*a^9*c^3*k^2*m^2 - 324*a^7*b^5*l^2*m^2 - 225*a^6*b^6*k^2*m^2 - 2048*
a^8*c^4*j^2*l^2 - 144*a^5*b^7*j^2*m^2 - 2400*a^8*c^4*h^2*m^2 - 81*a^4*b^8*h
^2*m^2 - 800*a^7*c^5*f^2*m^2 - 288*a^7*c^5*h^2*k^2 - 36*a^3*b^9*g^2*m^2 - 9
*a^2*b^10*f^2*m^2 - 21600*a^6*c^6*d^2*m^2 - 2048*a^6*c^6*e^2*l^2 - 864*a^6*
c^6*f^2*k^2 - 2592*a^5*c^7*d^2*k^2 - 1536*a^5*c^7*e^2*j^2 + 1536*a^8*b^2*c^
2*l^4 - 32*a^5*c^7*f^2*h^2 + 360*a^7*b^2*c^3*k^4 - 25*a^6*b^4*c^2*k^4 - 864
*a^4*c^8*d^2*h^2 - 4*b^7*c^5*d^2*g^2 - 9*b^6*c^6*d^2*f^2 - 288*a^3*c^9*d^2*
f^2 - 24*a^5*b^2*c^5*h^4 - 16*b^5*c^7*d^2*e^2 - 9*a^4*b^4*c^4*h^4 - 16*a^3*
b^4*c^5*g^4 - 24*a^3*b^2*c^7*f^4 - 9*a^2*b^4*c^6*f^4 - a^2*b^8*c^2*f^2*k^2
- a^2*b^6*c^4*f^2*h^2 + 630*a^7*b^5*k*m^3 + 8000*a^9*c^3*h*m^3 + 320*a^7*c^
5*h^3*m - 378*a^6*b^6*h*m^3 + 126*a^5*b^7*f*m^3 + 30*b^8*c^4*d^3*m + 24000*
a^8*c^4*d*m^3 + 8640*a^4*c^8*d^3*m - 1728*a^7*c^5*f*k^3 - 192*a^5*c^7*f^3*k
 - 4*b^11*c*d^2*l^2 + 126*a^4*b^8*d*m^3 - 10*b^7*c^5*d^3*k + 4200*a^9*b^2*c
*m^4 - 1024*a^6*c^6*e*j^3 - 1024*a^4*c^8*e^3*j - 144*a^7*b^4*c*l^4 - 10*b^6
*c^6*d^3*h - 1728*a^3*c^9*d^3*h - 192*a^5*c^7*d*h^3 + 30*b^5*c^7*d^3*f + 36
0*a*b^2*c^9*d^4 - 9*b^12*d^2*m^2 - 10000*a^10*c^2*m^4 - 4096*a^9*c^3*l^4 -
441*a^8*b^4*m^4 - 1296*a^8*c^4*k^4 - 256*a^7*c^5*j^4 - 16*a^6*c^6*h^4 - 16*
a^4*c^8*f^4 - 256*a^3*c^9*e^4 - 25*b^4*c^8*d^4 - 1296*a^2*c^10*d^4 - b^10*c
^2*d^2*k^2 - b^8*c^4*d^2*h^2, z, k1)*x*(8192*a^6*b*c^9 + 32*a^2*b^9*c^5 - 5
12*a^3*b^7*c^6 + 3072*a^4*b^5*c^7 - 8192*a^5*b^3*c^8))/(4*(64*a^5*c^6 - a^2
*b^6*c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5))) + (x*(2*b^6*c^6*d^2 - 576*a^3
*c^9*d^2 + 64*a^4*c^8*f^2 - 64*a^5*c^7*h^2 + 576*a^6*c^6*k^2 + 18*a^2*b^10*
m^2 - 1600*a^7*c^5*m^2 - 36*a*b^4*c^7*d^2 + 128*a^3*b*c^8*e^2 + 128*a^5*b*c
^6*j^2 + 8*a^2*b^9*c*l^2 + 3072*a^6*b*c^5*l^2 - 300*a^3*b^8*c*m^2 + 256*a^2
*b^2*c^8*d^2 - 32*a^2*b^3*c^7*e^2 + 20*a^2*b^4*c^6*f^2 - 96*a^3*b^2*c^7*f^2
 - 8*a^2*b^5*c^5*g^2 + 32*a^3*b^3*c^6*g^2 + 2*a^2*b^6*c^4*h^2 - 4*a^3*b^4*c
^5*h^2 - 32*a^4*b^3*c^5*j^2 + 2*a^2*b^8*c^2*k^2 - 40*a^3*b^6*c^3*k^2 + 276*
a^4*b^4*c^4*k^2 - 736*a^5*b^2*c^5*k^2 - 136*a^3*b^7*c^2*l^2 + 888*a^4*b^5*c
^3*l^2 - 2656*a^5*b^3*c^4*l^2 + 1874*a^4*b^6*c^2*m^2 - 5284*a^5*b^4*c^3*m^2
 + 6144*a^6*b^2*c^4*m^2 - 384*a^4*c^8*d*h + 1920*a^5*c^7*d*m - 1024*a^5*c^7
*e*l + 384*a^5*c^7*f*k + 640*a^6*c^6*h*m - 1024*a^6*c^6*j*l + 4*a*b^5*c^6*d
*f + 320*a^3*b*c^8*d*f + 64*a^4*b*c^7*f*h + 576*a^4*b*c^7*d*k + 256*a^4*b*c

^7*e*j - 1472*a^5*b*c^6*f*m + 512*a^5*b*c^6*g*l + 64*a^5*b*c^6*h*k - 12*a^2
*b^9*c*k*m - 3776*a^6*b*c^5*k*m - 96*a^2*b^3*c^7*d*f + 8*a^2*b^4*c^6*d*h +
32*a^2*b^4*c^6*e*g + 64*a^3*b^2*c^7*d*h - 128*a^3*b^2*c^7*e*g - 12*a^2*b^5*
c^5*f*h + 32*a^3*b^3*c^6*f*h + 20*a^2*b^5*c^5*d*k - 224*a^3*b^3*c^6*d*k - 6
4*a^3*b^3*c^6*e*j - 60*a^2*b^6*c^4*d*m - 12*a^2*b^6*c^4*f*k + 632*a^3*b^4*c
^5*d*m - 32*a^3*b^4*c^5*e*l + 152*a^3*b^4*c^5*f*k + 32*a^3*b^4*c^5*g*j - 20
48*a^4*b^2*c^6*d*m + 384*a^4*b^2*c^6*e*l - 512*a^4*b^2*c^6*f*k - 128*a^4*b^
2*c^6*g*j + 36*a^2*b^7*c^3*f*m + 4*a^2*b^7*c^3*h*k - 396*a^3*b^5*c^4*f*m +
16*a^3*b^5*c^4*g*l - 44*a^3*b^5*c^4*h*k + 1376*a^4*b^3*c^5*f*m - 192*a^4*b^
3*c^5*g*l + 96*a^4*b^3*c^5*h*k - 12*a^2*b^8*c^2*h*m + 112*a^3*b^6*c^3*h*m -
248*a^4*b^4*c^4*h*m - 192*a^5*b^2*c^5*h*m - 32*a^4*b^4*c^4*j*l + 384*a^5*b
^2*c^5*j*l + 220*a^3*b^7*c^2*k*m - 1436*a^4*b^5*c^3*k*m + 3936*a^5*b^3*c^4*
k*m))/(4*(64*a^5*c^6 - a^2*b^6*c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5))) - (
5*b^3*c^7*d^3 + 8*a^3*c^7*f^3 + 216*a^6*c^4*k^3 - 63*a^5*b^5*m^3 - 96*a^2*c
^8*d*e^2 + 72*a^2*c^8*d^2*f - 4*a^4*b*c^5*h^3 - 3*b^4*c^6*d^2*f - 32*a^3*c^
7*e^2*h + b^5*c^5*d^2*h - 96*a^4*c^6*d*j^2 + 8*a^4*c^6*f*h^2 + 216*a^3*c^7*
d^2*k + 573*a^6*b^3*c*m^3 - 1300*a^7*b*c^2*m^3 + 384*a^5*c^5*d*l^2 + b^6*c^
4*d^2*k + 72*a^4*c^6*f^2*k + 216*a^5*c^5*f*k^2 + 9*a^2*b^8*f*m^2 + 160*a^4*
c^6*e^2*m - 32*a^5*c^5*h*j^2 - 3*b^7*c^3*d^2*m + 24*a^5*c^5*h^2*k + 200*a^6
*c^4*f*m^2 - 27*a^3*b^7*h*m^2 + 128*a^6*c^4*h*l^2 + 45*a^4*b^6*k*m^2 + 160*
a^6*c^4*j^2*m + 600*a^7*c^3*k*m^2 - 640*a^7*c^3*l^2*m + 6*a^2*b^2*c^6*f^3 -
3*a^3*b^3*c^4*h^3 + 5*a^4*b^4*c^2*k^3 - 66*a^5*b^2*c^3*k^3 - 36*a*b*c^8*d^
3 + 9*a*b^9*d*m^2 + 4*a*b^8*c*d*l^2 + 48*a^3*c^7*d*f*h - 192*a^3*c^7*d*e*j
- 240*a^4*c^6*d*f*m + 144*a^4*c^6*d*h*k - 128*a^4*c^6*e*f*l - 64*a^4*c^6*e*
h*j - 80*a^5*c^5*f*h*m - 720*a^5*c^5*d*k*m + 320*a^5*c^5*e*j*m - 384*a^5*c^
5*e*k*l - 128*a^5*c^5*f*j*l - 240*a^6*c^4*h*k*m - 384*a^6*c^4*j*k*l + 16*a*
b^2*c^7*d*e^2 + 18*a*b^2*c^7*d^2*f + 3*a*b^3*c^6*d*f^2 - 60*a^2*b*c^7*d*f^2
+ 4*a*b^4*c^5*d*g^2 + 16*a^2*b*c^7*e^2*f - a*b^3*c^6*d^2*h + a*b^5*c^4*d*h
^2 - 60*a^2*b*c^7*d^2*h - 28*a^3*b*c^6*d*h^2 - 28*a^3*b*c^6*f^2*h - 10*a*b^
4*c^5*d^2*k + a*b^7*c^2*d*k^2 - 396*a^4*b*c^5*d*k^2 + 16*a^3*b*c^6*e^2*k +
16*a^4*b*c^5*f*j^2 + 25*a*b^5*c^4*d^2*m - 159*a^2*b^7*c*d*m^2 - 348*a^3*b*c
^6*d^2*m + 1460*a^5*b*c^4*d*m^2 + 4*a^2*b^7*c*f*l^2 + 128*a^5*b*c^4*f*l^2 -
78*a^3*b^6*c*f*m^2 - 76*a^4*b*c^5*f^2*m - 204*a^5*b*c^4*h*k^2 - 12*a^3*b^6
*c*h*l^2 + 279*a^4*b^5*c*h*m^2 - 12*a^5*b*c^4*h^2*m + 16*a^5*b*c^4*j^2*k +
420*a^6*b*c^3*h*m^2 + 20*a^4*b^5*c*k*l^2 + 512*a^6*b*c^3*k*l^2 - 30*a^4*b^5
*c*k^2*m - 402*a^5*b^4*c*k*m^2 - 924*a^6*b*c^3*k^2*m - 28*a^5*b^4*c*l^2*m -
24*a^2*b^2*c^6*d*g^2 - 9*a^2*b^3*c^5*d*h^2 + 4*a^2*b^3*c^5*f*g^2 - 5*a^2*b
^3*c^5*f^2*h + a^2*b^4*c^4*f*h^2 + 16*a^3*b^2*c^5*d*j^2 + 18*a^3*b^2*c^5*f*
h^2 - 6*a^2*b^2*c^6*d^2*k - 21*a^2*b^5*c^3*d*k^2 - 8*a^3*b^2*c^5*g^2*h + 15
5*a^3*b^3*c^4*d*k^2 - 72*a^2*b^6*c^2*d*l^2 + 436*a^3*b^4*c^3*d*l^2 - 952*a^
4*b^2*c^4*d*l^2 + 23*a^2*b^3*c^5*d^2*m - 5*a^2*b^4*c^4*f^2*k + a^2*b^6*c^2*
f*k^2 + 26*a^3*b^2*c^5*f^2*k - 12*a^3*b^4*c^3*f*k^2 + 970*a^3*b^5*c^2*d*m^2
+ 2*a^4*b^2*c^4*f*k^2 - 2289*a^4*b^3*c^3*d*m^2 - 48*a^3*b^2*c^5*e^2*m + 4*
a^3*b^3*c^4*g^2*k - 36*a^3*b^5*c^2*f*l^2 + 52*a^4*b^3*c^3*f*l^2 + 15*a^2*b^
5*c^3*f^2*m - 53*a^3*b^3*c^4*f^2*m - 6*a^3*b^4*c^3*h^2*k - 3*a^3*b^5*c^2*h*

k^2 + 42*a^4*b^2*c^4*h^2*k + 51*a^4*b^3*c^3*h*k^2 + 133*a^4*b^4*c^2*f*m^2 +
 114*a^5*b^2*c^3*f*m^2 - 12*a^3*b^4*c^3*g^2*m + 40*a^4*b^2*c^4*g^2*m + 128*
a^4*b^4*c^2*h*l^2 - 360*a^5*b^2*c^3*h*l^2 + 18*a^3*b^5*c^2*h^2*m - 81*a^4*b
^3*c^3*h^2*m - 801*a^5*b^3*c^2*h*m^2 - 48*a^5*b^2*c^3*j^2*m - 204*a^5*b^3*c
^2*k*l^2 + 339*a^5*b^3*c^2*k^2*m + 762*a^6*b^2*c^2*k*m^2 + 264*a^6*b^2*c^2*
l^2*m - 6*a*b^8*c*d*k*m - 16*a*b^3*c^6*d*e*g + 96*a^2*b*c^7*d*e*g - 4*a*b^4
*c^5*d*f*h + 32*a^3*b*c^6*e*g*h + 16*a*b^5*c^4*d*e*l - 4*a*b^5*c^4*d*f*k +
544*a^3*b*c^6*d*e*l - 312*a^3*b*c^6*d*f*k + 96*a^3*b*c^6*d*g*j + 32*a^3*b*c
^6*e*f*j + 12*a*b^6*c^3*d*f*m - 8*a*b^6*c^3*d*g*l + 2*a*b^6*c^3*d*h*k - 6*a
*b^7*c^2*d*h*m - 152*a^4*b*c^5*d*h*m - 160*a^4*b*c^5*e*g*m + 224*a^4*b*c^5*
e*h*l + 64*a^4*b*c^5*f*g*l - 152*a^4*b*c^5*f*h*k + 32*a^4*b*c^5*g*h*j + 544
*a^4*b*c^5*d*j*l + 32*a^4*b*c^5*e*j*k - 6*a^2*b^7*c*f*k*m + 32*a^5*b*c^4*e*
l*m - 536*a^5*b*c^4*f*k*m - 160*a^5*b*c^4*g*j*m + 192*a^5*b*c^4*g*k*l + 224
*a^5*b*c^4*h*j*l + 18*a^3*b^6*c*h*k*m + 32*a^6*b*c^3*j*l*m + 52*a^2*b^2*c^6
*d*f*h - 16*a^2*b^2*c^6*e*f*g + 32*a^2*b^2*c^6*d*e*j - 192*a^2*b^3*c^5*d*e*
l + 70*a^2*b^3*c^5*d*f*k - 16*a^2*b^3*c^5*d*g*j - 190*a^2*b^4*c^4*d*f*m + 9
6*a^2*b^4*c^4*d*g*l - 30*a^2*b^4*c^4*d*h*k + 16*a^2*b^4*c^4*e*f*l + 676*a^3
*b^2*c^5*d*f*m - 272*a^3*b^2*c^5*d*g*l + 100*a^3*b^2*c^5*d*h*k - 48*a^3*b^2
*c^5*e*f*l - 16*a^3*b^2*c^5*e*g*k - 16*a^3*b^2*c^5*f*g*j + 80*a^2*b^5*c^3*d
*h*m - 8*a^2*b^5*c^3*f*g*l + 2*a^2*b^5*c^3*f*h*k - 210*a^3*b^3*c^4*d*h*m +
48*a^3*b^3*c^4*e*g*m - 48*a^3*b^3*c^4*e*h*l + 24*a^3*b^3*c^4*f*g*l + 6*a^3*
b^3*c^4*f*h*k + 16*a^2*b^5*c^3*d*j*l - 192*a^3*b^3*c^4*d*j*l - 6*a^2*b^6*c^
2*f*h*m - 28*a^3*b^4*c^3*f*h*m + 24*a^3*b^4*c^3*g*h*l + 276*a^4*b^2*c^4*f*h
*m - 112*a^4*b^2*c^4*g*h*l + 116*a^2*b^6*c^2*d*k*m - 780*a^3*b^4*c^3*d*k*m
+ 16*a^3*b^4*c^3*f*j*l + 1876*a^4*b^2*c^4*d*k*m - 96*a^4*b^2*c^4*e*j*m + 80
*a^4*b^2*c^4*e*k*l - 48*a^4*b^2*c^4*f*j*l - 16*a^4*b^2*c^4*g*j*k + 62*a^3*b
^5*c^2*f*k*m - 42*a^4*b^3*c^3*f*k*m + 48*a^4*b^3*c^3*g*j*m - 40*a^4*b^3*c^3
*g*k*l - 48*a^4*b^3*c^3*h*j*l - 246*a^4*b^4*c^2*h*k*m - 16*a^5*b^2*c^3*g*l*
m + 804*a^5*b^2*c^3*h*k*m + 80*a^5*b^2*c^3*j*k*l)/(8*(64*a^5*c^6 - a^2*b^6*
c^3 + 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5)) + (x*(32*a^2*c^8*e^3 + 32*a^5*c^5*j
^3 - 2*b^3*c^7*d^2*e + b^4*c^6*d^2*g - 12*a^4*b^5*c*l^3 - 320*a^6*b*c^3*l^3
 + 96*a^3*c^7*e^2*j + 96*a^4*c^6*e*j^2 + 144*a^3*c^7*d^2*l + 128*a^5*c^5*e*
l^2 - b^6*c^4*d^2*l - 16*a^4*c^6*f^2*l - 9*a^2*b^8*g*m^2 + 16*a^5*c^5*h^2*l
 + 18*a^3*b^7*j*m^2 + 128*a^6*c^4*j*l^2 - 144*a^6*c^4*k^2*l - 27*a^4*b^6*l*
m^2 + 400*a^7*c^3*l*m^2 - 4*a^2*b^3*c^5*g^3 + 124*a^5*b^3*c^2*l^3 + 24*a*b*
c^8*d^2*e - 48*a^2*c^8*d*e*f - 16*a^3*c^7*e*f*h - 144*a^3*c^7*d*e*k - 48*a^
3*c^7*d*f*j + 96*a^4*c^6*d*h*l + 80*a^4*c^6*e*f*m - 48*a^4*c^6*e*h*k - 16*a
^4*c^6*f*h*j - 144*a^4*c^6*d*j*k - 480*a^5*c^5*d*l*m + 240*a^5*c^5*e*k*m +
80*a^5*c^5*f*j*m - 96*a^5*c^5*f*k*l - 48*a^5*c^5*h*j*k - 160*a^6*c^4*h*l*m
+ 240*a^6*c^4*j*k*m - 12*a*b^2*c^7*d^2*g + 16*a^2*b*c^7*e*f^2 - 48*a^2*b*c^
7*e^2*g + 8*a^3*b*c^6*e*h^2 - 2*a*b^3*c^6*d^2*j + 24*a^2*b*c^7*d^2*j + 18*a
*b^4*c^5*d^2*l + 16*a^3*b*c^6*f^2*j + 96*a^4*b*c^5*e*k^2 - 176*a^3*b*c^6*e^
2*l - 48*a^4*b*c^5*g*j^2 + 18*a^2*b^7*c*e*m^2 + 8*a^4*b*c^5*h^2*j - 520*a^5
*b*c^4*e*m^2 - 4*a^2*b^7*c*g*l^2 - 64*a^5*b*c^4*g*l^2 + 96*a^3*b^6*c*g*m^2
+ 96*a^5*b*c^4*j*k^2 + 8*a^3*b^6*c*j*l^2 - 176*a^5*b*c^4*j^2*l - 192*a^4*b^

5*c*j*m^2 - 520*a^6*b*c^3*j*m^2 + 270*a^5*b^4*c*l*m^2 + 24*a^2*b^2*c^6*e*g^
2 - 8*a^2*b^2*c^6*f^2*g + 2*a^2*b^3*c^5*e*h^2 - a^2*b^4*c^4*g*h^2 - 4*a^3*b
^2*c^5*g*h^2 - 100*a^2*b^2*c^6*d^2*l + 2*a^2*b^5*c^3*e*k^2 - 28*a^3*b^3*c^4
*e*k^2 + 32*a^2*b^3*c^5*e^2*l + 8*a^2*b^6*c^2*e*l^2 + 24*a^3*b^2*c^5*g^2*j
- 88*a^3*b^4*c^3*e*l^2 + 216*a^4*b^2*c^4*e*l^2 - a^2*b^4*c^4*f^2*l - a^2*b^
6*c^2*g*k^2 + 2*a^3*b^3*c^4*h^2*j + 14*a^3*b^4*c^3*g*k^2 - 192*a^3*b^5*c^2*
e*m^2 - 48*a^4*b^2*c^4*g*k^2 + 614*a^4*b^3*c^3*e*m^2 + 8*a^2*b^5*c^3*g^2*l
- 44*a^3*b^3*c^4*g^2*l + 44*a^3*b^5*c^2*g*l^2 - 108*a^4*b^3*c^3*g*l^2 - 12*
a^4*b^2*c^4*h^2*l - 307*a^4*b^4*c^2*g*m^2 + 260*a^5*b^2*c^3*g*m^2 + 2*a^3*b
^5*c^2*j*k^2 - 28*a^4*b^3*c^3*j*k^2 + 32*a^4*b^3*c^3*j^2*l - 88*a^4*b^4*c^2
*j*l^2 + 216*a^5*b^2*c^3*j*l^2 - 3*a^4*b^4*c^2*k^2*l + 40*a^5*b^2*c^3*k^2*l
 + 614*a^5*b^3*c^2*j*m^2 - 756*a^6*b^2*c^2*l*m^2 - 4*a*b^2*c^7*d*e*f + 2*a*
b^3*c^6*d*f*g + 32*a^2*b*c^7*d*e*h + 24*a^2*b*c^7*d*f*g + 8*a^3*b*c^6*f*g*h
 - 2*a*b^5*c^4*d*f*l + 272*a^3*b*c^6*d*e*m - 8*a^3*b*c^6*d*f*l + 72*a^3*b*c
^6*d*g*k + 32*a^3*b*c^6*d*h*j + 80*a^3*b*c^6*e*f*k - 96*a^3*b*c^6*e*g*j + 6
4*a^4*b*c^5*e*h*m - 40*a^4*b*c^5*f*g*m + 8*a^4*b*c^5*f*h*l + 24*a^4*b*c^5*g
*h*k + 272*a^4*b*c^5*d*j*m + 72*a^4*b*c^5*d*k*l - 352*a^4*b*c^5*e*j*l + 80*
a^4*b*c^5*f*j*k + 6*a^2*b^7*c*g*k*m + 248*a^5*b*c^4*f*l*m - 120*a^5*b*c^4*g
*k*m + 64*a^5*b*c^4*h*j*m + 56*a^5*b*c^4*h*k*l - 12*a^3*b^6*c*j*k*m + 18*a^
4*b^5*c*k*l*m + 584*a^6*b*c^3*k*l*m - 16*a^2*b^2*c^6*d*g*h - 12*a^2*b^2*c^6
*e*f*h + 20*a^2*b^2*c^6*d*e*k - 4*a^2*b^2*c^6*d*f*j + 6*a^2*b^3*c^5*f*g*h -
 60*a^2*b^3*c^5*d*e*m + 18*a^2*b^3*c^5*d*f*l - 10*a^2*b^3*c^5*d*g*k - 12*a^
2*b^3*c^5*e*f*k + 30*a^2*b^4*c^4*d*g*m + 6*a^2*b^4*c^4*d*h*l + 36*a^2*b^4*c
^4*e*f*m - 32*a^2*b^4*c^4*e*g*l + 4*a^2*b^4*c^4*e*h*k + 6*a^2*b^4*c^4*f*g*k
 - 136*a^3*b^2*c^5*d*g*m - 64*a^3*b^2*c^5*d*h*l - 180*a^3*b^2*c^5*e*f*m + 1
76*a^3*b^2*c^5*e*g*l - 20*a^3*b^2*c^5*e*h*k - 40*a^3*b^2*c^5*f*g*k - 12*a^3
*b^2*c^5*f*h*j + 20*a^3*b^2*c^5*d*j*k - 12*a^2*b^5*c^3*e*h*m - 18*a^2*b^5*c
^3*f*g*m - 2*a^2*b^5*c^3*g*h*k + 40*a^3*b^3*c^4*e*h*m + 90*a^3*b^3*c^4*f*g*
m + 6*a^3*b^3*c^4*f*h*l + 10*a^3*b^3*c^4*g*h*k - 60*a^3*b^3*c^4*d*j*m - 10*
a^3*b^3*c^4*d*k*l + 64*a^3*b^3*c^4*e*j*l - 12*a^3*b^3*c^4*f*j*k + 6*a^2*b^6
*c^2*g*h*m - 20*a^3*b^4*c^3*g*h*m - 32*a^4*b^2*c^4*g*h*m - 12*a^2*b^6*c^2*e
*k*m + 148*a^3*b^4*c^3*e*k*m + 36*a^3*b^4*c^3*f*j*m - 32*a^3*b^4*c^3*g*j*l
+ 4*a^3*b^4*c^3*h*j*k + 104*a^4*b^2*c^4*d*l*m - 476*a^4*b^2*c^4*e*k*m - 180
*a^4*b^2*c^4*f*j*m + 8*a^4*b^2*c^4*f*k*l + 176*a^4*b^2*c^4*g*j*l - 20*a^4*b
^2*c^4*h*j*k - 74*a^3*b^5*c^2*g*k*m - 12*a^3*b^5*c^2*h*j*m - 54*a^4*b^3*c^3
*f*l*m + 238*a^4*b^3*c^3*g*k*m + 40*a^4*b^3*c^3*h*j*m - 6*a^4*b^3*c^3*h*k*l
 + 18*a^4*b^4*c^2*h*l*m - 48*a^5*b^2*c^3*h*l*m + 148*a^4*b^4*c^2*j*k*m - 47
6*a^5*b^2*c^3*j*k*m - 210*a^5*b^3*c^2*k*l*m))/(4*(64*a^5*c^6 - a^2*b^6*c^3
+ 12*a^3*b^4*c^4 - 48*a^4*b^2*c^5)))*root(1572864*a^8*b^2*c^10*z^4 - 983040
*a^7*b^4*c^9*z^4 + 327680*a^6*b^6*c^8*z^4 - 61440*a^5*b^8*c^7*z^4 + 6144*a^
4*b^10*c^6*z^4 - 256*a^3*b^12*c^5*z^4 - 1048576*a^9*c^11*z^4 - 1572864*a^8*
b^2*c^8*l*z^3 + 983040*a^7*b^4*c^7*l*z^3 - 327680*a^6*b^6*c^6*l*z^3 + 61440
*a^5*b^8*c^5*l*z^3 - 6144*a^4*b^10*c^4*l*z^3 + 256*a^3*b^12*c^3*l*z^3 + 104
8576*a^9*c^9*l*z^3 + 96*a^3*b^12*c*k*m*z^2 + 98304*a^8*b*c^7*j*l*z^2 + 2457
6*a^8*b*c^7*h*m*z^2 + 155648*a^7*b*c^8*d*m*z^2 + 98304*a^7*b*c^8*e*l*z^2 +

57344*a^7*b*c^8*f*k*z^2 + 32768*a^7*b*c^8*g*j*z^2 + 57344*a^6*b*c^9*d*h*z^2 + 32768*a^6*b*c^9*e*g*z^2 - 32*a*b^10*c^5*d*f*z^2 - 491520*a^8*b^2*c^6*k*m*z^2 + 358400*a^7*b^4*c^5*k*m*z^2 - 129024*a^6*b^6*c^4*k*m*z^2 + 24768*a^5*b^8*c^3*k*m*z^2 - 2432*a^4*b^10*c^2*k*m*z^2 - 90112*a^7*b^3*c^6*j*l*z^2 + 30720*a^6*b^5*c^5*j*l*z^2 - 4608*a^5*b^7*c^4*j*l*z^2 + 256*a^4*b^9*c^3*j*l*z^2 - 21504*a^6*b^5*c^5*h*m*z^2 + 9216*a^5*b^7*c^4*h*m*z^2 + 8192*a^7*b^3*c^6*h*m*z^2 - 1568*a^4*b^9*c^3*h*m*z^2 + 96*a^3*b^11*c^2*h*m*z^2 - 172032*a^7*b^2*c^7*f*m*z^2 + 116736*a^6*b^4*c^6*f*m*z^2 - 49152*a^7*b^2*c^7*g*l*z^2 + 45056*a^6*b^4*c^6*g*l*z^2 - 35840*a^5*b^6*c^5*f*m*z^2 + 24576*a^7*b^2*c^7*h*k*z^2 - 15360*a^5*b^6*c^5*g*l*z^2 + 5184*a^4*b^8*c^4*f*m*z^2 - 3072*a^5*b^6*c^5*h*k*z^2 + 2304*a^4*b^8*c^4*g*l*z^2 + 2048*a^6*b^4*c^6*h*k*z^2 + 576*a^4*b^8*c^4*h*k*z^2 - 288*a^3*b^10*c^3*f*m*z^2 - 128*a^3*b^10*c^3*g*l*z^2 - 32*a^3*b^10*c^3*h*k*z^2 - 147456*a^6*b^3*c^7*d*m*z^2 - 90112*a^6*b^3*c^7*e*l*z^2 + 52224*a^5*b^5*c^6*d*m*z^2 - 49152*a^6*b^3*c^7*f*k*z^2 + 30720*a^5*b^5*c^6*e*l*z^2 - 24576*a^6*b^3*c^7*g*j*z^2 + 15360*a^5*b^5*c^6*f*k*z^2 - 8192*a^4*b^7*c^5*d*m*z^2 + 6144*a^5*b^5*c^6*g*j*z^2 - 4608*a^4*b^7*c^5*e*l*z^2 - 2048*a^4*b^7*c^5*f*k*z^2 - 512*a^4*b^7*c^5*g*j*z^2 + 480*a^3*b^9*c^4*d*m*z^2 + 256*a^3*b^9*c^4*e*l*z^2 + 96*a^3*b^9*c^4*f*k*z^2 + 131072*a^6*b^2*c^8*d*k*z^2 + 49152*a^6*b^2*c^8*e*j*z^2 - 43008*a^5*b^4*c^7*d*k*z^2 - 12288*a^5*b^4*c^7*e*j*z^2 + 6144*a^4*b^6*c^6*d*k*z^2 + 1024*a^4*b^6*c^6*e*j*z^2 - 320*a^3*b^8*c^5*d*k*z^2 + 6144*a^5*b^4*c^7*f*h*z^2 - 2048*a^4*b^6*c^6*f*h*z^2 + 192*a^3*b^8*c^5*f*h*z^2 - 49152*a^5*b^3*c^8*d*h*z^2 - 24576*a^5*b^3*c^8*e*g*z^2 + 15360*a^4*b^5*c^7*d*h*z^2 + 6144*a^4*b^5*c^7*e*g*z^2 - 2048*a^3*b^7*c^6*d*h*z^2 - 512*a^3*b^7*c^6*e*g*z^2 + 96*a^2*b^9*c^5*d*h*z^2 + 24576*a^5*b^2*c^9*d*f*z^2 - 3072*a^3*b^6*c^7*d*f*z^2 + 2048*a^4*b^4*c^8*d*f*z^2 + 576*a^2*b^8*c^6*d*f*z^2 - 430080*a^9*b*c^6*m^2*z^2 + 3408*a^4*b^11*c*m^2*z^2 - 64*a^3*b^12*c*l^2*z^2 + 61440*a^8*b*c^7*k^2*z^2 + 12288*a^7*b*c^8*h^2*z^2 + 12288*a^6*b*c^9*f^2*z^2 + 61440*a^5*b*c^10*d^2*z^2 + 432*a*b^9*c^6*d^2*z^2 + 245760*a^9*c^7*k*m*z^2 + 81920*a^8*c^8*f*m*z^2 - 49152*a^8*c^8*h*k*z^2 - 147456*a^7*c^9*d*k*z^2 - 65536*a^7*c^9*e*j*z^2 - 16384*a^7*c^9*f*h*z^2 - 49152*a^6*c^10*d*f*z^2 + 716800*a^8*b^3*c^5*m^2*z^2 - 483840*a^7*b^5*c^4*m^2*z^2 + 170496*a^6*b^7*c^3*m^2*z^2 - 33232*a^5*b^9*c^2*m^2*z^2 + 516096*a^8*b^2*c^6*l^2*z^2 - 288768*a^7*b^4*c^5*l^2*z^2 + 88576*a^6*b^6*c^4*l^2*z^2 - 15744*a^5*b^8*c^3*l^2*z^2 + 1536*a^4*b^10*c^2*l^2*z^2 - 61440*a^7*b^3*c^6*k^2*z^2 + 24064*a^6*b^5*c^5*k^2*z^2 - 4608*a^5*b^7*c^4*k^2*z^2 + 432*a^4*b^9*c^3*k^2*z^2 - 16*a^3*b^11*c^2*k^2*z^2 + 24576*a^7*b^2*c^7*j^2*z^2 - 6144*a^6*b^4*c^6*j^2*z^2 + 512*a^5*b^6*c^5*j^2*z^2 - 8192*a^6*b^3*c^7*h^2*z^2 + 1536*a^5*b^5*c^6*h^2*z^2 - 16*a^3*b^9*c^4*h^2*z^2 - 8192*a^6*b^2*c^8*g^2*z^2 + 6144*a^5*b^4*c^7*g^2*z^2 - 1536*a^4*b^6*c^6*g^2*z^2 + 128*a^3*b^8*c^5*g^2*z^2 - 8192*a^5*b^3*c^8*f^2*z^2 + 1536*a^4*b^5*c^7*f^2*z^2 - 16*a^2*b^9*c^5*f^2*z^2 + 24576*a^5*b^2*c^9*e^2*z^2 - 6144*a^4*b^4*c^8*e^2*z^2 + 512*a^3*b^6*c^7*e^2*z^2 - 61440*a^4*b^3*c^9*d^2*z^2 + 24064*a^3*b^5*c^8*d^2*z^2 - 4608*a^2*b^7*c^7*d^2*z^2 - 393216*a^9*c^7*l^2*z^2 - 144*a^3*b^13*m^2*z^2 - 32768*a^8*c^8*j^2*z^2 - 32768*a^6*c^10*e^2*z^2 - 16*b^11*c^5*d^2*z^2 + 18432*a^8*b*c^5*h*l*m*z - 96*a^3*b^10*c*g*k*m*z + 90112*a^7*b*c^6*e*k*m*

z + 36864*a^7*b*c^6*f*j*m*z - 16384*a^7*b*c^6*g*j*l*z + 14336*a^7*b*c^6*d*l
*m*z - 10240*a^7*b*c^6*f*k*l*z + 4096*a^7*b*c^6*h*j*k*z + 10240*a^7*b*c^6*g
*h*m*z - 47104*a^6*b*c^7*d*h*l*z + 36864*a^6*b*c^7*e*f*m*z + 30720*a^6*b*c^
7*d*g*m*z - 16384*a^6*b*c^7*e*g*l*z + 6144*a^6*b*c^7*f*g*k*z + 4096*a^6*b*c
^7*e*h*k*z + 32*a*b^10*c^3*d*f*l*z - 4096*a^5*b*c^8*d*f*j*z - 6144*a^5*b*c^
8*d*g*h*z - 32*a*b^8*c^5*d*f*g*z - 4096*a^4*b*c^9*d*e*f*z + 64*a*b^7*c^6*d*
e*f*z + 110592*a^8*b^2*c^4*k*l*m*z - 36864*a^7*b^4*c^3*k*l*m*z + 5376*a^6*b
^6*c^2*k*l*m*z - 79872*a^7*b^3*c^4*j*k*m*z + 26112*a^6*b^5*c^3*j*k*m*z - 37
12*a^5*b^7*c^2*j*k*m*z - 13824*a^7*b^3*c^4*h*l*m*z + 3456*a^6*b^5*c^3*h*l*m
*z - 288*a^5*b^7*c^2*h*l*m*z - 45056*a^7*b^2*c^5*g*k*m*z + 39936*a^6*b^4*c^
4*g*k*m*z + 30720*a^7*b^2*c^5*f*l*m*z - 18432*a^7*b^2*c^5*h*k*l*z - 13056*a
^5*b^6*c^3*g*k*m*z - 7680*a^6*b^4*c^4*f*l*m*z + 5376*a^6*b^4*c^4*h*j*m*z +
4608*a^6*b^4*c^4*h*k*l*z + 3072*a^7*b^2*c^5*h*j*m*z - 1984*a^5*b^6*c^3*h*j*
m*z + 1856*a^4*b^8*c^2*g*k*m*z + 640*a^5*b^6*c^3*f*l*m*z - 384*a^5*b^6*c^3*
h*k*l*z + 192*a^4*b^8*c^2*h*j*m*z - 79872*a^6*b^3*c^5*e*k*m*z - 27648*a^6*b
^3*c^5*f*j*m*z + 26112*a^5*b^5*c^4*e*k*m*z + 12288*a^6*b^3*c^5*g*j*l*z - 10
752*a^6*b^3*c^5*d*l*m*z + 7680*a^6*b^3*c^5*f*k*l*z + 6912*a^5*b^5*c^4*f*j*m
*z - 3712*a^4*b^7*c^3*e*k*m*z - 3072*a^6*b^3*c^5*h*j*k*z - 3072*a^5*b^5*c^4
*g*j*l*z + 2688*a^5*b^5*c^4*d*l*m*z - 1920*a^5*b^5*c^4*f*k*l*z + 768*a^5*b^
5*c^4*h*j*k*z - 576*a^4*b^7*c^3*f*j*m*z + 256*a^4*b^7*c^3*g*j*l*z - 224*a^4
*b^7*c^3*d*l*m*z + 192*a^3*b^9*c^2*e*k*m*z + 160*a^4*b^7*c^3*f*k*l*z - 64*a
^4*b^7*c^3*h*j*k*z - 2688*a^5*b^5*c^4*g*h*m*z - 1536*a^6*b^3*c^5*g*h*m*z +
992*a^4*b^7*c^3*g*h*m*z - 96*a^3*b^9*c^2*g*h*m*z - 65536*a^6*b^2*c^6*d*k*l*
z + 46080*a^6*b^2*c^6*d*j*m*z - 24576*a^6*b^2*c^6*e*j*l*z + 21504*a^5*b^4*c
^5*d*k*l*z - 11520*a^5*b^4*c^5*d*j*m*z + 9216*a^6*b^2*c^6*f*j*k*z + 6144*a^
5*b^4*c^5*e*j*l*z - 3072*a^4*b^6*c^4*d*k*l*z - 2304*a^5*b^4*c^5*f*j*k*z + 9
60*a^4*b^6*c^4*d*j*m*z - 512*a^4*b^6*c^4*e*j*l*z + 192*a^4*b^6*c^4*f*j*k*z
+ 160*a^3*b^8*c^3*d*k*l*z - 18432*a^6*b^2*c^6*f*g*m*z + 13824*a^5*b^4*c^5*f
*g*m*z + 5376*a^5*b^4*c^5*e*h*m*z - 3456*a^4*b^6*c^4*f*g*m*z + 3072*a^6*b^2
*c^6*e*h*m*z - 3072*a^5*b^4*c^5*f*h*l*z - 2048*a^6*b^2*c^6*g*h*k*z - 1984*a
^4*b^6*c^4*e*h*m*z + 1536*a^5*b^4*c^5*g*h*k*z + 1024*a^4*b^6*c^4*f*h*l*z -
384*a^4*b^6*c^4*g*h*k*z + 288*a^3*b^8*c^3*f*g*m*z + 192*a^3*b^8*c^3*e*h*m*z
 - 96*a^3*b^8*c^3*f*h*l*z + 32*a^3*b^8*c^3*g*h*k*z + 41472*a^5*b^3*c^6*d*h*
l*z - 27648*a^5*b^3*c^6*e*f*m*z - 23040*a^5*b^3*c^6*d*g*m*z - 13440*a^4*b^5
*c^5*d*h*l*z + 12288*a^5*b^3*c^6*e*g*l*z + 6912*a^4*b^5*c^5*e*f*m*z + 5760*
a^4*b^5*c^5*d*g*m*z - 4608*a^5*b^3*c^6*f*g*k*z - 3072*a^5*b^3*c^6*e*h*k*z -
 3072*a^4*b^5*c^5*e*g*l*z + 1888*a^3*b^7*c^4*d*h*l*z + 1152*a^4*b^5*c^5*f*g
*k*z + 768*a^4*b^5*c^5*e*h*k*z - 576*a^3*b^7*c^4*e*f*m*z - 480*a^3*b^7*c^4*
d*g*m*z + 256*a^3*b^7*c^4*e*g*l*z - 96*a^3*b^7*c^4*f*g*k*z - 96*a^2*b^9*c^3
*d*h*l*z - 64*a^3*b^7*c^4*e*h*k*z + 46080*a^5*b^2*c^7*d*e*m*z - 11520*a^4*b
^4*c^6*d*e*m*z + 9216*a^5*b^2*c^7*e*f*k*z - 9216*a^5*b^2*c^7*d*h*j*z - 6656
*a^4*b^4*c^6*d*f*l*z - 6144*a^5*b^2*c^7*d*f*l*z + 3456*a^3*b^6*c^5*d*f*l*z
- 2304*a^4*b^4*c^6*e*f*k*z + 2304*a^4*b^4*c^6*d*h*j*z + 960*a^3*b^6*c^5*d*e
*m*z - 576*a^2*b^8*c^4*d*f*l*z + 192*a^3*b^6*c^5*e*f*k*z - 192*a^3*b^6*c^5*
d*h*j*z + 3072*a^4*b^3*c^7*d*f*j*z - 768*a^3*b^5*c^6*d*f*j*z + 64*a^2*b^7*c

^5*d*f*j*z + 4608*a^4*b^3*c^7*d*g*h*z - 1152*a^3*b^5*c^6*d*g*h*z + 96*a^2*b
^7*c^5*d*g*h*z - 9216*a^4*b^2*c^8*d*e*h*z + 2304*a^3*b^4*c^7*d*e*h*z + 2048
*a^4*b^2*c^8*d*f*g*z - 1536*a^3*b^4*c^7*d*f*g*z + 384*a^2*b^6*c^6*d*f*g*z -
 192*a^2*b^6*c^6*d*e*h*z + 3072*a^3*b^3*c^8*d*e*f*z - 768*a^2*b^5*c^7*d*e*f
*z - 288*a^5*b^8*c*k*l*m*z + 90112*a^8*b*c^5*j*k*m*z + 192*a^4*b^9*c*j*k*m*
z + 138240*a^9*b*c^4*l*m^2*z - 7344*a^6*b^7*c*l*m^2*z + 5088*a^5*b^8*c*j*m^
2*z - 3072*a^8*b*c^5*k^2*l*z - 49152*a^8*b*c^5*j*l^2*z - 128*a^4*b^9*c*j*l^
2*z - 25600*a^8*b*c^5*g*m^2*z - 9216*a^7*b*c^6*h^2*l*z - 2544*a^4*b^9*c*g*m
^2*z + 64*a^3*b^10*c*g*l^2*z + 9216*a^7*b*c^6*g*k^2*z - 3072*a^6*b*c^7*f^2*
l*z - 288*a^3*b^10*c*e*m^2*z - 49152*a^7*b*c^6*e*l^2*z - 58368*a^5*b*c^8*d^
2*l*z - 432*a*b^9*c^4*d^2*l*z - 1024*a^6*b*c^7*g*h^2*z + 32*a*b^8*c^5*d^2*j
*z + 1024*a^5*b*c^8*f^2*g*z - 9216*a^4*b*c^9*d^2*g*z + 336*a*b^7*c^6*d^2*g*
z - 672*a*b^6*c^7*d^2*e*z - 122880*a^9*c^5*k*l*m*z - 40960*a^8*c^6*f*l*m*z
+ 24576*a^8*c^6*h*k*l*z - 20480*a^8*c^6*h*j*m*z + 73728*a^7*c^7*d*k*l*z - 6
1440*a^7*c^7*d*j*m*z + 32768*a^7*c^7*e*j*l*z - 12288*a^7*c^7*f*j*k*z - 2048
0*a^7*c^7*e*h*m*z + 8192*a^7*c^7*f*h*l*z - 61440*a^6*c^8*d*e*m*z + 24576*a^
6*c^8*d*f*l*z - 12288*a^6*c^8*e*f*k*z + 12288*a^6*c^8*d*h*j*z + 12288*a^5*c
^9*d*e*h*z - 131328*a^8*b^3*c^3*l*m^2*z + 46656*a^7*b^5*c^2*l*m^2*z - 14284
8*a^8*b^2*c^4*j*m^2*z + 106368*a^7*b^4*c^3*j*m^2*z - 34208*a^6*b^6*c^2*j*m^
2*z + 2304*a^7*b^3*c^4*k^2*l*z - 576*a^6*b^5*c^3*k^2*l*z + 48*a^5*b^7*c^2*k
^2*l*z + 45056*a^7*b^3*c^4*j*l^2*z - 15360*a^6*b^5*c^3*j*l^2*z - 12288*a^7*
b^2*c^5*j^2*l*z + 3072*a^6*b^4*c^4*j^2*l*z + 2304*a^5*b^7*c^2*j*l^2*z - 256
*a^5*b^6*c^3*j^2*l*z + 15872*a^7*b^2*c^5*j*k^2*z - 4992*a^6*b^4*c^4*j*k^2*z
 + 672*a^5*b^6*c^3*j*k^2*z - 32*a^4*b^8*c^2*j*k^2*z + 71424*a^7*b^3*c^4*g*m
^2*z - 53184*a^6*b^5*c^3*g*m^2*z + 17104*a^5*b^7*c^2*g*m^2*z + 6912*a^6*b^3
*c^5*h^2*l*z - 1728*a^5*b^5*c^4*h^2*l*z + 144*a^4*b^7*c^3*h^2*l*z + 24576*a
^7*b^2*c^5*g*l^2*z - 22528*a^6*b^4*c^4*g*l^2*z + 7680*a^5*b^6*c^3*g*l^2*z +
 4096*a^6*b^2*c^6*g^2*l*z - 3072*a^5*b^4*c^5*g^2*l*z - 1152*a^4*b^8*c^2*g*l
^2*z + 768*a^4*b^6*c^4*g^2*l*z - 64*a^3*b^8*c^3*g^2*l*z - 142848*a^7*b^2*c^
5*e*m^2*z + 106368*a^6*b^4*c^4*e*m^2*z - 34208*a^5*b^6*c^3*e*m^2*z - 7936*a
^6*b^3*c^5*g*k^2*z + 5088*a^4*b^8*c^2*e*m^2*z + 2496*a^5*b^5*c^4*g*k^2*z -
1536*a^6*b^2*c^6*h^2*j*z + 1280*a^5*b^3*c^6*f^2*l*z + 384*a^5*b^4*c^5*h^2*j
*z - 336*a^4*b^7*c^3*g*k^2*z + 192*a^4*b^5*c^5*f^2*l*z - 144*a^3*b^7*c^4*f^
2*l*z - 32*a^4*b^6*c^4*h^2*j*z + 16*a^3*b^9*c^2*g*k^2*z + 16*a^2*b^9*c^3*f^
2*l*z + 45056*a^6*b^3*c^5*e*l^2*z - 15360*a^5*b^5*c^4*e*l^2*z - 12288*a^5*b
^2*c^7*e^2*l*z + 3072*a^4*b^4*c^6*e^2*l*z + 2304*a^4*b^7*c^3*e*l^2*z - 256*
a^3*b^6*c^5*e^2*l*z - 128*a^3*b^9*c^2*e*l^2*z + 59136*a^4*b^3*c^7*d^2*l*z -
 23488*a^3*b^5*c^6*d^2*l*z + 15872*a^6*b^2*c^6*e*k^2*z - 4992*a^5*b^4*c^5*e
*k^2*z + 4560*a^2*b^7*c^5*d^2*l*z + 1536*a^5*b^2*c^7*f^2*j*z + 672*a^4*b^6*
c^4*e*k^2*z - 384*a^4*b^4*c^6*f^2*j*z - 32*a^3*b^8*c^3*e*k^2*z + 32*a^3*b^6
*c^5*f^2*j*z + 768*a^5*b^3*c^6*g*h^2*z - 192*a^4*b^5*c^5*g*h^2*z + 16*a^3*b
^7*c^4*g*h^2*z - 15872*a^4*b^2*c^8*d^2*j*z + 4992*a^3*b^4*c^7*d^2*j*z - 672
*a^2*b^6*c^6*d^2*j*z - 1536*a^5*b^2*c^7*e*h^2*z - 768*a^4*b^3*c^7*f^2*g*z +
 384*a^4*b^4*c^6*e*h^2*z + 192*a^3*b^5*c^6*f^2*g*z - 32*a^3*b^6*c^5*e*h^2*z
 - 16*a^2*b^7*c^5*f^2*g*z + 7936*a^3*b^3*c^8*d^2*g*z - 2496*a^2*b^5*c^7*d^2

*g*z + 1536*a^4*b^2*c^8*e*f^2*z - 384*a^3*b^4*c^7*e*f^2*z + 32*a^2*b^6*c^6*
e*f^2*z - 15872*a^3*b^2*c^9*d^2*e*z + 4992*a^2*b^4*c^8*d^2*e*z - 61440*a^8*
b^2*c^4*l^3*z + 21504*a^7*b^4*c^3*l^3*z - 3328*a^6*b^6*c^2*l^3*z + 432*a^5*
b^9*l*m^2*z + 51200*a^9*c^5*j*m^2*z + 16384*a^8*c^6*j^2*l*z - 288*a^4*b^10*
j*m^2*z - 18432*a^8*c^6*j*k^2*z + 144*a^3*b^11*g*m^2*z + 51200*a^8*c^6*e*m^
2*z + 2048*a^7*c^7*h^2*j*z + 16384*a^6*c^8*e^2*l*z + 16*b^11*c^3*d^2*l*z -
18432*a^7*c^7*e*k^2*z - 2048*a^6*c^8*f^2*j*z + 18432*a^5*c^9*d^2*j*z + 192*
a^5*b^8*c*l^3*z + 2048*a^6*c^8*e*h^2*z - 16*b^9*c^5*d^2*g*z - 2048*a^5*c^9*
e*f^2*z + 32*b^8*c^6*d^2*e*z + 18432*a^4*c^10*d^2*e*z + 65536*a^9*c^5*l^3*z
 - 11008*a^8*b*c^3*j*k*l*m - 288*a^6*b^5*c*j*k*l*m + 144*a^5*b^6*c*g*k*l*m
- 11008*a^7*b*c^4*e*k*l*m - 5376*a^7*b*c^4*f*j*l*m + 3840*a^7*b*c^4*g*j*k*m
 - 3328*a^7*b*c^4*h*j*k*l - 96*a^4*b^7*c*g*j*k*m - 2560*a^7*b*c^4*g*h*l*m -
 36*a^3*b^8*c*f*h*k*m - 6912*a^6*b*c^5*d*j*k*l - 7872*a^6*b*c^5*d*h*k*m - 7
680*a^6*b*c^5*d*g*l*m - 5376*a^6*b*c^5*e*f*l*m + 3840*a^6*b*c^5*e*g*k*m - 3
328*a^6*b*c^5*e*h*k*l - 1536*a^6*b*c^5*f*g*k*l + 1280*a^6*b*c^5*f*g*j*m - 7
68*a^6*b*c^5*g*h*j*k - 768*a^6*b*c^5*f*h*j*l - 768*a^6*b*c^5*e*h*j*m - 36*a
^2*b^9*c*d*h*k*m - 6912*a^5*b*c^6*d*e*k*l - 4864*a^5*b*c^6*d*e*j*m - 2304*a
^5*b*c^6*d*g*j*k - 1792*a^5*b*c^6*e*f*j*k - 1280*a^5*b*c^6*d*f*j*l - 4544*a
^5*b*c^6*d*f*h*m + 1536*a^5*b*c^6*d*g*h*l + 1280*a^5*b*c^6*e*f*g*m - 768*a^
5*b*c^6*e*g*h*k - 768*a^5*b*c^6*e*f*h*l - 256*a^5*b*c^6*f*g*h*j + 12*a*b^9*
c^2*d*f*h*m + 16*a*b^8*c^3*d*f*g*l - 4*a*b^8*c^3*d*f*h*k - 2304*a^4*b*c^7*d
*e*g*k - 1792*a^4*b*c^7*d*e*h*j - 1280*a^4*b*c^7*d*e*f*l - 768*a^4*b*c^7*d*
f*g*j - 32*a*b^7*c^4*d*e*f*l - 256*a^4*b*c^7*e*f*g*h - 768*a^3*b*c^8*d*e*f*
g + 32*a*b^5*c^6*d*e*f*g + 12*a*b^10*c*d*f*k*m + 3648*a^7*b^3*c^2*j*k*l*m +
 5504*a^7*b^2*c^3*g*k*l*m - 1824*a^6*b^4*c^2*g*k*l*m + 384*a^7*b^2*c^3*h*j*
l*m - 288*a^6*b^4*c^2*h*j*l*m - 4800*a^6*b^3*c^3*g*j*k*m + 3648*a^6*b^3*c^3
*e*k*l*m + 1280*a^5*b^5*c^2*g*j*k*m + 1088*a^6*b^3*c^3*f*j*l*m + 576*a^6*b^
3*c^3*h*j*k*l - 288*a^5*b^5*c^2*e*k*l*m - 192*a^6*b^3*c^3*g*h*l*m + 144*a^5
*b^5*c^2*g*h*l*m + 9600*a^6*b^2*c^4*e*j*k*m - 4224*a^6*b^2*c^4*d*j*l*m - 25
60*a^5*b^4*c^3*e*j*k*m + 384*a^6*b^2*c^4*f*j*k*l + 224*a^5*b^4*c^3*d*j*l*m
+ 192*a^4*b^6*c^2*e*j*k*m - 160*a^5*b^4*c^3*f*j*k*l - 4608*a^6*b^2*c^4*f*h*
k*m + 2688*a^6*b^2*c^4*f*g*l*m + 1664*a^6*b^2*c^4*g*h*k*l - 744*a^5*b^4*c^3
*f*h*k*m - 544*a^5*b^4*c^3*f*g*l*m + 492*a^4*b^6*c^2*f*h*k*m + 416*a^5*b^4*
c^3*g*h*j*m + 384*a^6*b^2*c^4*g*h*j*m + 384*a^6*b^2*c^4*e*h*l*m - 288*a^5*b
^4*c^3*g*h*k*l - 288*a^5*b^4*c^3*e*h*l*m - 96*a^4*b^6*c^2*g*h*j*m + 2112*a^
5*b^3*c^4*d*j*k*l - 160*a^4*b^5*c^3*d*j*k*l + 16992*a^5*b^3*c^4*d*h*k*m - 6
252*a^4*b^5*c^3*d*h*k*m - 4800*a^5*b^3*c^4*e*g*k*m + 2112*a^5*b^3*c^4*d*g*l
*m - 1728*a^5*b^3*c^4*f*g*j*m + 1280*a^4*b^5*c^3*e*g*k*m + 1088*a^5*b^3*c^4
*e*f*l*m - 832*a^5*b^3*c^4*e*h*j*m + 816*a^3*b^7*c^2*d*h*k*m + 576*a^5*b^3*
c^4*e*h*k*l - 448*a^5*b^3*c^4*f*h*j*l + 288*a^4*b^5*c^3*f*g*j*m - 192*a^5*b
^3*c^4*g*h*j*k - 192*a^5*b^3*c^4*f*g*k*l + 192*a^4*b^5*c^3*e*h*j*m - 112*a^
4*b^5*c^3*d*g*l*m + 96*a^4*b^5*c^3*f*h*j*l - 96*a^3*b^7*c^2*e*g*k*m + 80*a^
4*b^5*c^3*f*g*k*l + 32*a^4*b^5*c^3*g*h*j*k - 11456*a^5*b^2*c^5*d*f*k*m + 49
92*a^5*b^2*c^5*d*h*j*l - 4608*a^5*b^2*c^5*e*g*j*l - 4224*a^5*b^2*c^5*d*e*l*
m + 3456*a^5*b^2*c^5*e*f*j*m + 3456*a^5*b^2*c^5*d*g*k*l + 2432*a^5*b^2*c^5*

d*g*j*m - 1312*a^4*b^4*c^4*d*h*j*l + 1272*a^3*b^6*c^3*d*f*k*m - 1056*a^4*b^4*c^4*d*g*k*l + 896*a^5*b^2*c^5*f*g*j*k + 768*a^4*b^4*c^4*e*g*j*l - 576*a^4*b^4*c^4*e*f*j*m - 480*a^4*b^4*c^4*d*g*j*m + 384*a^5*b^2*c^5*e*h*j*k + 384*a^5*b^2*c^5*e*f*k*l - 232*a^2*b^8*c^2*d*f*k*m + 224*a^4*b^4*c^4*d*e*l*m - 160*a^4*b^4*c^4*e*f*k*l - 96*a^4*b^4*c^4*f*g*j*k + 96*a^3*b^6*c^3*d*h*j*l + 80*a^3*b^6*c^3*d*g*k*l - 64*a^4*b^4*c^4*e*h*j*k - 24*a^4*b^4*c^4*d*f*k*m + 416*a^4*b^4*c^4*e*g*h*m + 384*a^5*b^2*c^5*f*g*h*l + 384*a^5*b^2*c^5*e*g*h*m + 224*a^4*b^4*c^4*f*g*h*l - 96*a^3*b^6*c^3*e*g*h*m - 48*a^3*b^6*c^3*f*g*h*l + 2112*a^4*b^3*c^5*d*e*k*l - 960*a^4*b^3*c^5*d*f*j*l + 960*a^4*b^3*c^5*d*e*j*m + 384*a^3*b^5*c^4*d*f*j*l + 320*a^4*b^3*c^5*d*g*j*k + 192*a^4*b^3*c^5*e*f*j*k - 160*a^3*b^5*c^4*d*e*k*l - 32*a^2*b^7*c^3*d*f*j*l + 7392*a^4*b^3*c^5*d*f*h*m - 2496*a^4*b^3*c^5*d*g*h*l - 1728*a^4*b^3*c^5*e*f*g*m - 1500*a^3*b^5*c^4*d*f*h*m + 656*a^3*b^5*c^4*d*g*h*l - 448*a^4*b^3*c^5*e*f*h*l + 288*a^3*b^5*c^4*e*f*g*m - 192*a^4*b^3*c^5*f*g*h*j - 192*a^4*b^3*c^5*e*g*h*k + 96*a^3*b^5*c^4*e*f*h*l - 48*a^2*b^7*c^3*d*g*h*l + 32*a^3*b^5*c^4*e*g*h*k - 16*a^2*b^7*c^3*d*f*h*m - 640*a^4*b^2*c^6*d*e*j*k + 4992*a^4*b^2*c^6*d*e*h*l - 3584*a^4*b^2*c^6*d*f*h*k + 2432*a^4*b^2*c^6*d*e*g*m - 1312*a^3*b^4*c^5*d*e*h*l + 896*a^4*b^2*c^6*e*f*g*k + 896*a^4*b^2*c^6*d*g*h*j + 640*a^4*b^2*c^6*d*f*g*l + 600*a^3*b^4*c^5*d*f*h*k + 480*a^3*b^4*c^5*d*f*g*l - 480*a^3*b^4*c^5*d*e*g*m + 384*a^4*b^2*c^6*e*f*h*j - 192*a^2*b^6*c^4*d*f*g*l - 96*a^3*b^4*c^5*e*f*g*k - 96*a^3*b^4*c^5*d*g*h*j + 96*a^2*b^6*c^4*d*e*h*l + 12*a^2*b^6*c^4*d*f*h*k - 960*a^3*b^3*c^6*d*e*f*l + 384*a^2*b^5*c^5*d*e*f*l + 320*a^3*b^3*c^6*d*e*g*k - 192*a^3*b^3*c^6*d*f*g*j + 192*a^3*b^3*c^6*d*e*h*j + 32*a^2*b^5*c^5*d*f*g*j - 192*a^3*b^3*c^6*e*f*g*h + 384*a^3*b^2*c^7*d*e*f*j - 64*a^2*b^4*c^6*d*e*f*j + 896*a^3*b^2*c^7*d*e*g*h - 96*a^2*b^4*c^6*d*e*g*h - 192*a^2*b^3*c^7*d*e*f*g + 496*a^7*b^4*c*k*l^2*m - 4752*a^7*b^4*c*j*l*m^2 + 96*a^5*b^6*c*j^2*k*m - 6144*a^8*b*c^3*h*l^2*m - 168*a^6*b^5*c*h*l^2*m + 6400*a^8*b*c^3*g*l*m^2 - 2862*a^6*b^5*c*h*k*m^2 + 2376*a^6*b^5*c*g*l*m^2 - 1632*a^7*b*c^4*h^2*k*m - 480*a^8*b*c^3*h*k*m^2 - 180*a^5*b^6*c*h*k^2*m + 54*a^4*b^7*c*h^2*k*m - 384*a^7*b*c^4*h*j^2*m + 120*a^5*b^6*c*h*k*l^2 + 56*a^5*b^6*c*f*l^2*m + 24*a^3*b^8*c*g^2*k*m + 4512*a^7*b*c^4*f*k^2*m - 2304*a^7*b*c^4*g*k^2*l - 1680*a^5*b^6*c*g*j*m^2 + 1184*a^6*b*c^5*f^2*k*m + 804*a^5*b^6*c*f*k*m^2 + 432*a^5*b^6*c*e*l*m^2 + 60*a^4*b^7*c*f*k^2*m + 6*a^2*b^9*c*f^2*k*m - 13312*a^7*b*c^4*d*l^2*m + 2048*a^7*b*c^4*g*j*l^2 - 1024*a^7*b*c^4*f*k*l^2 + 64*a^4*b^7*c*g*j*l^2 + 56*a^4*b^7*c*d*l^2*m - 40*a^4*b^7*c*f*k*l^2 + 13440*a^7*b*c^4*e*j*m^2 - 8928*a^5*b*c^6*d^2*k*m - 6240*a^7*b*c^4*d*k*m^2 + 1614*a^4*b^7*c*d*k*m^2 - 288*a^4*b^7*c*e*j*m^2 - 170*a*b^9*c^2*d^2*k*m + 60*a^3*b^8*c*d*k^2*m + 4608*a^6*b*c^5*e*j^2*l + 4608*a^5*b*c^6*e^2*j*l - 2432*a^6*b*c^5*d*j^2*m + 1440*a^7*b*c^4*f*h*m^2 - 896*a^6*b*c^5*f*j^2*k - 864*a^6*b*c^5*f*h^2*m - 558*a^4*b^7*c*f*h*m^2 + 256*a^6*b*c^5*g*h^2*l - 40*a^3*b^8*c*d*k*l^2 - 1920*a^6*b*c^5*e*j*k^2 - 384*a^5*b*c^6*e^2*h*m + 24*a^3*b^8*c*f*h*l^2 - 16*a*b^8*c^3*d^2*j*l + 2208*a^6*b*c^5*f*h*k^2 - 1044*a^3*b^8*c*d*h*m^2 + 800*a^5*b*c^6*f^2*h*k - 256*a^5*b*c^6*f^2*g*l + 144*a^3*b^8*c*e*g*m^2 - 116*a*b^8*c^3*d^2*h*m + 8192*a^6*b*c^5*d*h*l^2 + 2048*a^6*b*c^5*e*g*l^2 + 24*a^2*b^9*c*d*h*l^2 - 5856*a^4*b*c^7*d^2*f*m + 4896*a^4*b*c^7*d^2*h

*k + 2720*a^6*b*c^5*d*f*m^2 + 2304*a^4*b*c^7*d^2*g*l + 1824*a^5*b*c^6*d*h^2
*k + 438*a*b^7*c^4*d^2*f*m - 384*a^5*b*c^6*e*h^2*j + 318*a^2*b^9*c*d*f*m^2
- 168*a*b^7*c^4*d^2*g*l + 42*a*b^7*c^4*d^2*h*k - 36*a*b^8*c^3*d*f^2*m - 243
2*a^4*b*c^7*d*e^2*m + 1536*a^5*b*c^6*e*g*j^2 + 1536*a^4*b*c^7*e^2*g*j - 896
*a^5*b*c^6*d*h*j^2 - 896*a^4*b*c^7*e^2*f*k + 4896*a^5*b*c^6*d*f*k^2 + 1824*
a^4*b*c^7*d*f^2*k - 384*a^4*b*c^7*e*f^2*j + 336*a*b^6*c^5*d^2*e*l - 156*a*b
^6*c^5*d^2*f*k + 16*a*b^6*c^5*d^2*g*j + 12*a*b^7*c^4*d*f^2*k - 2*a*b^9*c^2*
d*f*k^2 - 1920*a^3*b*c^8*d^2*e*j - 32*a*b^5*c^6*d^2*e*j + 2208*a^3*b*c^8*d^
2*f*h + 800*a^4*b*c^7*d*f*h^2 - 102*a*b^5*c^6*d^2*f*h + 12*a*b^6*c^5*d*f^2*
h - 2*a*b^7*c^4*d*f*h^2 - 896*a^3*b*c^8*d*e^2*h - 8*a*b^6*c^5*d*f*g^2 - 240
*a*b^4*c^7*d^2*e*g - 32*a*b^4*c^7*d*e^2*f + 5120*a^8*c^4*h*j*l*m + 15360*a^
7*c^5*d*j*l*m - 7680*a^7*c^5*e*j*k*m + 3072*a^7*c^5*f*j*k*l + 5120*a^7*c^5*
e*h*l*m + 1920*a^7*c^5*f*h*k*m + 15360*a^6*c^6*d*e*l*m + 5760*a^6*c^6*d*f*k
*m + 3072*a^6*c^6*e*f*k*l - 3072*a^6*c^6*d*h*j*l - 2560*a^6*c^6*e*f*j*m + 1
536*a^6*c^6*e*h*j*k + 4608*a^5*c^7*d*e*j*k - 3072*a^5*c^7*d*e*h*l - 1152*a^
5*c^7*d*f*h*k + 512*a^5*c^7*e*f*h*j + 1536*a^4*c^8*d*e*f*j - 8*a*b^10*c*d*f
*l^2 - 5568*a^8*b^2*c^2*k*l^2*m + 15552*a^8*b^2*c^2*j*l*m^2 + 4800*a^7*b^2*
c^3*j^2*k*m - 1280*a^6*b^4*c^2*j^2*k*m + 2080*a^7*b^3*c^2*h*l^2*m - 1088*a^
7*b^2*c^3*j*k^2*l + 48*a^6*b^4*c^2*j*k^2*l - 8544*a^7*b^2*c^3*h*k^2*m - 777
6*a^7*b^3*c^2*g*l*m^2 + 7632*a^7*b^3*c^2*h*k*m^2 + 3600*a^6*b^3*c^3*h^2*k*m
 + 2484*a^6*b^4*c^2*h*k^2*m - 918*a^5*b^5*c^2*h^2*k*m + 4800*a^7*b^2*c^3*h*
k*l^2 - 1424*a^6*b^4*c^2*h*k*l^2 + 1200*a^5*b^4*c^3*g^2*k*m - 960*a^6*b^2*c
^4*g^2*k*m - 528*a^6*b^4*c^2*f*l^2*m - 416*a^6*b^3*c^3*h*j^2*m - 320*a^4*b^
6*c^2*g^2*k*m + 192*a^7*b^2*c^3*f*l^2*m + 96*a^5*b^5*c^2*h*j^2*m + 15552*a^
7*b^2*c^3*e*l*m^2 - 6720*a^7*b^2*c^3*g*j*m^2 + 6160*a^6*b^4*c^2*g*j*m^2 - 4
752*a^6*b^4*c^2*e*l*m^2 - 2016*a^7*b^2*c^3*f*k*m^2 - 1164*a^6*b^4*c^2*f*k*m
^2 + 1104*a^5*b^3*c^4*f^2*k*m + 1008*a^6*b^3*c^3*f*k^2*m + 960*a^6*b^2*c^4*
h^2*j*l - 678*a^5*b^5*c^2*f*k^2*m + 544*a^6*b^3*c^3*g*k^2*l - 144*a^5*b^4*c
^3*h^2*j*l - 102*a^4*b^5*c^3*f^2*k*m - 62*a^3*b^7*c^2*f^2*k*m - 24*a^5*b^5*
c^2*g*k^2*l + 6432*a^6*b^3*c^3*d*l^2*m + 4800*a^5*b^2*c^5*e^2*k*m - 2304*a^
6*b^2*c^4*g*j^2*l + 1920*a^6*b^3*c^3*g*j*l^2 + 1728*a^6*b^2*c^4*f*j^2*m - 1
280*a^4*b^4*c^4*e^2*k*m + 1152*a^5*b^3*c^4*g^2*j*l - 1032*a^5*b^5*c^2*d*l^2
*m - 864*a^6*b^3*c^3*f*k*l^2 - 768*a^5*b^5*c^2*g*j*l^2 + 408*a^5*b^5*c^2*f*
k*l^2 + 384*a^5*b^4*c^3*g*j^2*l - 288*a^5*b^4*c^3*f*j^2*m + 192*a^6*b^2*c^4
*h*j^2*k - 192*a^4*b^5*c^3*g^2*j*l + 96*a^3*b^6*c^3*e^2*k*m - 32*a^5*b^4*c^
3*h*j^2*k - 21120*a^6*b^2*c^4*d*k^2*m + 20880*a^6*b^3*c^3*d*k*m^2 + 19760*a
^4*b^3*c^5*d^2*k*m - 12320*a^6*b^3*c^3*e*j*m^2 - 9750*a^5*b^5*c^2*d*k*m^2 -
 9390*a^3*b^5*c^4*d^2*k*m + 8460*a^5*b^4*c^3*d*k^2*m + 3360*a^5*b^5*c^2*e*j
*m^2 + 1860*a^2*b^7*c^3*d^2*k*m - 1218*a^4*b^6*c^2*d*k^2*m - 1088*a^6*b^2*c
^4*e*k^2*l + 960*a^6*b^2*c^4*g*j*k^2 - 240*a^5*b^4*c^3*g*j*k^2 + 192*a^5*b^
2*c^5*f^2*j*l - 104*a^4*b^5*c^3*g^2*h*m - 96*a^5*b^3*c^4*g^2*h*m + 48*a^5*b
^4*c^3*e*k^2*l + 48*a^4*b^4*c^4*f^2*j*l + 24*a^3*b^7*c^2*g^2*h*m + 16*a^4*b
^6*c^2*g*j*k^2 - 16*a^3*b^6*c^3*f^2*j*l + 13376*a^6*b^2*c^4*d*k*l^2 - 5136*
a^5*b^4*c^3*d*k*l^2 - 3840*a^6*b^2*c^4*e*j*l^2 + 1536*a^5*b^4*c^3*e*j*l^2 +
 1392*a^5*b^3*c^4*f*h^2*m + 1386*a^5*b^5*c^2*f*h*m^2 - 768*a^5*b^3*c^4*e*j^

2*l + 768*a^4*b^6*c^2*d*k*l^2 - 768*a^4*b^3*c^5*e^2*j*l - 588*a^4*b^4*c^4*f
^2*h*m - 480*a^5*b^3*c^4*g*h^2*l + 480*a^5*b^3*c^4*d*j^2*m - 480*a^5*b^2*c^
5*f^2*h*m - 128*a^4*b^6*c^2*e*j*l^2 + 100*a^3*b^6*c^3*f^2*h*m + 96*a^5*b^3*
c^4*f*j^2*k + 72*a^4*b^5*c^3*g*h^2*l - 54*a^4*b^5*c^3*f*h^2*m - 48*a^6*b^3*
c^3*f*h*m^2 - 36*a^3*b^7*c^2*f*h^2*m + 6*a^2*b^8*c^2*f^2*h*m + 6848*a^4*b^2
*c^6*d^2*j*l - 2448*a^3*b^4*c^5*d^2*j*l + 624*a^5*b^4*c^3*f*h*l^2 + 576*a^6
*b^2*c^4*f*h*l^2 + 480*a^5*b^3*c^4*e*j*k^2 + 432*a^4*b^4*c^4*f*g^2*m - 416*
a^4*b^3*c^5*e^2*h*m + 336*a^2*b^6*c^4*d^2*j*l - 320*a^5*b^2*c^5*f*g^2*m - 2
56*a^4*b^6*c^2*f*h*l^2 + 192*a^5*b^2*c^5*g^2*h*k + 96*a^3*b^5*c^4*e^2*h*m -
 72*a^3*b^6*c^3*f*g^2*m + 48*a^4*b^4*c^4*g^2*h*k - 32*a^4*b^5*c^3*e*j*k^2 -
 8*a^3*b^6*c^3*g^2*h*k + 24768*a^6*b^2*c^4*d*h*m^2 - 21108*a^5*b^4*c^3*d*h*
m^2 - 10048*a^4*b^2*c^6*d^2*h*m + 7218*a^4*b^6*c^2*d*h*m^2 - 6720*a^6*b^2*c
^4*e*g*m^2 + 6160*a^5*b^4*c^3*e*g*m^2 - 2592*a^5*b^2*c^5*d*h^2*m - 1680*a^4
*b^6*c^2*e*g*m^2 + 1068*a^3*b^4*c^5*d^2*h*m + 960*a^5*b^2*c^5*e*h^2*l - 876
*a^4*b^4*c^4*d*h^2*m - 864*a^5*b^2*c^5*f*h^2*k + 546*a^2*b^6*c^4*d^2*h*m +
432*a^3*b^6*c^3*d*h^2*m + 336*a^4*b^3*c^5*f^2*h*k - 320*a^5*b^2*c^5*d*j^2*k
 + 192*a^5*b^2*c^5*g*h^2*j + 144*a^5*b^3*c^4*f*h*k^2 - 144*a^4*b^4*c^4*e*h^
2*l - 102*a^4*b^5*c^3*f*h*k^2 - 96*a^4*b^3*c^5*f^2*g*l - 36*a^2*b^8*c^2*d*h
^2*m - 30*a^3*b^5*c^4*f^2*h*k - 24*a^3*b^5*c^4*f^2*g*l + 16*a^4*b^4*c^4*g*h
^2*j - 12*a^4*b^4*c^4*f*h^2*k + 12*a^3*b^6*c^3*f*h^2*k + 8*a^2*b^7*c^3*f^2*
g*l + 6*a^3*b^7*c^2*f*h*k^2 - 2*a^2*b^7*c^3*f^2*h*k - 9312*a^5*b^3*c^4*d*h*
l^2 + 3288*a^4*b^5*c^3*d*h*l^2 - 2304*a^4*b^2*c^6*e^2*g*l + 1920*a^5*b^3*c^
4*e*g*l^2 + 1728*a^4*b^2*c^6*e^2*f*m + 1152*a^4*b^3*c^5*e*g^2*l - 768*a^4*b
^5*c^3*e*g*l^2 - 608*a^4*b^3*c^5*d*g^2*m - 472*a^3*b^7*c^2*d*h*l^2 + 384*a^
3*b^4*c^5*e^2*g*l - 288*a^3*b^4*c^5*e^2*f*m - 224*a^4*b^3*c^5*f*g^2*k + 192
*a^5*b^2*c^5*f*h*j^2 + 192*a^4*b^2*c^6*e^2*h*k - 192*a^3*b^5*c^4*e*g^2*l +
120*a^3*b^5*c^4*d*g^2*m + 64*a^3*b^7*c^2*e*g*l^2 - 32*a^3*b^4*c^5*e^2*h*k +
 24*a^3*b^5*c^4*f*g^2*k + 9936*a^3*b^3*c^6*d^2*f*m + 3786*a^4*b^5*c^3*d*f*m
^2 - 3552*a^5*b^2*c^5*d*h*k^2 - 3486*a^2*b^5*c^5*d^2*f*m - 3424*a^3*b^3*c^6
*d^2*g*l - 1868*a^3*b^7*c^2*d*f*m^2 + 1332*a^4*b^4*c^4*d*h*k^2 - 1296*a^5*b
^3*c^4*d*f*m^2 - 1236*a^3*b^4*c^5*d*f^2*m + 1224*a^2*b^5*c^5*d^2*g*l - 1152
*a^4*b^2*c^6*d*f^2*m + 960*a^5*b^2*c^5*e*g*k^2 - 496*a^3*b^3*c^6*d^2*h*k +
462*a^2*b^6*c^4*d*f^2*m + 432*a^4*b^3*c^5*d*h^2*k - 240*a^4*b^4*c^4*e*g*k^2
 - 222*a^2*b^5*c^5*d^2*h*k + 192*a^4*b^2*c^6*f^2*g*j + 192*a^4*b^2*c^6*e*f^
2*l - 174*a^3*b^5*c^4*d*h^2*k - 156*a^3*b^6*c^3*d*h*k^2 + 48*a^3*b^4*c^5*e*
f^2*l - 32*a^4*b^3*c^5*e*h^2*j + 16*a^3*b^6*c^3*e*g*k^2 + 16*a^3*b^4*c^5*f^
2*g*j - 16*a^2*b^6*c^4*e*f^2*l + 12*a^2*b^7*c^3*d*h^2*k + 6*a^2*b^8*c^2*d*h
*k^2 + 1728*a^5*b^2*c^5*d*f*l^2 + 1392*a^4*b^4*c^4*d*f*l^2 - 840*a^3*b^6*c^
3*d*f*l^2 - 768*a^4*b^2*c^6*e*g^2*j + 576*a^4*b^2*c^6*d*g^2*k + 480*a^3*b^3
*c^6*d*e^2*m + 144*a^2*b^8*c^2*d*f*l^2 + 96*a^4*b^3*c^5*d*h*j^2 + 96*a^3*b^
3*c^6*e^2*f*k - 80*a^3*b^4*c^5*d*g^2*k + 6848*a^3*b^2*c^7*d^2*e*l - 3552*a^
3*b^2*c^7*d^2*f*k - 2448*a^2*b^4*c^6*d^2*e*l + 1332*a^2*b^4*c^6*d^2*f*k + 9
60*a^3*b^2*c^7*d^2*g*j - 496*a^4*b^3*c^5*d*f*k^2 + 432*a^3*b^3*c^6*d*f^2*k
- 240*a^2*b^4*c^6*d^2*g*j - 222*a^3*b^5*c^4*d*f*k^2 - 174*a^2*b^5*c^5*d*f^2
*k + 64*a^4*b^2*c^6*f*g^2*h + 48*a^3*b^4*c^5*f*g^2*h + 42*a^2*b^7*c^3*d*f*k

^2 - 32*a^3*b^3*c^6*e*f^2*j - 320*a^3*b^2*c^7*d*e^2*k + 192*a^4*b^2*c^6*e*g
*h^2 + 192*a^4*b^2*c^6*d*f*j^2 - 32*a^3*b^4*c^5*d*f*j^2 + 16*a^3*b^4*c^5*e*
g*h^2 + 480*a^2*b^3*c^7*d^2*e*j - 224*a^3*b^3*c^6*d*g^2*h + 192*a^3*b^2*c^7
*e^2*f*h + 24*a^2*b^5*c^5*d*g^2*h - 864*a^3*b^2*c^7*d*f^2*h + 336*a^3*b^3*c
^6*d*f*h^2 + 192*a^3*b^2*c^7*e*f^2*g + 144*a^2*b^3*c^7*d^2*f*h - 30*a^2*b^5
*c^5*d*f*h^2 + 16*a^2*b^4*c^6*e*f^2*g - 12*a^2*b^4*c^6*d*f^2*h + 192*a^3*b^
2*c^7*d*f*g^2 + 96*a^2*b^3*c^7*d*e^2*h + 48*a^2*b^4*c^6*d*f*g^2 + 960*a^2*b
^2*c^8*d^2*e*g + 192*a^2*b^2*c^8*d*e^2*f - 7680*a^9*b*c^2*l^2*m^2 + 3152*a^
8*b^3*c*l^2*m^2 + 2070*a^7*b^4*c*k^2*m^2 - 1840*a^7*b^3*c^2*k^3*m + 6720*a^
8*b*c^3*j^2*m^2 - 3072*a^8*b*c^3*k^2*l^2 + 1680*a^6*b^5*c*j^2*m^2 - 100*a^6
*b^5*c*k^2*l^2 - 2176*a^7*b^3*c^2*j*l^3 - 256*a^6*b^3*c^3*j^3*l - 64*a^5*b^
6*c*j^2*l^2 - 12480*a^8*b^2*c^2*h*m^3 + 972*a^5*b^6*c*h^2*m^2 - 960*a^7*b*c
^4*j^2*k^2 - 252*a^5*b^4*c^3*h^3*m - 192*a^6*b^2*c^4*h^3*m + 54*a^4*b^6*c^2
*h^3*m + 1536*a^7*b*c^4*h^2*l^2 + 420*a^4*b^7*c*g^2*m^2 - 36*a^4*b^7*c*h^2*
l^2 - 3072*a^7*b^2*c^3*g*l^3 + 2096*a^7*b^3*c^2*f*m^3 + 1088*a^6*b^4*c^2*g*
l^3 - 496*a^6*b^3*c^3*h*k^3 - 192*a^4*b^4*c^4*g^3*l + 176*a^4*b^3*c^5*f^3*m
 + 144*a^5*b^3*c^4*h^3*k + 78*a^3*b^8*c*f^2*m^2 + 54*a^3*b^5*c^4*f^3*m + 32
*a^3*b^6*c^3*g^3*l + 30*a^5*b^5*c^2*h*k^3 - 18*a^4*b^5*c^3*h^3*k - 18*a^2*b
^7*c^3*f^3*m - 16*a^3*b^8*c*g^2*l^2 + 6720*a^6*b*c^5*e^2*m^2 - 192*a^6*b*c^
5*h^2*j^2 - 4*a^2*b^9*c*f^2*l^2 - 35040*a^7*b^2*c^3*d*m^3 + 14300*a^6*b^4*c
^2*d*m^3 - 12000*a^3*b^2*c^7*d^3*m + 4380*a^2*b^4*c^6*d^3*m - 2176*a^6*b^3*
c^3*e*l^3 - 256*a^3*b^3*c^6*e^3*l - 192*a^6*b^2*c^4*f*k^3 + 192*a^5*b^5*c^2
*e*l^3 - 192*a^4*b^2*c^6*f^3*k + 132*a^5*b^4*c^3*f*k^3 + 128*a^4*b^3*c^5*g^
3*j - 28*a^3*b^4*c^5*f^3*k - 10*a^4*b^6*c^2*f*k^3 + 6*a^2*b^6*c^4*f^3*k + 1
0752*a^5*b*c^6*d^2*l^2 - 960*a^5*b*c^6*e^2*k^2 - 192*a^5*b*c^6*f^2*j^2 + 10
8*a*b^9*c^2*d^2*l^2 - 1680*a^5*b^3*c^4*d*k^3 - 1680*a^2*b^3*c^7*d^3*k + 222
*a^4*b^5*c^3*d*k^3 + 30*a*b^8*c^3*d^2*k^2 - 10*a^3*b^7*c^2*d*k^3 - 960*a^4*
b*c^7*d^2*j^2 + 80*a^4*b^3*c^5*f*h^3 + 80*a^3*b^3*c^6*f^3*h + 6*a^3*b^5*c^4
*f*h^3 + 6*a^2*b^5*c^5*f^3*h - 192*a^4*b*c^7*e^2*h^2 - 192*a^4*b^2*c^6*d*h^
3 - 192*a^2*b^2*c^8*d^3*h + 128*a^3*b^3*c^6*e*g^3 - 28*a^3*b^4*c^5*d*h^3 +
12*a*b^6*c^5*d^2*h^2 + 6*a^2*b^6*c^4*d*h^3 - 192*a^3*b*c^8*e^2*f^2 + 60*a*b
^5*c^6*d^2*g^2 + 198*a*b^4*c^7*d^2*f^2 + 144*a^2*b^3*c^7*d*f^3 - 960*a^2*b*
c^9*d^2*e^2 + 240*a*b^3*c^8*d^2*e^2 + 15360*a^9*c^3*k*l^2*m - 12800*a^9*c^3
*j*l*m^2 - 3840*a^8*c^4*j^2*k*m + 432*a^6*b^6*j*l*m^2 + 4608*a^8*c^4*j*k^2*
l + 2880*a^8*c^4*h*k^2*m + 5120*a^8*c^4*f*l^2*m - 3072*a^8*c^4*h*k*l^2 + 27
0*a^5*b^7*h*k*m^2 - 216*a^5*b^7*g*l*m^2 - 12800*a^8*c^4*e*l*m^2 - 4800*a^8*
c^4*f*k*m^2 - 512*a^7*c^5*h^2*j*l - 3840*a^6*c^6*e^2*k*m - 1280*a^7*c^5*f*j
^2*m + 768*a^7*c^5*h*j^2*k + 144*a^4*b^8*g*j*m^2 - 90*a^4*b^8*f*k*m^2 + 864
0*a^7*c^5*d*k^2*m + 4608*a^7*c^5*e*k^2*l + 512*a^6*c^6*f^2*j*l - 9216*a^7*c
^5*d*k*l^2 - 4096*a^7*c^5*e*j*l^2 + 320*a^6*c^6*f^2*h*m - 90*a^3*b^9*d*k*m^
2 + 15200*a^9*b*c^2*k*m^3 - 6192*a^8*b^3*c*k*m^3 + 5472*a^8*b*c^3*k^3*m - 4
608*a^5*c^7*d^2*j*l - 1024*a^7*c^5*f*h*l^2 + 150*a^6*b^5*c*k^3*m + 54*a^3*b
^9*f*h*m^2 + 6*b^10*c^2*d^2*h*m - 14400*a^7*c^5*d*h*m^2 + 8640*a^5*c^7*d^2*
h*m + 2880*a^6*c^6*d*h^2*m + 2304*a^6*c^6*d*j^2*k - 512*a^6*c^6*e*h^2*l - 1
92*a^6*c^6*f*h^2*k + 6144*a^8*b*c^3*j*l^3 + 1536*a^7*b*c^4*j^3*l - 1280*a^5

*c^7*e^2*f*m + 768*a^5*c^7*e^2*h*k + 256*a^6*c^6*f*h*j^2 + 192*a^6*b^5*c*j*
l^3 + 54*a^2*b^10*d*h*m^2 - 18*b^9*c^3*d^2*f*m + 8*b^9*c^3*d^2*g*l - 2*b^9*
c^3*d^2*h*k + 4068*a^7*b^4*c*h*m^3 - 1728*a^6*c^6*d*h*k^2 + 960*a^5*c^7*d*f
^2*m + 512*a^5*c^7*e*f^2*l - 3072*a^6*c^6*d*f*l^2 - 16*b^8*c^4*d^2*e*l + 6*
b^8*c^4*d^2*f*k - 4608*a^4*c^8*d^2*e*l + 2400*a^8*b*c^3*f*m^3 + 2016*a^7*b*
c^4*h*k^3 - 1728*a^4*c^8*d^2*f*k - 1146*a^6*b^5*c*f*m^3 + 224*a^6*b*c^5*h^3
*k - 96*a^5*b^6*c*g*l^3 + 96*a^5*b*c^6*f^3*m + 2304*a^4*c^8*d*e^2*k + 768*a
^5*c^7*d*f*j^2 + 6144*a^7*b*c^4*e*l^3 - 2280*a^5*b^6*c*d*m^3 + 1536*a^4*b*c
^7*e^3*l - 616*a*b^6*c^5*d^3*m + 512*a^6*b*c^5*g*j^3 + 256*a^4*c^8*e^2*f*h
+ 240*a*b^10*c*d^2*m^2 + 6*b^7*c^5*d^2*f*h - 192*a^4*c^8*d*f^2*h + 4320*a^6
*b*c^5*d*k^3 + 4320*a^3*b*c^8*d^3*k + 222*a*b^5*c^6*d^3*k + 16*b^6*c^6*d^2*
e*g + 96*a^5*b*c^6*f*h^3 + 96*a^4*b*c^7*f^3*h + 768*a^3*c^9*d*e^2*f + 512*a
^3*b*c^8*e^3*g + 132*a*b^4*c^7*d^3*h + 2016*a^2*b*c^9*d^3*f - 496*a*b^3*c^8
*d^3*f + 224*a^3*b*c^8*d*f^3 - 18*a*b^5*c^6*d*f^3 - 3264*a^8*b^2*c^2*k^2*m^
2 - 6160*a^7*b^3*c^2*j^2*m^2 + 1104*a^7*b^3*c^2*k^2*l^2 - 1920*a^7*b^2*c^3*
j^2*l^2 + 768*a^6*b^4*c^2*j^2*l^2 + 3888*a^7*b^2*c^3*h^2*m^2 - 3510*a^6*b^4
*c^2*h^2*m^2 + 240*a^6*b^3*c^3*j^2*k^2 - 16*a^5*b^5*c^2*j^2*k^2 + 1680*a^6*
b^3*c^3*g^2*m^2 - 1648*a^6*b^3*c^3*h^2*l^2 - 1540*a^5*b^5*c^2*g^2*m^2 + 444
*a^5*b^5*c^2*h^2*l^2 - 960*a^6*b^2*c^4*h^2*k^2 - 576*a^6*b^2*c^4*f^2*m^2 -
512*a^6*b^2*c^4*g^2*l^2 - 480*a^5*b^4*c^3*g^2*l^2 + 198*a^5*b^4*c^3*h^2*k^2
 + 192*a^4*b^6*c^2*g^2*l^2 - 186*a^5*b^4*c^3*f^2*m^2 - 97*a^4*b^6*c^2*f^2*m
^2 - 9*a^4*b^6*c^2*h^2*k^2 - 6160*a^5*b^3*c^4*e^2*m^2 + 1680*a^4*b^5*c^3*e^
2*m^2 - 240*a^5*b^3*c^4*g^2*k^2 - 240*a^5*b^3*c^4*f^2*l^2 - 144*a^3*b^7*c^2
*e^2*m^2 + 60*a^4*b^5*c^3*g^2*k^2 - 36*a^4*b^5*c^3*f^2*l^2 + 36*a^3*b^7*c^2
*f^2*l^2 - 16*a^5*b^3*c^4*h^2*j^2 - 4*a^3*b^7*c^2*g^2*k^2 + 38512*a^5*b^2*c
^5*d^2*m^2 - 32310*a^4*b^4*c^4*d^2*m^2 + 12720*a^3*b^6*c^3*d^2*m^2 - 2500*a
^2*b^8*c^2*d^2*m^2 - 1920*a^5*b^2*c^5*e^2*l^2 + 768*a^4*b^4*c^4*e^2*l^2 - 4
64*a^5*b^2*c^5*f^2*k^2 - 384*a^5*b^2*c^5*g^2*j^2 - 64*a^3*b^6*c^3*e^2*l^2 +
 42*a^4*b^4*c^4*f^2*k^2 + 12*a^3*b^6*c^3*f^2*k^2 - 13104*a^4*b^3*c^5*d^2*l^
2 + 5628*a^3*b^5*c^4*d^2*l^2 - 1128*a^2*b^7*c^3*d^2*l^2 + 240*a^4*b^3*c^5*e
^2*k^2 - 16*a^4*b^3*c^5*f^2*j^2 - 16*a^3*b^5*c^4*e^2*k^2 - 2880*a^4*b^2*c^6
*d^2*k^2 + 1750*a^3*b^4*c^5*d^2*k^2 - 345*a^2*b^6*c^4*d^2*k^2 - 48*a^4*b^3*
c^5*g^2*h^2 - 4*a^3*b^5*c^4*g^2*h^2 + 240*a^3*b^3*c^6*d^2*j^2 - 192*a^4*b^2
*c^6*f^2*h^2 - 42*a^3*b^4*c^5*f^2*h^2 - 16*a^2*b^5*c^5*d^2*j^2 - 48*a^3*b^3
*c^6*f^2*g^2 - 16*a^3*b^3*c^6*e^2*h^2 - 4*a^2*b^5*c^5*f^2*g^2 - 464*a^3*b^2
*c^7*d^2*h^2 - 384*a^3*b^2*c^7*e^2*g^2 + 42*a^2*b^4*c^6*d^2*h^2 - 240*a^2*b
^3*c^7*d^2*g^2 - 16*a^2*b^3*c^7*e^2*f^2 - 960*a^2*b^2*c^8*d^2*f^2 + 6*b^11*
c*d^2*k*m - 18*a*b^11*d*f*m^2 - 7200*a^9*c^3*k^2*m^2 - 324*a^7*b^5*l^2*m^2
- 225*a^6*b^6*k^2*m^2 - 2048*a^8*c^4*j^2*l^2 - 144*a^5*b^7*j^2*m^2 - 2400*a
^8*c^4*h^2*m^2 - 81*a^4*b^8*h^2*m^2 - 800*a^7*c^5*f^2*m^2 - 288*a^7*c^5*h^2
*k^2 - 36*a^3*b^9*g^2*m^2 - 9*a^2*b^10*f^2*m^2 - 21600*a^6*c^6*d^2*m^2 - 20
48*a^6*c^6*e^2*l^2 - 864*a^6*c^6*f^2*k^2 - 2592*a^5*c^7*d^2*k^2 - 1536*a^5*
c^7*e^2*j^2 + 1536*a^8*b^2*c^2*l^4 - 32*a^5*c^7*f^2*h^2 + 360*a^7*b^2*c^3*k
^4 - 25*a^6*b^4*c^2*k^4 - 864*a^4*c^8*d^2*h^2 - 4*b^7*c^5*d^2*g^2 - 9*b^6*c
^6*d^2*f^2 - 288*a^3*c^9*d^2*f^2 - 24*a^5*b^2*c^5*h^4 - 16*b^5*c^7*d^2*e^2

```
- 9*a^4*b^4*c^4*h^4 - 16*a^3*b^4*c^5*g^4 - 24*a^3*b^2*c^7*f^4 - 9*a^2*b^4*c
^6*f^4 - a^2*b^8*c^2*f^2*k^2 - a^2*b^6*c^4*f^2*h^2 + 630*a^7*b^5*k*m^3 + 80
00*a^9*c^3*h*m^3 + 320*a^7*c^5*h^3*m - 378*a^6*b^6*h*m^3 + 126*a^5*b^7*f*m^
3 + 30*b^8*c^4*d^3*m + 24000*a^8*c^4*d*m^3 + 8640*a^4*c^8*d^3*m - 1728*a^7*
c^5*f*k^3 - 192*a^5*c^7*f^3*k - 4*b^11*c*d^2*l^2 + 126*a^4*b^8*d*m^3 - 10*b
^7*c^5*d^3*k + 4200*a^9*b^2*c*m^4 - 1024*a^6*c^6*e*j^3 - 1024*a^4*c^8*e^3*j
 - 144*a^7*b^4*c*l^4 - 10*b^6*c^6*d^3*h - 1728*a^3*c^9*d^3*h - 192*a^5*c^7*
d*h^3 + 30*b^5*c^7*d^3*f + 360*a*b^2*c^9*d^4 - 9*b^12*d^2*m^2 - 10000*a^10*
c^2*m^4 - 4096*a^9*c^3*l^4 - 441*a^8*b^4*m^4 - 1296*a^8*c^4*k^4 - 256*a^7*c
^5*j^4 - 16*a^6*c^6*h^4 - 16*a^4*c^8*f^4 - 256*a^3*c^9*e^4 - 25*b^4*c^8*d^4
 - 1296*a^2*c^10*d^4 - b^10*c^2*d^2*k^2 - b^8*c^4*d^2*h^2, z, k1), k1, 1, 4
) + ((b*c^2*e - 2*a*c^2*g - a*b^2*l + 2*a^2*c*l + a*b*c*j)/(2*(4*a*c - b^2)
) + (x^2*(2*c^3*e - b^3*l - b*c^2*g - 2*a*c^2*j + b^2*c*j + 3*a*b*c*l))/(2*
(4*a*c - b^2)) + (x*(2*a*c^3*d - 2*a^2*c^2*h - a^2*b^2*m - b^2*c^2*d + 2*a^
3*c*m + a*b*c^2*f + a^2*b*c*k))/(2*a*(4*a*c - b^2)) - (x^3*(2*a^2*c^2*k + b
*c^3*d - 2*a*c^3*f + a*b^3*m + a*b*c^2*h - a*b^2*c*k - 3*a^2*b*c*m))/(2*a*(
4*a*c - b^2)))/(a*c^2 + c^3*x^4 + b*c^2*x^2) + (m*x)/c^2
```

3.42 $\int \frac{d+ex}{(4-5x^2+x^4)^3} dx$

Optimal result

Integrand size = 18, antiderivative size = 143

$$\int \frac{d+ex}{(4-5x^2+x^4)^3} dx = \frac{dx(17-5x^2)}{144(4-5x^2+x^4)^2} + \frac{e(5-2x^2)}{36(4-5x^2+x^4)^2} - \frac{dx(59-35x^2)}{3456(4-5x^2+x^4)} - \frac{e(5-2x^2)}{54(4-5x^2+x^4)} - \frac{313d\text{arctanh}\left(\frac{x}{2}\right)}{20736} + \frac{13}{648}d\text{arctanh}(x) - \frac{1}{81}e\log\left(1-x^2\right) + \frac{1}{81}e\log\left(4-x^2\right)$$

```
[Out] 1/144*d*x*(-5*x^2+17)/(x^4-5*x^2+4)^2+1/36*e*(-2*x^2+5)/(x^4-5*x^2+4)^2-1/3
456*d*x*(-35*x^2+59)/(x^4-5*x^2+4)-1/54*e*(-2*x^2+5)/(x^4-5*x^2+4)-313/2073
6*d*arctanh(1/2*x)+13/648*d*arctanh(x)-1/81*e*ln(-x^2+1)+1/81*e*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 143, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.556, Rules used = {1687, 12, 1106, 1192, 1180, 213, 1121, 628, 630, 31}

$$\int \frac{d+ex}{(4-5x^2+x^4)^3} dx = -\frac{313d\text{arctanh}\left(\frac{x}{2}\right)}{20736} + \frac{13}{648}d\text{arctanh}(x) - \frac{dx(59-35x^2)}{3456(x^4-5x^2+4)} + \frac{dx(17-5x^2)}{144(x^4-5x^2+4)^2} - \frac{1}{81}e\log\left(1-x^2\right) + \frac{1}{81}e\log\left(4-x^2\right) - \frac{e(5-2x^2)}{54(x^4-5x^2+4)} + \frac{e(5-2x^2)}{36(x^4-5x^2+4)^2}$$

```
[In] Int[(d + e*x)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (d*x*(17 - 5*x^2))/(144*(4 - 5*x^2 + x^4)^2) + (e*(5 - 2*x^2))/(36*(4 - 5*x
^2 + x^4)^2) - (d*x*(59 - 35*x^2))/(3456*(4 - 5*x^2 + x^4)) - (e*(5 - 2*x^2
))/(54*(4 - 5*x^2 + x^4)) - (313*d*ArcTanh[x/2])/20736 + (13*d*ArcTanh[x])/
648 - (e*Log[1 - x^2])/81 + (e*Log[4 - x^2])/81
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 && !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d}{\left(4-5x^2+x^4\right)^3}\,dx+\int \frac{ex}{\left(4-5x^2+x^4\right)^3}\,dx\\
&= d\int \frac{1}{\left(4-5x^2+x^4\right)^3}\,dx+e\int \frac{x}{\left(4-5x^2+x^4\right)^3}\,dx\\
&= \frac{dx(17-5x^2)}{144\left(4-5x^2+x^4\right)^2}-\frac{1}{144}d\int \frac{-19+25x^2}{\left(4-5x^2+x^4\right)^2}\,dx+\frac{1}{2}e\text{Subst}\left(\int \frac{1}{\left(4-5x+x^2\right)^3}\,dx,x,x^2\right)\\
&= \frac{dx(17-5x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2}-\frac{dx(59-35x^2)}{3456\left(4-5x^2+x^4\right)}\\
&\quad+\frac{d\int \frac{519+105x^2}{4-5x^2+x^4}\,dx}{10368}-\frac{1}{6}e\text{Subst}\left(\int \frac{1}{\left(4-5x+x^2\right)^2}\,dx,x,x^2\right)\\
&= \frac{dx(17-5x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2}-\frac{dx(59-35x^2)}{3456\left(4-5x^2+x^4\right)}-\frac{e(5-2x^2)}{54\left(4-5x^2+x^4\right)}\\
&\quad-\frac{1}{648}(13d)\int \frac{1}{-1+x^2}\,dx+\frac{(313d)\int \frac{1}{-4+x^2}\,dx}{10368}+\frac{1}{27}e\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx,x,x^2\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{dx(17-5x^2)}{144\,(4-5x^2+x^4)^2} + \frac{e(5-2x^2)}{36\,(4-5x^2+x^4)^2} - \frac{dx(59-35x^2)}{3456\,(4-5x^2+x^4)} \\
&\quad - \frac{e(5-2x^2)}{54\,(4-5x^2+x^4)} - \frac{313d\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{13}{648}d\tanh^{-1}(x) \\
&\quad + \frac{1}{81}e\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) - \frac{1}{81}e\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) \\
&= \frac{dx(17-5x^2)}{144\,(4-5x^2+x^4)^2} + \frac{e(5-2x^2)}{36\,(4-5x^2+x^4)^2} - \frac{dx(59-35x^2)}{3456\,(4-5x^2+x^4)} - \frac{e(5-2x^2)}{54\,(4-5x^2+x^4)} \\
&\quad - \frac{313d\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{13}{648}d\tanh^{-1}(x) - \frac{1}{81}e\log\left(1-x^2\right) + \frac{1}{81}e\log\left(4-x^2\right)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.07 (sec) , antiderivative size = 128, normalized size of antiderivative = 0.90

$$
\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx
= \frac{\frac{288(e(20-8x^2)+dx(17-5x^2))}{(4-5x^2+x^4)^2} + \frac{12(64e(-5+2x^2)+dx(-59+35x^2))}{4-5x^2+x^4} - 32(13d+16e)\log(1-x) + (313d+512e)\log(2-x}{41472}
$$

```
[In] Integrate[(d + e*x)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] ((288*(e*(20 - 8*x^2) + d*x*(17 - 5*x^2)))/(4 - 5*x^2 + x^4)^2 + (12*(64*e*
(-5 + 2*x^2) + d*x*(-59 + 35*x^2)))/(4 - 5*x^2 + x^4) - 32*(13*d + 16*e)*Lo
g[1 - x] + (313*d + 512*e)*Log[2 - x] + 32*(13*d - 16*e)*Log[1 + x] + (-313
*d + 512*e)*Log[2 + x])/41472
```

Maple [A] (verified)

Time = 0.09 (sec) , antiderivative size = 107, normalized size of antiderivative = 0.75

method	result
norman	$\frac{\frac{5}{9}e\,x^2+\frac{1}{27}e\,x^6-\frac{5}{18}e\,x^4+\frac{43}{864}dx+\frac{35}{3456}d\,x^7+\frac{35}{384}x^3d-\frac{13}{192}x^5d-\frac{25}{108}e}{(x^4-5x^2+4)^2}+\left(-\frac{313d}{41472}+\frac{e}{81}\right)\ln\left(x+2\right)+\left(-\frac{13d}{1296}-\frac{e}{81}\right)\ln$
risch	$\frac{\frac{5}{9}e\,x^2+\frac{1}{27}e\,x^6-\frac{5}{18}e\,x^4+\frac{43}{864}dx+\frac{35}{3456}d\,x^7+\frac{35}{384}x^3d-\frac{13}{192}x^5d-\frac{25}{108}e}{(x^4-5x^2+4)^2}+\frac{13\ln(x+1)d}{1296}-\frac{\ln(x+1)e}{81}-\frac{313\ln(x+2)d}{41472}+\frac{\ln(x+2)}{81}$
default	$\left(-\frac{313d}{41472}+\frac{e}{81}\right)\ln\left(x+2\right)-\frac{-\frac{19d}{6912}+\frac{17e}{3456}}{x+2}-\frac{-\frac{d}{1728}+\frac{e}{864}}{2(x+2)^2}-\frac{-\frac{d}{432}+\frac{e}{144}}{x+1}-\frac{\frac{d}{216}-\frac{e}{216}}{2(x+1)^2}+\left(\frac{13d}{1296}-\frac{e}{81}\right)\ln\left(x+\right.$
parallelrisch	$\frac{1536e\,x^6-11520e\,x^4-9600e+420d\,x^7+2064dx+5008\ln(x-2)d+8192\ln(x-2)e-6656\ln(x-1)d-8192\ln(x-1)e-4160\ln(x+1)x}{}$

```
[In] int((e*x+d)/(x^4-5*x^2+4)^3,x,method=_RETURNVERBOSE)
```

```
[Out] (5/9*e*x^2+1/27*e*x^6-5/18*e*x^4+43/864*d*x+35/3456*d*x^7+35/384*x^3*d-13/1
92*x^5*d-25/108*e)/(x^4-5*x^2+4)^2+(-313/41472*d+1/81*e)*ln(x+2)+(-13/1296*
d-1/81*e)*ln(x-1)+(13/1296*d-1/81*e)*ln(x+1)+(313/41472*d+1/81*e)*ln(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 307 vs. $2(125) = 250$.

Time = 0.29 (sec) , antiderivative size = 307, normalized size of antiderivative = 2.15

$$\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx$$
$$= \frac{420\,dx^7 + 1536\,ex^6 - 2808\,dx^5 - 11520\,ex^4 + 3780\,dx^3 + 23040\,ex^2 + 2064\,dx - ((313\,d - 512\,e)x^8 - 10}{}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="fricas")
```

```
[Out] 1/41472*(420*d*x^7 + 1536*e*x^6 - 2808*d*x^5 - 11520*e*x^4 + 3780*d*x^3 + 2
3040*e*x^2 + 2064*d*x - ((313*d - 512*e)*x^8 - 10*(313*d - 512*e)*x^6 + 33*
(313*d - 512*e)*x^4 - 40*(313*d - 512*e)*x^2 + 5008*d - 8192*e)*log(x + 2)
+ 32*((13*d - 16*e)*x^8 - 10*(13*d - 16*e)*x^6 + 33*(13*d - 16*e)*x^4 - 40*
(13*d - 16*e)*x^2 + 208*d - 256*e)*log(x + 1) - 32*((13*d + 16*e)*x^8 - 10*
(13*d + 16*e)*x^6 + 33*(13*d + 16*e)*x^4 - 40*(13*d + 16*e)*x^2 + 208*d + 2
56*e)*log(x - 1) + ((313*d + 512*e)*x^8 - 10*(313*d + 512*e)*x^6 + 33*(313*
d + 512*e)*x^4 - 40*(313*d + 512*e)*x^2 + 5008*d + 8192*e)*log(x - 2) - 960
0*e)/(x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 668 vs. $2(126) = 252$.

Time = 2.23 (sec) , antiderivative size = 668, normalized size of antiderivative = 4.67

$$\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx$$
$$= \frac{(13d-16e)\log\left(x+\frac{-1106258459719280d^4e-13113710954343d^4\cdot(13d-16e)-817263343042560d^2e^3+153628968222720d^2e^2\cdot(13d-16e)+\cdots}{229\cdots}\right.}{\cdots}$$
$$-\frac{(13d+16e)\log\left(x+\frac{-1106258459719280d^4e+13113710954343d^4\cdot(13d+16e)-817263343042560d^2e^3-153628968222720d^2e^2\cdot(13d+16\cdots}{\cdots}\right.}{\cdots}$$
$$-\frac{(313d-512e)\log\left(x+\frac{-1106258459719280d^4e+\frac{13113710954343d^4\cdot(313d-512e)}{32}-817263343042560d^2e^3-4800905256960d^2e^2\cdot(313d-5\cdots}{2\cdots}\right.}{\cdots}$$
$$+\frac{(313d+512e)\log\left(x+\frac{-1106258459719280d^4e-\frac{13113710954343d^4\cdot(313d+512e)}{32}-817263343042560d^2e^3+4800905256960d^2e^2\cdot(313d+5\cdots}{2\cdots}\right.}{\cdots}$$
$$+\frac{35dx^7-234dx^5+315dx^3+172dx+128ex^6-960ex^4+1920ex^2-800e}{3456x^8-34560x^6+114048x^4-138240x^2+55296}$$

```
[In] integrate((e*x+d)/(x**4-5*x**2+4)**3,x)
```

```
[Out] (13*d - 16*e)*log(x + (-1106258459719280*d**4*e - 13113710954343*d**4*(13*d
 - 16*e) - 817263343042560*d**2*e**3 + 153628968222720*d**2*e**2*(13*d - 16
*e) + 9530197557248*d**2*e*(13*d - 16*e)**2 + 88038005760*d**2*(13*d - 16*e
)**3 + 5035763255214080*e**5 + 142661633703936*e**4*(13*d - 16*e) - 1967095
0215680*e**3*(13*d - 16*e)**2 - 557272006656*e**2*(13*d - 16*e)**3)/(229412
56248261*d**5 - 2312740746035200*d**3*e**2 + 4473912813420544*d*e**4))/1296
 - (13*d + 16*e)*log(x + (-1106258459719280*d**4*e + 13113710954343*d**4*(1
3*d + 16*e) - 817263343042560*d**2*e**3 - 153628968222720*d**2*e**2*(13*d +
 16*e) + 9530197557248*d**2*e*(13*d + 16*e)**2 - 88038005760*d**2*(13*d + 1
6*e)**3 + 5035763255214080*e**5 - 142661633703936*e**4*(13*d + 16*e) - 1967
0950215680*e**3*(13*d + 16*e)**2 + 557272006656*e**2*(13*d + 16*e)**3)/(229
41256248261*d**5 - 2312740746035200*d**3*e**2 + 4473912813420544*d*e**4))/1
296 - (313*d - 512*e)*log(x + (-1106258459719280*d**4*e + 13113710954343*d*
*4*(313*d - 512*e)/32 - 817263343042560*d**2*e**3 - 4800905256960*d**2*e**2
*(313*d - 512*e) + 9306833552*d**2*e*(313*d - 512*e)**2 - 85974615*d**2*(31
3*d - 512*e)**3/32 + 5035763255214080*e**5 - 4458176053248*e**4*(313*d - 51
2*e) - 19209912320*e**3*(313*d - 512*e)**2 + 17006592*e**2*(313*d - 512*e)*
*3)/(22941256248261*d**5 - 2312740746035200*d**3*e**2 + 4473912813420544*d*
e**4))/41472 + (313*d + 512*e)*log(x + (-1106258459719280*d**4*e - 13113710
954343*d**4*(313*d + 512*e)/32 - 817263343042560*d**2*e**3 + 4800905256960*
d**2*e**2*(313*d + 512*e) + 9306833552*d**2*e*(313*d + 512*e)**2 + 85974615
*d**2*(313*d + 512*e)**3/32 + 5035763255214080*e**5 + 4458176053248*e**4*(3
13*d + 512*e) - 19209912320*e**3*(313*d + 512*e)**2 - 17006592*e**2*(313*d
+ 512*e)**3)/(22941256248261*d**5 - 2312740746035200*d**3*e**2 + 4473912813
```

```
420544*d*e**4))/41472 + (35*d*x**7 - 234*d*x**5 + 315*d*x**3 + 172*d*x + 12
8*e*x**6 - 960*e*x**4 + 1920*e*x**2 - 800*e)/(3456*x**8 - 34560*x**6 + 1140
48*x**4 - 138240*x**2 + 55296)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 121, normalized size of antiderivative = 0.85

$$\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx = -\frac{1}{41472}\,(313\,d-512\,e)\log{(x+2)}+\frac{1}{1296}\,(13\,d-16\,e)\log{(x+1)} - \frac{1}{1296}\,(13\,d+16\,e)\log{(x-1)}+\frac{1}{41472}\,(313\,d+512\,e)\log{(x-2)} + \frac{35\,dx^7+128\,ex^6-234\,dx^5-960\,ex^4+315\,dx^3+1920\,ex^2+172\,dx-800\,e}{3456\,(x^8-10\,x^6+33\,x^4-40\,x^2+16)}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="maxima")
```

```
[Out] -1/41472*(313*d - 512*e)*log(x + 2) + 1/1296*(13*d - 16*e)*log(x + 1) - 1/1
296*(13*d + 16*e)*log(x - 1) + 1/41472*(313*d + 512*e)*log(x - 2) + 1/3456*
(35*d*x^7 + 128*e*x^6 - 234*d*x^5 - 960*e*x^4 + 315*d*x^3 + 1920*e*x^2 + 17
2*d*x - 800*e)/(x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 115, normalized size of antiderivative = 0.80

$$\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx = -\frac{1}{41472}\,(313\,d-512\,e)\log{(|x+2|)}+\frac{1}{1296}\,(13\,d-16\,e)\log{(|x+1|)} - \frac{1}{1296}\,(13\,d+16\,e)\log{(|x-1|)}+\frac{1}{41472}\,(313\,d+512\,e)\log{(|x-2|)} + \frac{35\,dx^7+128\,ex^6-234\,dx^5-960\,ex^4+315\,dx^3+1920\,ex^2+172\,dx-800\,e}{3456\,(x^4-5\,x^2+4)^2}$$

```
[In] integrate((e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="giac")
```

```
[Out] -1/41472*(313*d - 512*e)*log(abs(x + 2)) + 1/1296*(13*d - 16*e)*log(abs(x +
 1)) - 1/1296*(13*d + 16*e)*log(abs(x - 1)) + 1/41472*(313*d + 512*e)*log(a
bs(x - 2)) + 1/3456*(35*d*x^7 + 128*e*x^6 - 234*d*x^5 - 960*e*x^4 + 315*d*x
^3 + 1920*e*x^2 + 172*d*x - 800*e)/(x^4 - 5*x^2 + 4)^2
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 118, normalized size of antiderivative = 0.83

$$\int \frac{d+ex}{(4-5x^2+x^4)^3}\,dx = \ln(x+1)\left(\frac{13\,d}{1296}-\frac{e}{81}\right)-\ln(x-1)\left(\frac{13\,d}{1296}+\frac{e}{81}\right) + \ln(x-2)\left(\frac{313\,d}{41472}+\frac{e}{81}\right)-\ln(x+2)\left(\frac{313\,d}{41472}-\frac{e}{81}\right) + \frac{\frac{35\,d\,x^7}{3456}+\frac{e\,x^6}{27}-\frac{13\,d\,x^5}{192}-\frac{5\,e\,x^4}{18}+\frac{35\,d\,x^3}{384}+\frac{5\,e\,x^2}{9}+\frac{43\,d\,x}{864}-\frac{25\,e}{108}}{x^8-10\,x^6+33\,x^4-40\,x^2+16}$$

```
[In] int((d + e*x)/(x^4 - 5*x^2 + 4)^3,x)
```

```
[Out] log(x + 1)*((13*d)/1296 - e/81) - log(x - 1)*((13*d)/1296 + e/81) + log(x -
 2)*((313*d)/41472 + e/81) - log(x + 2)*((313*d)/41472 - e/81) + ((43*d*x)/
864 - (25*e)/108 + (35*d*x^3)/384 - (13*d*x^5)/192 + (35*d*x^7)/3456 + (5*e
*x^2)/9 - (5*e*x^4)/18 + (e*x^6)/27)/(33*x^4 - 40*x^2 - 10*x^6 + x^8 + 16)
```

3.43 $\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3} dx$

Optimal result

Integrand size = 23, antiderivative size = 175

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3} dx = \frac{e(5-2x^2)}{36(4-5x^2+x^4)^2} + \frac{x(17d+20f-(5d+8f)x^2)}{144(4-5x^2+x^4)^2} - \frac{e(5-2x^2)}{54(4-5x^2+x^4)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456(4-5x^2+x^4)} - \frac{(313d+820f)\operatorname{arctanh}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\operatorname{arctanh}(x) - \frac{1}{81}e\log(1-x^2) + \frac{1}{81}e\log(4-x^2)$$

```
[Out] 1/36*e*(-2*x^2+5)/(x^4-5*x^2+4)^2+1/144*x*(17*d+20*f-(5*d+8*f)*x^2)/(x^4-5*
x^2+4)^2-1/54*e*(-2*x^2+5)/(x^4-5*x^2+4)-1/3456*x*(59*d+380*f-35*(d+4*f)*x^
2)/(x^4-5*x^2+4)-1/20736*(313*d+820*f)*arctanh(1/2*x)+1/648*(13*d+25*f)*arc
tanh(x)-1/81*e*ln(-x^2+1)+1/81*e*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 175, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.391, Rules used

= {1687, 1192, 1180, 213, 12, 1121, 628, 630, 31}

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx = -\frac{\operatorname{arctanh}\left(\frac{x}{2}\right)(313d+820f)}{20736} + \frac{1}{648}\operatorname{arctanh}(x)(13d+25f) - \frac{x(-35x^2(d+4f)+59d+380f)}{3456\,(x^4-5x^2+4)} + \frac{x(-(x^2(5d+8f))+17d+20f)}{144\,(x^4-5x^2+4)^2} - \frac{1}{81}e\log\left(1-x^2\right) + \frac{1}{81}e\log\left(4-x^2\right) - \frac{e(5-2x^2)}{54\,(x^4-5x^2+4)} + \frac{e(5-2x^2)}{36\,(x^4-5x^2+4)^2}$$

```
[In] Int[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (e*(5 - 2*x^2))/(36*(4 - 5*x^2 + x^4)^2) + (x*(17*d + 20*f - (5*d + 8*f)*x^
2))/(144*(4 - 5*x^2 + x^4)^2) - (e*(5 - 2*x^2))/(54*(4 - 5*x^2 + x^4)) - (x
*(59*d + 380*f - 35*(d + 4*f)*x^2))/(3456*(4 - 5*x^2 + x^4)) - ((313*d + 82
0*f)*ArcTanh[x/2])/20736 + ((13*d + 25*f)*ArcTanh[x])/648 - (e*Log[1 - x^2]
)/81 + (e*Log[4 - x^2])/81
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
```

```
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{ex}{(4-5x^2+x^4)^3}\,dx + \int \frac{d+fx^2}{(4-5x^2+x^4)^3}\,dx \\ &= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} \\ &\quad - \frac{1}{144}\int \frac{-19d+20f+5(5d+8f)x^2}{(4-5x^2+x^4)^2}\,dx + e\int \frac{x}{(4-5x^2+x^4)^3}\,dx\end{aligned}$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\left(4-5x^2+x^4\right)^2} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\left(4-5x^2+x^4\right)} + \frac{\int \frac{3(173d+260f)+105(d+4f)x^2}{4-5x^2+x^4}\,dx}{10368} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{\left(4-5x+x^2\right)^3}\,dx, x, x^2\right)$$

$$= \frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2} + \frac{x(17d+20f-(5d+8f)x^2)}{144\left(4-5x^2+x^4\right)^2} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\left(4-5x^2+x^4\right)} - \frac{1}{6}e\text{Subst}\left(\int \frac{1}{\left(4-5x+x^2\right)^2}\,dx, x, x^2\right) + \frac{1}{648}(-13d-25f)\int \frac{1}{-1+x^2}\,dx + \frac{(313d+820f)\int \frac{1}{-4+x^2}\,dx}{10368}$$

$$= \frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2} + \frac{x(17d+20f-(5d+8f)x^2)}{144\left(4-5x^2+x^4\right)^2} - \frac{e(5-2x^2)}{54\left(4-5x^2+x^4\right)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\left(4-5x^2+x^4\right)} - \frac{(313d+820f)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\tanh^{-1}(x) + \frac{1}{27}e\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right)$$

$$= \frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2} + \frac{x(17d+20f-(5d+8f)x^2)}{144\left(4-5x^2+x^4\right)^2} - \frac{e(5-2x^2)}{54\left(4-5x^2+x^4\right)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\left(4-5x^2+x^4\right)} - \frac{(313d+820f)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\tanh^{-1}(x) + \frac{1}{81}e\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right) - \frac{1}{81}e\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right)$$

$$= \frac{e(5-2x^2)}{36\left(4-5x^2+x^4\right)^2} + \frac{x(17d+20f-(5d+8f)x^2)}{144\left(4-5x^2+x^4\right)^2} - \frac{e(5-2x^2)}{54\left(4-5x^2+x^4\right)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\left(4-5x^2+x^4\right)} - \frac{(313d+820f)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\tanh^{-1}(x) - \frac{1}{81}e\log\left(1-x^2\right) + \frac{1}{81}e\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.09 (sec) , antiderivative size = 161, normalized size of antiderivative = 0.92

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx = \frac{\frac{288(17dx+20fx-5dx^3-8fx^3+e(20-8x^2))}{(4-5x^2+x^4)^2}+\frac{12(64e(-5+2x^2)+20fx(-19+7x^2)+dx(-59+35x^2))}{4-5x^2+x^4}-32(13d+16e+25f)\log(1}{}$$

```
[In] Integrate[(d + e*x + f*x^2)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] ((288*(17*d*x + 20*f*x - 5*d*x^3 - 8*f*x^3 + e*(20 - 8*x^2)))/(4 - 5*x^2 +
x^4)^2 + (12*(64*e*(-5 + 2*x^2) + 20*f*x*(-19 + 7*x^2) + d*x*(-59 + 35*x^2)
))/(4 - 5*x^2 + x^4) - 32*(13*d + 16*e + 25*f)*Log[1 - x] + (313*d + 512*e
+ 820*f)*Log[2 - x] + 32*(13*d - 16*e + 25*f)*Log[1 + x] + (-313*d + 512*e
- 820*f)*Log[2 + x])/41472
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 139, normalized size of antiderivative = 0.79

method	result
norman	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}\right)x+\frac{5e\,x^2}{9}+\frac{e\,x^6}{27}-\frac{5e\,x^4}{18}-\frac{25e}{108}}{(x^4-5x^2+4)^2}+\left(-\frac{313d}{41472}+\frac{e}{81}-\frac{205f}{10368}\right)$
risch	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}\right)x+\frac{5e\,x^2}{9}+\frac{e\,x^6}{27}-\frac{5e\,x^4}{18}-\frac{25e}{108}}{(x^4-5x^2+4)^2}+\frac{313\ln(2-x)d}{41472}+\frac{\ln(2-x)e}{81}+$
default	$\left(-\frac{313d}{41472}+\frac{e}{81}-\frac{205f}{10368}\right)\ln(x+2)-\frac{-\frac{19d}{6912}+\frac{17e}{3456}-\frac{5f}{576}}{x+2}-\frac{-\frac{d}{1728}+\frac{e}{864}-\frac{f}{432}}{2(x+2)^2}-\frac{-\frac{d}{432}+\frac{e}{144}-\frac{5f}{432}}{x+1}-\frac{\frac{d}{216}-\frac{e}{216}+}{2(x+1)}$
parallelrisch	$\frac{-12960f\,x^5+1536e\,x^6-11520e\,x^4-9600e+27216f\,x^3+420d\,x^7+1680f\,x^7+2064dx+5008\ln(x-2)d+8192\ln(x-2)e-6656\ln(}{}$

```
[In] int((f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x,method=_RETURNVERBOSE)
```

```
[Out] ((-13/192*d-5/16*f)*x^5+(35/384*d+21/32*f)*x^3+(35/3456*d+35/864*f)*x^7+(43
/864*d-65/216*f)*x+5/9*e*x^2+1/27*e*x^6-5/18*e*x^4-25/108*e)/(x^4-5*x^2+4)^
2+(-313/41472*d+1/81*e-205/10368*f)*ln(x+2)+(-13/1296*d-1/81*e-25/1296*f)*l
n(x-1)+(13/1296*d-1/81*e+25/1296*f)*ln(x+1)+(313/41472*d+1/81*e+205/10368*f
)*ln(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 389 vs. $2(156) = 312$.

Time = 0.32 (sec) , antiderivative size = 389, normalized size of antiderivative = 2.22

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx = \frac{420\,(d+4\,f)x^7+1536\,ex^6-216\,(13\,d+60\,f)x^5-11520\,ex^4+756\,(5\,d+36\,f)x^3+23040\,ex^2+48\,(43\,d\cdots}{}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="fricas")
```

```
[Out] 1/41472*(420*(d + 4*f)*x^7 + 1536*e*x^6 - 216*(13*d + 60*f)*x^5 - 11520*e*x
^4 + 756*(5*d + 36*f)*x^3 + 23040*e*x^2 + 48*(43*d - 260*f)*x - ((313*d - 5
12*e + 820*f)*x^8 - 10*(313*d - 512*e + 820*f)*x^6 + 33*(313*d - 512*e + 82
0*f)*x^4 - 40*(313*d - 512*e + 820*f)*x^2 + 5008*d - 8192*e + 13120*f)*log(
x + 2) + 32*((13*d - 16*e + 25*f)*x^8 - 10*(13*d - 16*e + 25*f)*x^6 + 33*(1
3*d - 16*e + 25*f)*x^4 - 40*(13*d - 16*e + 25*f)*x^2 + 208*d - 256*e + 400*
f)*log(x + 1) - 32*((13*d + 16*e + 25*f)*x^8 - 10*(13*d + 16*e + 25*f)*x^6
+ 33*(13*d + 16*e + 25*f)*x^4 - 40*(13*d + 16*e + 25*f)*x^2 + 208*d + 256*e
 + 400*f)*log(x - 1) + ((313*d + 512*e + 820*f)*x^8 - 10*(313*d + 512*e + 8
20*f)*x^6 + 33*(313*d + 512*e + 820*f)*x^4 - 40*(313*d + 512*e + 820*f)*x^2
 + 5008*d + 8192*e + 13120*f)*log(x - 2) - 9600*e)/(x^8 - 10*x^6 + 33*x^4 -
 40*x^2 + 16)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4-5*x**2+4)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 155, normalized size of antiderivative = 0.89

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx$$
$$= -\frac{1}{41472}(313\,d-512\,e+820\,f)\log(x+2)+\frac{1}{1296}(13\,d-16\,e+25\,f)\log(x+1)$$
$$-\frac{1}{1296}(13\,d+16\,e+25\,f)\log(x-1)+\frac{1}{41472}(313\,d+512\,e+820\,f)\log(x-2)$$
$$+\frac{35\,(d+4\,f)x^7+128\,ex^6-18\,(13\,d+60\,f)x^5-960\,ex^4+63\,(5\,d+36\,f)x^3+1920\,ex^2+4\,(43\,d-260}{3456\,(x^8-10\,x^6+33\,x^4-40\,x^2+16)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="maxima")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f)*log(x + 2) + 1/1296*(13*d - 16*e + 25*f)*l
og(x + 1) - 1/1296*(13*d + 16*e + 25*f)*log(x - 1) + 1/41472*(313*d + 512*e
 + 820*f)*log(x - 2) + 1/3456*(35*(d + 4*f)*x^7 + 128*e*x^6 - 18*(13*d + 60
*f)*x^5 - 960*e*x^4 + 63*(5*d + 36*f)*x^3 + 1920*e*x^2 + 4*(43*d - 260*f)*x
 - 800*e)/(x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Giac [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 149, normalized size of antiderivative = 0.85

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx$$
$$= -\frac{1}{41472}(313\,d-512\,e+820\,f)\log(|x+2|)+\frac{1}{1296}(13\,d-16\,e+25\,f)\log(|x+1|)$$
$$-\frac{1}{1296}(13\,d+16\,e+25\,f)\log(|x-1|)+\frac{1}{41472}(313\,d+512\,e+820\,f)\log(|x-2|)$$
$$+\frac{35\,dx^7+140\,fx^7+128\,ex^6-234\,dx^5-1080\,fx^5-960\,ex^4+315\,dx^3+2268\,fx^3+1920\,ex^2+172\,dx}{3456\,(x^4-5\,x^2+4)^2}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="giac")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f)*log(abs(x + 2)) + 1/1296*(13*d - 16*e + 25
*f)*log(abs(x + 1)) - 1/1296*(13*d + 16*e + 25*f)*log(abs(x - 1)) + 1/41472
*(313*d + 512*e + 820*f)*log(abs(x - 2)) + 1/3456*(35*d*x^7 + 140*f*x^7 + 1
28*e*x^6 - 234*d*x^5 - 1080*f*x^5 - 960*e*x^4 + 315*d*x^3 + 2268*f*x^3 + 19
20*e*x^2 + 172*d*x - 1040*f*x - 800*e)/(x^4 - 5*x^2 + 4)^2
```

Mupad [B] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 151, normalized size of antiderivative = 0.86

$$\int \frac{d+ex+fx^2}{(4-5x^2+x^4)^3}\,dx = \ln(x+1)\left(\frac{13\,d}{1296}-\frac{e}{81}+\frac{25\,f}{1296}\right)-\ln(x-1)\left(\frac{13\,d}{1296}+\frac{e}{81}+\frac{25\,f}{1296}\right)+\ln(x-2)\left(\frac{313\,d}{41472}+\frac{e}{81}+\frac{205\,f}{10368}\right)-\ln(x+2)\left(\frac{313\,d}{41472}-\frac{e}{81}+\frac{205\,f}{10368}\right)+\frac{\left(\frac{35\,d}{3456}+\frac{35\,f}{864}\right)x^7+\frac{e\,x^6}{27}+\left(-\frac{13\,d}{192}-\frac{5\,f}{16}\right)x^5-\frac{5\,e\,x^4}{18}+\left(\frac{35\,d}{384}+\frac{21\,f}{32}\right)x^3+\frac{5\,e\,x^2}{9}+\left(\frac{43\,d}{864}-\frac{65\,f}{216}\right)x-\frac{25\,e}{108}}{x^8-10\,x^6+33\,x^4-40\,x^2+16}$$

```
[In] int((d + e*x + f*x^2)/(x^4 - 5*x^2 + 4)^3,x)
```

```
[Out] log(x + 1)*((13*d)/1296 - e/81 + (25*f)/1296) - log(x - 1)*((13*d)/1296 + e
/81 + (25*f)/1296) + log(x - 2)*((313*d)/41472 + e/81 + (205*f)/10368) - lo
g(x + 2)*((313*d)/41472 - e/81 + (205*f)/10368) + (x^3*((35*d)/384 + (21*f)
/32) - x^5*((13*d)/192 + (5*f)/16) - (25*e)/108 + x^7*((35*d)/3456 + (35*f)
/864) + (5*e*x^2)/9 - (5*e*x^4)/18 + (e*x^6)/27 + x*((43*d)/864 - (65*f)/21
6))/(33*x^4 - 40*x^2 - 10*x^6 + x^8 + 16)
```

3.44 $\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx$

Optimal result

Integrand size = 28, antiderivative size = 204

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx =& \frac{x(17d+20f-(5d+8f)x^2)}{144(4-5x^2+x^4)^2}+\frac{5e+8g-(2e+5g)x^2}{36(4-5x^2+x^4)^2}\\&-\frac{(2e+5g)(5-2x^2)}{108(4-5x^2+x^4)}-\frac{x(59d+380f-35(d+4f)x^2)}{3456(4-5x^2+x^4)}\\&-\frac{(313d+820f)\text{arctanh}\left(\frac{x}{2}\right)}{20736}+\frac{1}{648}(13d+25f)\text{arctanh}(x)\\&-\frac{1}{162}(2e+5g)\log\left(1-x^2\right)+\frac{1}{162}(2e+5g)\log\left(4-x^2\right)\end{aligned}$$

```
[Out] 1/144*x*(17*d+20*f-(5*d+8*f)*x^2)/(x^4-5*x^2+4)^2+1/36*(5*e+8*g-(2*e+5*g)*x
^2)/(x^4-5*x^2+4)^2-1/108*(2*e+5*g)*(-2*x^2+5)/(x^4-5*x^2+4)-1/3456*x*(59*d
+380*f-35*(d+4*f)*x^2)/(x^4-5*x^2+4)-1/20736*(313*d+820*f)*arctanh(1/2*x)+1
/648*(13*d+25*f)*arctanh(x)-1/162*(2*e+5*g)*ln(-x^2+1)+1/162*(2*e+5*g)*ln(-
x^2+4)
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 204, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.321, Rules used

$= \{1687, 1192, 1180, 213, 1261, 652, 628, 630, 31\}$

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx = &-\frac{\operatorname{arctanh}\left(\frac{x}{2}\right)(313d+820f)}{20736}+\frac{1}{648}\operatorname{arctanh}(x)(13d+25f)\\ &-\frac{x(-35x^2(d+4f)+59d+380f)}{3456\,(x^4-5x^2+4)}\\ &+\frac{x(-(x^2(5d+8f))+17d+20f)}{144\,(x^4-5x^2+4)^2}\\ &-\frac{1}{162}(2e+5g)\log\left(1-x^2\right)+\frac{1}{162}(2e+5g)\log\left(4-x^2\right)\\ &-\frac{(5-2x^2)(2e+5g)}{108\,(x^4-5x^2+4)}+\frac{-(x^2(2e+5g))+5e+8g}{36\,(x^4-5x^2+4)^2}\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (x*(17*d + 20*f - (5*d + 8*f)*x^2))/(144*(4 - 5*x^2 + x^4)^2) + (5*e + 8*g
- (2*e + 5*g)*x^2)/(36*(4 - 5*x^2 + x^4)^2) - ((2*e + 5*g)*(5 - 2*x^2))/(10
8*(4 - 5*x^2 + x^4)) - (x*(59*d + 380*f - 35*(d + 4*f)*x^2))/(3456*(4 - 5*x
^2 + x^4)) - ((313*d + 820*f)*ArcTanh[x/2])/20736 + ((13*d + 25*f)*ArcTanh[
x])/648 - ((2*e + 5*g)*Log[1 - x^2])/162 + ((2*e + 5*g)*Log[4 - x^2])/162
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d + fx^2}{(4 - 5x^2 + x^4)^3}\,dx + \int \frac{x(e + gx^2)}{(4 - 5x^2 + x^4)^3}\,dx$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} - \frac{1}{144}\int \frac{-19d+20f+5(5d+8f)x^2}{(4-5x^2+x^4)^2}\,dx + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(4-5x+x^2)^3}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\,(4-5x^2+x^4)} + \frac{\int \frac{3(173d+260f)+105(d+4f)x^2}{4-5x^2+x^4}\,dx}{10368} + \frac{1}{12}(-2e-5g)\text{Subst}\left(\int \frac{1}{(4-5x+x^2)^2}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} - \frac{(2e+5g)\,(5-2x^2)}{108\,(4-5x^2+x^4)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\,(4-5x^2+x^4)} + \frac{1}{648}(-13d-25f)\int \frac{1}{-1+x^2}\,dx + \frac{(313d+820f)\int \frac{1}{-4+x^2}\,dx}{10368} + \frac{1}{54}(2e+5g)\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} - \frac{(2e+5g)\,(5-2x^2)}{108\,(4-5x^2+x^4)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\,(4-5x^2+x^4)} - \frac{(313d+820f)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\tanh^{-1}(x) + \frac{1}{162}(-2e-5g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) + \frac{1}{162}(2e+5g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right)$$

$$= \frac{x(17d+20f-(5d+8f)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} - \frac{(2e+5g)\,(5-2x^2)}{108\,(4-5x^2+x^4)} - \frac{x(59d+380f-35(d+4f)x^2)}{3456\,(4-5x^2+x^4)} - \frac{(313d+820f)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f)\tanh^{-1}(x) - \frac{1}{162}(2e+5g)\log\left(1-x^2\right) + \frac{1}{162}(2e+5g)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 193, normalized size of antiderivative = 0.95

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx$$

$$= \frac{\frac{288\left(17dx+20fx-5dx^3-8fx^3+e\left(20-8x^2\right)-4g\left(-8+5x^2\right)\right)}{(4-5x^2+x^4)^2} + \frac{12\left(64e\left(-5+2x^2\right)+160g\left(-5+2x^2\right)+20fx\left(-19+7x^2\right)+dx\left(-59+35x^2\right)\right)}{4-5x^2+x^4} - 32(13}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] ((288*(17*d*x + 20*f*x - 5*d*x^3 - 8*f*x^3 + e*(20 - 8*x^2) - 4*g*(-8 + 5*x
^2)))/(4 - 5*x^2 + x^4)^2 + (12*(64*e*(-5 + 2*x^2) + 160*g*(-5 + 2*x^2) + 2
0*f*x*(-19 + 7*x^2) + d*x*(-59 + 35*x^2)))/(4 - 5*x^2 + x^4) - 32*(13*d + 1
6*e + 25*f + 40*g)*Log[1 - x] + (313*d + 512*e + 820*f + 1280*g)*Log[2 - x]
 + 32*(13*d - 16*e + 25*f - 40*g)*Log[1 + x] + (-313*d + 512*e - 820*f + 12
80*g)*Log[2 + x])/41472
```

Maple [A] (verified)

Time = 0.11 (sec) , antiderivative size = 169, normalized size of antiderivative = 0.83

method	result
norman	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}\right)x+\left(-\frac{5e}{18}-\frac{25g}{36}\right)x^4+\left(\frac{5e}{9}+\frac{25g}{18}\right)x^2+\left(\frac{e}{27}+\frac{5g}{54}\right)x^6-\frac{25e}{108}-\frac{19g}{27}}{\left(x^4-5x^2+4\right)^2}$ –
risch	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}\right)x+\left(-\frac{5e}{18}-\frac{25g}{36}\right)x^4+\left(\frac{5e}{9}+\frac{25g}{18}\right)x^2+\left(\frac{e}{27}+\frac{5g}{54}\right)x^6-\frac{25e}{108}-\frac{19g}{27}}{\left(x^4-5x^2+4\right)^2}$ –
default	$\left(-\frac{313d}{41472}+\frac{e}{81}-\frac{205f}{10368}+\frac{5g}{162}\right)\ln\left(x+2\right)-\frac{-\frac{19d}{6912}+\frac{17e}{3456}-\frac{5f}{576}+\frac{13g}{864}}{x+2}-\frac{-\frac{d}{1728}+\frac{e}{864}-\frac{f}{432}+\frac{g}{216}}{2\left(x+2\right)^2}-\frac{-\frac{d}{432}+\frac{e}{144}-}{x+1}$
parallelrisch	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x,method=_RETURNVERBOSE)
```

```
[Out] ((-13/192*d-5/16*f)*x^5+(35/384*d+21/32*f)*x^3+(35/3456*d+35/864*f)*x^7+(43
/864*d-65/216*f)*x+(-5/18*e-25/36*g)*x^4+(5/9*e+25/18*g)*x^2+(1/27*e+5/54*g
)*x^6-25/108*e-19/27*g)/(x^4-5*x^2+4)^2+(-313/41472*d+1/81*e-205/10368*f+5/
162*g)*ln(x+2)+(-13/1296*d-1/81*e-25/1296*f-5/162*g)*ln(x-1)+(13/1296*d-1/8
1*e+25/1296*f-5/162*g)*ln(x+1)+(313/41472*d+1/81*e+205/10368*f+5/162*g)*ln(
x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 470 vs. $2(184) = 368$.

Time = 0.50 (sec) , antiderivative size = 470, normalized size of antiderivative = 2.30

$$\int \frac{d+ex+fx^2+gx^3}{\left(4-5x^2+x^4\right)^3}\,dx = \frac{420\,(d+4\,f)x^7+768\,(2\,e+5\,g)x^6-216\,(13\,d+60\,f)x^5-5760\,(2\,e+5\,g)x^4+756\,(5\,d+36\,f)x^3+1152}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="fricas")
```

```
[Out] 1/41472*(420*(d + 4*f)*x^7 + 768*(2*e + 5*g)*x^6 - 216*(13*d + 60*f)*x^5 -
5760*(2*e + 5*g)*x^4 + 756*(5*d + 36*f)*x^3 + 11520*(2*e + 5*g)*x^2 + 48*(4
3*d - 260*f)*x - ((313*d - 512*e + 820*f - 1280*g)*x^8 - 10*(313*d - 512*e
```

```
+ 820*f - 1280*g)*x^6 + 33*(313*d - 512*e + 820*f - 1280*g)*x^4 - 40*(313*d
 - 512*e + 820*f - 1280*g)*x^2 + 5008*d - 8192*e + 13120*f - 20480*g)*log(x
 + 2) + 32*((13*d - 16*e + 25*f - 40*g)*x^8 - 10*(13*d - 16*e + 25*f - 40*g
)*x^6 + 33*(13*d - 16*e + 25*f - 40*g)*x^4 - 40*(13*d - 16*e + 25*f - 40*g)
*x^2 + 208*d - 256*e + 400*f - 640*g)*log(x + 1) - 32*((13*d + 16*e + 25*f
+ 40*g)*x^8 - 10*(13*d + 16*e + 25*f + 40*g)*x^6 + 33*(13*d + 16*e + 25*f +
 40*g)*x^4 - 40*(13*d + 16*e + 25*f + 40*g)*x^2 + 208*d + 256*e + 400*f + 6
40*g)*log(x - 1) + ((313*d + 512*e + 820*f + 1280*g)*x^8 - 10*(313*d + 512*
e + 820*f + 1280*g)*x^6 + 33*(313*d + 512*e + 820*f + 1280*g)*x^4 - 40*(313
*d + 512*e + 820*f + 1280*g)*x^2 + 5008*d + 8192*e + 13120*f + 20480*g)*log
(x - 2) - 9600*e - 29184*g)/(x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 188, normalized size of antiderivative = 0.92

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx = -\frac{1}{41472}\,(313\,d-512\,e+820\,f-1280\,g)\log{(x+2)} + \frac{1}{1296}\,(13\,d-16\,e+25\,f-40\,g)\log{(x+1)} - \frac{1}{1296}\,(13\,d+16\,e+25\,f+40\,g)\log{(x-1)} + \frac{1}{41472}\,(313\,d+512\,e+820\,f+1280\,g)\log{(x-2)} + \frac{35\,(d+4\,f)x^7+64\,(2\,e+5\,g)x^6-18\,(13\,d+60\,f)x^5-480\,(2\,e+5\,g)x^4+63\,(5\,d+36\,f)x^3+960\,(2\,e+}{3456\,(x^8-10\,x^6+33\,x^4-40\,x^2+16)}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="maxima")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g)*log(x + 2) + 1/1296*(13*d - 16*e
+ 25*f - 40*g)*log(x + 1) - 1/1296*(13*d + 16*e + 25*f + 40*g)*log(x - 1) +
 1/41472*(313*d + 512*e + 820*f + 1280*g)*log(x - 2) + 1/3456*(35*(d + 4*f)
*x^7 + 64*(2*e + 5*g)*x^6 - 18*(13*d + 60*f)*x^5 - 480*(2*e + 5*g)*x^4 + 63
*(5*d + 36*f)*x^3 + 960*(2*e + 5*g)*x^2 + 4*(43*d - 260*f)*x - 800*e - 2432
*g)/(x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 182, normalized size of antiderivative = 0.89

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx = -\frac{1}{41472}\,(313\,d-512\,e+820\,f-1280\,g)\log\left(|x+2|\right) + \frac{1}{1296}\,(13\,d-16\,e+25\,f-40\,g)\log\left(|x+1|\right) - \frac{1}{1296}\,(13\,d+16\,e+25\,f+40\,g)\log\left(|x-1|\right) + \frac{1}{41472}\,(313\,d+512\,e+820\,f+1280\,g)\log\left(|x-2|\right) + \frac{35\,dx^7+140\,fx^7+128\,ex^6+320\,gx^6-234\,dx^5-1080\,fx^5-960\,ex^4-2400\,gx^4+315\,dx^3+2268\,fx}{3456\,(x^4-5\,x^2+4)^2}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="giac")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g)*log(abs(x + 2)) + 1/1296*(13*d -
16*e + 25*f - 40*g)*log(abs(x + 1)) - 1/1296*(13*d + 16*e + 25*f + 40*g)*lo
g(abs(x - 1)) + 1/41472*(313*d + 512*e + 820*f + 1280*g)*log(abs(x - 2)) +
1/3456*(35*d*x^7 + 140*f*x^7 + 128*e*x^6 + 320*g*x^6 - 234*d*x^5 - 1080*f*x
^5 - 960*e*x^4 - 2400*g*x^4 + 315*d*x^3 + 2268*f*x^3 + 1920*e*x^2 + 4800*g*
x^2 + 172*d*x - 1040*f*x - 800*e - 2432*g)/(x^4 - 5*x^2 + 4)^2
```

Mupad [B] (verification not implemented)

Time = 7.86 (sec) , antiderivative size = 182, normalized size of antiderivative = 0.89

$$\int \frac{d+ex+fx^2+gx^3}{(4-5x^2+x^4)^3}\,dx = \frac{\left(\frac{35\,d}{3456}+\frac{35\,f}{864}\right)\,x^7+\left(\frac{e}{27}+\frac{5\,g}{54}\right)\,x^6+\left(-\frac{13\,d}{192}-\frac{5\,f}{16}\right)\,x^5+\left(-\frac{5\,e}{18}-\frac{25\,g}{36}\right)\,x^4+\left(\frac{35\,d}{384}+\frac{21\,f}{32}\right)\,x^3+\left(\frac{5\,e}{9}+\frac{25\,g}{18}\right)\,x^2}{x^8-10\,x^6+33\,x^4-40\,x^2+16} - \ln\left(x-1\right)\,\left(\frac{13\,d}{1296}+\frac{e}{81}+\frac{25\,f}{1296}+\frac{5\,g}{162}\right)+\ln\left(x+1\right)\,\left(\frac{13\,d}{1296}-\frac{e}{81}+\frac{25\,f}{1296}-\frac{5\,g}{162}\right) + \ln\left(x-2\right)\,\left(\frac{313\,d}{41472}+\frac{e}{81}+\frac{205\,f}{10368}+\frac{5\,g}{162}\right)-\ln\left(x+2\right)\,\left(\frac{313\,d}{41472}-\frac{e}{81}+\frac{205\,f}{10368}-\frac{5\,g}{162}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^4 - 5*x^2 + 4)^3,x)
```

```
[Out] (x^3*((35*d)/384 + (21*f)/32) - (19*g)/27 - x^5*((13*d)/192 + (5*f)/16) - (
25*e)/108 + x^7*((35*d)/3456 + (35*f)/864) + x^2*((5*e)/9 + (25*g)/18) - x^
4*((5*e)/18 + (25*g)/36) + x^6*(e/27 + (5*g)/54) + x*((43*d)/864 - (65*f)/2
16))/(33*x^4 - 40*x^2 - 10*x^6 + x^8 + 16) - log(x - 1)*((13*d)/1296 + e/81
```

```
 + (25*f)/1296 + (5*g)/162) + log(x + 1)*((13*d)/1296 - e/81 + (25*f)/1296
- (5*g)/162) + log(x - 2)*((313*d)/41472 + e/81 + (205*f)/10368 + (5*g)/162
) - log(x + 2)*((313*d)/41472 - e/81 + (205*f)/10368 - (5*g)/162)
```

3.45 $\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3} dx$

Optimal result

Integrand size = 33, antiderivative size = 224

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3} dx &= \frac{5e+8g-(2e+5g)x^2}{36(4-5x^2+x^4)^2} \\ &+ \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144(4-5x^2+x^4)^2} \\ &- \frac{(2e+5g)(5-2x^2)}{108(4-5x^2+x^4)} \\ &- \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456(4-5x^2+x^4)} \\ &- \frac{(313d+820f+1936h)\text{arctanh}\left(\frac{x}{2}\right)}{20736} \\ &+ \frac{1}{648}(13d+25f+61h)\text{arctanh}(x) \\ &- \frac{1}{162}(2e+5g)\log\left(1-x^2\right) + \frac{1}{162}(2e+5g)\log\left(4-x^2\right)\end{aligned}$$

```
[Out] 1/36*(5*e+8*g-(2*e+5*g)*x^2)/(x^4-5*x^2+4)^2+1/144*x*(17*d+20*f+32*h-(5*d+8
*f+20*h)*x^2)/(x^4-5*x^2+4)^2-1/108*(2*e+5*g)*(-2*x^2+5)/(x^4-5*x^2+4)-1/34
56*x*(59*d+380*f+848*h-5*(7*d+28*f+64*h)*x^2)/(x^4-5*x^2+4)-1/20736*(313*d+
820*f+1936*h)*arctanh(1/2*x)+1/648*(13*d+25*f+61*h)*arctanh(x)-1/162*(2*e+5
*g)*ln(-x^2+1)+1/162*(2*e+5*g)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 224, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.303, Rules used = {1687, 1692, 1192, 1180, 213, 1261, 652, 628, 630, 31}

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx = & -\frac{\operatorname{arctanh}\left(\frac{x}{2}\right)(313d+820f+1936h)}{20736} \\ & +\frac{1}{648}\operatorname{arctanh}(x)(13d+25f+61h) \\ & -\frac{x(-5x^2(7d+28f+64h)+59d+380f+848h)}{3456\,(x^4-5x^2+4)} \\ & +\frac{x(-(x^2(5d+8f+20h))+17d+20f+32h)}{144\,(x^4-5x^2+4)^2} \\ & -\frac{1}{162}(2e+5g)\log\left(1-x^2\right)+\frac{1}{162}(2e+5g)\log\left(4-x^2\right) \\ & -\frac{(5-2x^2)\,(2e+5g)}{108\,(x^4-5x^2+4)}+\frac{-(x^2(2e+5g))+5e+8g}{36\,(x^4-5x^2+4)^2}\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (5*e + 8*g - (2*e + 5*g)*x^2)/(36*(4 - 5*x^2 + x^4)^2) + (x*(17*d + 20*f +
32*h - (5*d + 8*f + 20*h)*x^2))/(144*(4 - 5*x^2 + x^4)^2) - ((2*e + 5*g)*(5
 - 2*x^2))/(108*(4 - 5*x^2 + x^4)) - (x*(59*d + 380*f + 848*h - 5*(7*d + 28
*f + 64*h)*x^2))/(3456*(4 - 5*x^2 + x^4)) - ((313*d + 820*f + 1936*h)*ArcTa
nh[x/2])/20736 + ((13*d + 25*f + 61*h)*ArcTanh[x])/648 - ((2*e + 5*g)*Log[1
 - x^2])/162 + ((2*e + 5*g)*Log[4 - x^2])/162
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
 (LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{(4-5x^2+x^4)^3}\,dx + \int \frac{d+fx^2+hx^4}{(4-5x^2+x^4)^3}\,dx \\
&= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} \\
&\quad - \frac{1}{144}\int \frac{-19d+20f+32h+5(5d+8f+20h)x^2}{(4-5x^2+x^4)^2}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(4-5x+x^2)^3}\,dx, x, x^2\right) \\
&= \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} + \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} \\
&\quad - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\,(4-5x^2+x^4)} \\
&\quad + \frac{\int \frac{3(173d+260f+656h)+15(7d+28f+64h)x^2}{4-5x^2+x^4}\,dx}{10368} \\
&\quad + \frac{1}{12}(-2e-5g)\text{Subst}\left(\int \frac{1}{(4-5x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{5e+8g-(2e+5g)x^2}{36\,(4-5x^2+x^4)^2} + \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} \\
&\quad - \frac{(2e+5g)\,(5-2x^2)}{108\,(4-5x^2+x^4)} - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\,(4-5x^2+x^4)} \\
&\quad + \frac{1}{54}(2e+5g)\text{Subst}\left(\int \frac{1}{4-5x+x^2}\,dx, x, x^2\right) \\
&\quad + \frac{1}{648}(-13d-25f-61h)\int \frac{1}{-1+x^2}\,dx + \frac{(313d+820f+1936h)\int \frac{1}{-4+x^2}\,dx}{10368}
\end{aligned}
$$

$$= \frac{5e+8g-(2e+5g)x^2}{36(4-5x^2+x^4)^2} + \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144(4-5x^2+x^4)^2} - \frac{(2e+5g)(5-2x^2)}{108(4-5x^2+x^4)} - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456(4-5x^2+x^4)} - \frac{(313d+820f+1936h)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f+61h)\tanh^{-1}(x) + \frac{1}{162}(-2e-5g)\text{Subst}\left(\int \frac{1}{-1+x}\,dx, x, x^2\right) + \frac{1}{162}(2e+5g)\text{Subst}\left(\int \frac{1}{-4+x}\,dx, x, x^2\right)$$

$$= \frac{5e+8g-(2e+5g)x^2}{36(4-5x^2+x^4)^2} + \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144(4-5x^2+x^4)^2} - \frac{(2e+5g)(5-2x^2)}{108(4-5x^2+x^4)} - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456(4-5x^2+x^4)} - \frac{(313d+820f+1936h)\tanh^{-1}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f+61h)\tanh^{-1}(x) - \frac{1}{162}(2e+5g)\log\left(1-x^2\right) + \frac{1}{162}(2e+5g)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.09 (sec) , antiderivative size = 231, normalized size of antiderivative = 1.03

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx = \frac{20e+32g+17dx+20fx+32hx-8ex^2-20gx^2-5dx^3-8fx^3-20hx^3}{144(4-5x^2+x^4)^2} + \frac{-320e-800g-59dx-380fx-848hx+128ex^2+320gx^2+35dx^3+140fx^3+320hx^3}{3456(4-5x^2+x^4)} + \frac{(-13d-16e-25f-40g-61h)\log(1-x)}{1296} + \frac{(313d+512e+820f+1280g+1936h)\log(2-x)}{41472} + \frac{(13d-16e+25f-40g+61h)\log(1+x)}{1296} + \frac{(-313d+512e-820f+1280g-1936h)\log(2+x)}{41472}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (20*e + 32*g + 17*d*x + 20*f*x + 32*h*x - 8*e*x^2 - 20*g*x^2 - 5*d*x^3 - 8*
f*x^3 - 20*h*x^3)/(144*(4 - 5*x^2 + x^4)^2) + (-320*e - 800*g - 59*d*x - 38
0*f*x - 848*h*x + 128*e*x^2 + 320*g*x^2 + 35*d*x^3 + 140*f*x^3 + 320*h*x^3)
/(3456*(4 - 5*x^2 + x^4)) + ((-13*d - 16*e - 25*f - 40*g - 61*h)*Log[1 - x]
```

```
)/1296 + ((313*d + 512*e + 820*f + 1280*g + 1936*h)*Log[2 - x])/41472 + ((1
3*d - 16*e + 25*f - 40*g + 61*h)*Log[1 + x])/1296 + ((-313*d + 512*e - 820*
f + 1280*g - 1936*h)*Log[2 + x])/41472
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 193, normalized size of antiderivative = 0.86

method	result
norman	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}-\frac{17h}{24}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}+\frac{35h}{24}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}+\frac{5h}{54}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}-\frac{41h}{54}\right)x+\left(-\frac{5e}{18}-\frac{25g}{36}\right)x^4+\left(\frac{5e}{9}+\frac{25g}{18}\right)x^2+\left(\frac{e}{27}+\right.}{\left(x^4-5x^2+4\right)^2}$
risch	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}-\frac{17h}{24}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}+\frac{35h}{24}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}+\frac{5h}{54}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}-\frac{41h}{54}\right)x+\left(-\frac{5e}{18}-\frac{25g}{36}\right)x^4+\left(\frac{5e}{9}+\frac{25g}{18}\right)x^2+\left(\frac{e}{27}+\right.}{\left(x^4-5x^2+4\right)^2}$
default	$\left(-\frac{313d}{41472}+\frac{e}{81}-\frac{205f}{10368}+\frac{5g}{162}-\frac{121h}{2592}\right)\ln\left(x+2\right)-\frac{-\frac{19d}{6912}+\frac{17e}{3456}-\frac{5f}{576}+\frac{13g}{864}-\frac{11h}{432}}{x+2}-\frac{-\frac{d}{1728}+\frac{e}{864}-\frac{f}{432}+\frac{g}{216}-\frac{h}{10}}{2(x+2)^2}$
parallelrisch	Expression too large to display

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x,method=_RETURNVERBOSE)
```

```
[Out] ((-13/192*d-5/16*f-17/24*h)*x^5+(35/384*d+21/32*f+35/24*h)*x^3+(35/3456*d+3
5/864*f+5/54*h)*x^7+(43/864*d-65/216*f-41/54*h)*x+(-5/18*e-25/36*g)*x^4+(5/
9*e+25/18*g)*x^2+(1/27*e+5/54*g)*x^6-25/108*e-19/27*g)/(x^4-5*x^2+4)^2+(-31
3/41472*d+1/81*e-205/10368*f+5/162*g-121/2592*h)*ln(x+2)+(-13/1296*d-1/81*e
-25/1296*f-5/162*g-61/1296*h)*ln(x-1)+(13/1296*d-1/81*e+25/1296*f-5/162*g+6
1/1296*h)*ln(x+1)+(313/41472*d+1/81*e+205/10368*f+5/162*g+121/2592*h)*ln(x-
2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 544 vs. 2(204) = 408.

Time = 1.45 (sec) , antiderivative size = 544, normalized size of antiderivative = 2.43

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{\left(4-5x^2+x^4\right)^3}\,dx$$
$$=\frac{60\,(7\,d+28\,f+64\,h)x^7+768\,(2\,e+5\,g)x^6-216\,(13\,d+60\,f+136\,h)x^5-5760\,(2\,e+5\,g)x^4+756\,(5\,d+}{}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="fricas")
```

```
[Out] 1/41472*(60*(7*d + 28*f + 64*h)*x^7 + 768*(2*e + 5*g)*x^6 - 216*(13*d + 60*
f + 136*h)*x^5 - 5760*(2*e + 5*g)*x^4 + 756*(5*d + 36*f + 80*h)*x^3 + 11520
*(2*e + 5*g)*x^2 + 48*(43*d - 260*f - 656*h)*x - ((313*d - 512*e + 820*f -
1280*g + 1936*h)*x^8 - 10*(313*d - 512*e + 820*f - 1280*g + 1936*h)*x^6 + 3
3*(313*d - 512*e + 820*f - 1280*g + 1936*h)*x^4 - 40*(313*d - 512*e + 820*f
```

```
 - 1280*g + 1936*h)*x^2 + 5008*d - 8192*e + 13120*f - 20480*g + 30976*h)*lo
g(x + 2) + 32*((13*d - 16*e + 25*f - 40*g + 61*h)*x^8 - 10*(13*d - 16*e + 2
5*f - 40*g + 61*h)*x^6 + 33*(13*d - 16*e + 25*f - 40*g + 61*h)*x^4 - 40*(13
*d - 16*e + 25*f - 40*g + 61*h)*x^2 + 208*d - 256*e + 400*f - 640*g + 976*h
)*log(x + 1) - 32*((13*d + 16*e + 25*f + 40*g + 61*h)*x^8 - 10*(13*d + 16*e
 + 25*f + 40*g + 61*h)*x^6 + 33*(13*d + 16*e + 25*f + 40*g + 61*h)*x^4 - 40
*(13*d + 16*e + 25*f + 40*g + 61*h)*x^2 + 208*d + 256*e + 400*f + 640*g + 9
76*h)*log(x - 1) + ((313*d + 512*e + 820*f + 1280*g + 1936*h)*x^8 - 10*(313
*d + 512*e + 820*f + 1280*g + 1936*h)*x^6 + 33*(313*d + 512*e + 820*f + 128
0*g + 1936*h)*x^4 - 40*(313*d + 512*e + 820*f + 1280*g + 1936*h)*x^2 + 5008
*d + 8192*e + 13120*f + 20480*g + 30976*h)*log(x - 2) - 9600*e - 29184*g)/(
x^8 - 10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 214, normalized size of antiderivative = 0.96

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx \\
&= -\frac{1}{41472}\,(313\,d-512\,e+820\,f-1280\,g+1936\,h)\log(x+2) \\
&\quad+\frac{1}{1296}\,(13\,d-16\,e+25\,f-40\,g+61\,h)\log(x+1) \\
&\quad-\frac{1}{1296}\,(13\,d+16\,e+25\,f+40\,g+61\,h)\log(x-1) \\
&\quad+\frac{1}{41472}\,(313\,d+512\,e+820\,f+1280\,g+1936\,h)\log(x-2) \\
&\quad+\frac{5\,(7\,d+28\,f+64\,h)x^7+64\,(2\,e+5\,g)x^6-18\,(13\,d+60\,f+136\,h)x^5-480\,(2\,e+5\,g)x^4+63\,(5\,d+3}{3456\,(x^8-10\,x^6+33\,x^4-40\,x}
\end{aligned}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="maxima")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g + 1936*h)*log(x + 2) + 1/1296*(13*
d - 16*e + 25*f - 40*g + 61*h)*log(x + 1) - 1/1296*(13*d + 16*e + 25*f + 40
```

```
*g + 61*h)*log(x - 1) + 1/41472*(313*d + 512*e + 820*f + 1280*g + 1936*h)*l
og(x - 2) + 1/3456*(5*(7*d + 28*f + 64*h)*x^7 + 64*(2*e + 5*g)*x^6 - 18*(13
*d + 60*f + 136*h)*x^5 - 480*(2*e + 5*g)*x^4 + 63*(5*d + 36*f + 80*h)*x^3 +
 960*(2*e + 5*g)*x^2 + 4*(43*d - 260*f - 656*h)*x - 800*e - 2432*g)/(x^8 -
10*x^6 + 33*x^4 - 40*x^2 + 16)
```

Giac [A] (verification not implemented)

none

Time = 0.36 (sec) , antiderivative size = 216, normalized size of antiderivative = 0.96

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx$$
$$= -\frac{1}{41472}(313\,d-512\,e+820\,f-1280\,g+1936\,h)\log(|x+2|) + \frac{1}{1296}(13\,d-16\,e+25\,f-40\,g+61\,h)\log(|x+1|) - \frac{1}{1296}(13\,d+16\,e+25\,f+40\,g+61\,h)\log(|x-1|) + \frac{1}{41472}(313\,d+512\,e+820\,f+1280\,g+1936\,h)\log(|x-2|) + \frac{35\,dx^7+140\,fx^7+320\,hx^7+128\,ex^6+320\,gx^6-234\,dx^5-1080\,fx^5-2448\,hx^5-960\,ex^4-2400\,gx^4}{3456\,(x^4-}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="giac")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g + 1936*h)*log(abs(x + 2)) + 1/1296
*(13*d - 16*e + 25*f - 40*g + 61*h)*log(abs(x + 1)) - 1/1296*(13*d + 16*e +
 25*f + 40*g + 61*h)*log(abs(x - 1)) + 1/41472*(313*d + 512*e + 820*f + 128
0*g + 1936*h)*log(abs(x - 2)) + 1/3456*(35*d*x^7 + 140*f*x^7 + 320*h*x^7 +
128*e*x^6 + 320*g*x^6 - 234*d*x^5 - 1080*f*x^5 - 2448*h*x^5 - 960*e*x^4 - 2
400*g*x^4 + 315*d*x^3 + 2268*f*x^3 + 5040*h*x^3 + 1920*e*x^2 + 4800*g*x^2 +
 172*d*x - 1040*f*x - 2624*h*x - 800*e - 2432*g)/(x^4 - 5*x^2 + 4)^2
```

Mupad [B] (verification not implemented)

Time = 0.15 (sec) , antiderivative size = 209, normalized size of antiderivative = 0.93

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(4-5x^2+x^4)^3}\,dx = \ln(x+1)\left(\frac{13\,d}{1296}-\frac{e}{81}+\frac{25\,f}{1296}-\frac{5\,g}{162}+\frac{61\,h}{1296}\right)$$
$$-\ln(x-1)\left(\frac{13\,d}{1296}+\frac{e}{81}+\frac{25\,f}{1296}+\frac{5\,g}{162}+\frac{61\,h}{1296}\right)$$
$$-\frac{\left(-\frac{35\,d}{3456}-\frac{35\,f}{864}-\frac{5\,h}{54}\right)x^7+\left(-\frac{e}{27}-\frac{5\,g}{54}\right)x^6+\left(\frac{13\,d}{192}+\frac{5\,f}{16}+\frac{17\,h}{24}\right)x^5+\left(\frac{5\,e}{18}+\frac{25\,g}{36}\right)x^4+\left(-\frac{35\,d}{384}-\frac{21\,f}{32}-\frac{35\,h}{24}\right.}{x^8-10\,x^6+33\,x^4-40\,x^2+16}$$
$$+\ln(x-2)\left(\frac{313\,d}{41472}+\frac{e}{81}+\frac{205\,f}{10368}+\frac{5\,g}{162}+\frac{121\,h}{2592}\right)$$
$$-\ln(x+2)\left(\frac{313\,d}{41472}-\frac{e}{81}+\frac{205\,f}{10368}-\frac{5\,g}{162}+\frac{121\,h}{2592}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^4 - 5*x^2 + 4)^3,x)
```

```
[Out] log(x + 1)*((13*d)/1296 - e/81 + (25*f)/1296 - (5*g)/162 + (61*h)/1296) - l
og(x - 1)*((13*d)/1296 + e/81 + (25*f)/1296 + (5*g)/162 + (61*h)/1296) - ((
25*e)/108 + (19*g)/27 - x^2*((5*e)/9 + (25*g)/18) + x^4*((5*e)/18 + (25*g)/
36) - x^6*(e/27 + (5*g)/54) + x*((65*f)/216 - (43*d)/864 + (41*h)/54) + x^5
*((13*d)/192 + (5*f)/16 + (17*h)/24) - x^3*((35*d)/384 + (21*f)/32 + (35*h)
/24) - x^7*((35*d)/3456 + (35*f)/864 + (5*h)/54))/(33*x^4 - 40*x^2 - 10*x^6
 + x^8 + 16) + log(x - 2)*((313*d)/41472 + e/81 + (205*f)/10368 + (5*g)/162
 + (121*h)/2592) - log(x + 2)*((313*d)/41472 - e/81 + (205*f)/10368 - (5*g)
/162 + (121*h)/2592)
```

3.46 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx$

Optimal result

Integrand size = 38, antiderivative size = 239

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx = \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{36\,(4-5x^2+x^4)^2} - \frac{(2e+5g+11i)\,(5-2x^2)}{108\,(4-5x^2+x^4)} - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\,(4-5x^2+x^4)} - \frac{(313d+820f+1936h)\text{arctanh}\left(\frac{x}{2}\right)}{20736} + \frac{1}{648}(13d+25f+61h)\text{arctanh}(x) - \frac{1}{162}(2e+5g+11i)\log\left(1-x^2\right) + \frac{1}{162}(2e+5g+11i)\log\left(4-x^2\right)$$

```
[Out] 1/144*x*(17*d+20*f+32*h-(5*d+8*f+20*h)*x^2)/(x^4-5*x^2+4)^2+1/36*(5*e+8*g+2
0*i-(2*e+5*g+17*i)*x^2)/(x^4-5*x^2+4)^2-1/108*(2*e+5*g+11*i)*(-2*x^2+5)/(x^
4-5*x^2+4)-1/3456*x*(59*d+380*f+848*h-5*(7*d+28*f+64*h)*x^2)/(x^4-5*x^2+4)-
1/20736*(313*d+820*f+1936*h)*arctanh(1/2*x)+1/648*(13*d+25*f+61*h)*arctanh(
x)-1/162*(2*e+5*g+11*i)*ln(-x^2+1)+1/162*(2*e+5*g+11*i)*ln(-x^2+4)
```

Rubi [A] (verified)

Time = 0.22 (sec) , antiderivative size = 239, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 11, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.289, Rules used = {1687, 1692, 1192, 1180, 213, 1677, 1674, 12, 628, 630, 31}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx = -\frac{\operatorname{arctanh}\left(\frac{x}{2}\right)(313d+820f+1936h)}{20736} + \frac{1}{648}\operatorname{arctanh}(x)(13d+25f+61h) - \frac{x(-5x^2(7d+28f+64h)+59d+380f+848h)}{3456\,(x^4-5x^2+4)} + \frac{x(-(x^2(5d+8f+20h))+17d+20f+32h)}{144\,(x^4-5x^2+4)^2} - \frac{1}{162}\log\left(1-x^2\right)(2e+5g+11i) + \frac{1}{162}\log\left(4-x^2\right)(2e+5g+11i) - \frac{(5-2x^2)(2e+5g+11i)}{108\,(x^4-5x^2+4)} + \frac{-(x^2(2e+5g+17i))+5e+8g+20i}{36\,(x^4-5x^2+4)^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (x*(17*d + 20*f + 32*h - (5*d + 8*f + 20*h)*x^2))/(144*(4 - 5*x^2 + x^4)^2)
 + (5*e + 8*g + 20*i - (2*e + 5*g + 17*i)*x^2)/(36*(4 - 5*x^2 + x^4)^2) - (
(2*e + 5*g + 11*i)*(5 - 2*x^2))/(108*(4 - 5*x^2 + x^4)) - (x*(59*d + 380*f
+ 848*h - 5*(7*d + 28*f + 64*h)*x^2))/(3456*(4 - 5*x^2 + x^4)) - ((313*d +
820*f + 1936*h)*ArcTanh[x/2])/20736 + ((13*d + 25*f + 61*h)*ArcTanh[x])/648
 - ((2*e + 5*g + 11*i)*Log[1 - x^2])/162 + ((2*e + 5*g + 11*i)*Log[4 - x^2]
)/162
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 213

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[b, 2])^(-
1))*ArcTanh[Rt[b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] &&
```

```
(LtQ[a, 0] || GtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+fx^2+hx^4}{(4-5x^2+x^4)^3}\,dx + \int \frac{x(e+gx^2+ix^4)}{(4-5x^2+x^4)^3}\,dx \\
&= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} \\
&\quad - \frac{1}{144}\int \frac{-19d+20f+32h+5(5d+8f+20h)x^2}{(4-5x^2+x^4)^2}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{(4-5x+x^2)^3}\,dx, x, x^2\right) \\
&= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\,(4-5x^2+x^4)^2} + \frac{5e+8g+20i-(2e+5g+17i)x^2}{36\,(4-5x^2+x^4)^2} \\
&\quad - \frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\,(4-5x^2+x^4)} \\
&\quad + \frac{\int \frac{3(173d+260f+656h)+15(7d+28f+64h)x^2}{4-5x^2+x^4}\,dx}{10368} - \frac{1}{36}\text{Subst}\left(\int \frac{3(2e+5g+11i)}{(4-5x+x^2)^2}\,dx, x, x^2\right)
\end{aligned}$$

$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{5e+8g+20i-(2e+5g+17i)x^2}{36\left(4-5x^2+x^4\right)^2}$$
$$-\frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\left(4-5x^2+x^4\right)}$$
$$+\frac{1}{648}(-13d-25f-61h)\int\frac{1}{-1+x^2}\,dx+\frac{(313d+820f+1936h)\int\frac{1}{-4+x^2}\,dx}{10368}$$
$$-\frac{1}{12}(2e+5g+11i)\text{Subst}\left(\int\frac{1}{(4-5x+x^2)^2}\,dx,x,x^2\right)$$
$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{5e+8g+20i-(2e+5g+17i)x^2}{36\left(4-5x^2+x^4\right)^2}$$
$$-\frac{(2e+5g+11i)\left(5-2x^2\right)}{108\left(4-5x^2+x^4\right)}-\frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\left(4-5x^2+x^4\right)}$$
$$-\frac{(313d+820f+1936h)\tanh^{-1}\left(\frac{x}{2}\right)}{20736}+\frac{1}{648}(13d+25f+61h)\tanh^{-1}(x)$$
$$-\frac{1}{54}(-2e-5g-11i)\text{Subst}\left(\int\frac{1}{4-5x+x^2}\,dx,x,x^2\right)$$
$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{5e+8g+20i-(2e+5g+17i)x^2}{36\left(4-5x^2+x^4\right)^2}$$
$$-\frac{(2e+5g+11i)\left(5-2x^2\right)}{108\left(4-5x^2+x^4\right)}-\frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\left(4-5x^2+x^4\right)}$$
$$-\frac{(313d+820f+1936h)\tanh^{-1}\left(\frac{x}{2}\right)}{20736}+\frac{1}{648}(13d+25f+61h)\tanh^{-1}(x)$$
$$-\frac{1}{162}(-2e-5g-11i)\text{Subst}\left(\int\frac{1}{-4+x}\,dx,x,x^2\right)$$
$$-\frac{1}{162}(2e+5g+11i)\text{Subst}\left(\int\frac{1}{-1+x}\,dx,x,x^2\right)$$
$$= \frac{x(17d+20f+32h-(5d+8f+20h)x^2)}{144\left(4-5x^2+x^4\right)^2}+\frac{5e+8g+20i-(2e+5g+17i)x^2}{36\left(4-5x^2+x^4\right)^2}$$
$$-\frac{(2e+5g+11i)\left(5-2x^2\right)}{108\left(4-5x^2+x^4\right)}-\frac{x(59d+380f+848h-5(7d+28f+64h)x^2)}{3456\left(4-5x^2+x^4\right)}$$
$$-\frac{(313d+820f+1936h)\tanh^{-1}\left(\frac{x}{2}\right)}{20736}+\frac{1}{648}(13d+25f+61h)\tanh^{-1}(x)$$
$$-\frac{1}{162}(2e+5g+11i)\log\left(1-x^2\right)+\frac{1}{162}(2e+5g+11i)\log\left(4-x^2\right)$$

Mathematica [A] (verified)

Time = 0.09 (sec) , antiderivative size = 261, normalized size of antiderivative = 1.09

$$
\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx \\
&= \frac{20e+32g+80i+17dx+20fx+32hx-8ex^2-20gx^2-68ix^2-5dx^3-8fx^3-20hx^3}{144\,(4-5x^2+x^4)^2} \\
&\quad + \frac{-320e-800g-1760i-59dx-380fx-848hx+128ex^2+320gx^2+704ix^2+35dx^3+140fx^3+32}{3456\,(4-5x^2+x^4)} \\
&\quad + \frac{(-13d-16e-25f-40g-61h-88i)\log(1-x)}{1296} \\
&\quad + \frac{(313d+512e+820f+1280g+1936h+2816i)\log(2-x)}{41472} \\
&\quad + \frac{(13d-16e+25f-40g+61h-88i)\log(1+x)}{1296} \\
&\quad + \frac{(-313d+512e-820f+1280g-1936h+2816i)\log(2+x)}{41472}
\end{aligned}
$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(4 - 5*x^2 + x^4)^3,x]
```

```
[Out] (20*e + 32*g + 80*i + 17*d*x + 20*f*x + 32*h*x - 8*e*x^2 - 20*g*x^2 - 68*i*
x^2 - 5*d*x^3 - 8*f*x^3 - 20*h*x^3)/(144*(4 - 5*x^2 + x^4)^2) + (-320*e - 8
00*g - 1760*i - 59*d*x - 380*f*x - 848*h*x + 128*e*x^2 + 320*g*x^2 + 704*i*
x^2 + 35*d*x^3 + 140*f*x^3 + 320*h*x^3)/(3456*(4 - 5*x^2 + x^4)) + ((-13*d
- 16*e - 25*f - 40*g - 61*h - 88*i)*Log[1 - x])/1296 + ((313*d + 512*e + 82
0*f + 1280*g + 1936*h + 2816*i)*Log[2 - x])/41472 + ((13*d - 16*e + 25*f -
40*g + 61*h - 88*i)*Log[1 + x])/1296 + ((-313*d + 512*e - 820*f + 1280*g -
1936*h + 2816*i)*Log[2 + x])/41472
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 217, normalized size of antiderivative = 0.91

method	result
norman	$\frac{\left(-\frac{13d}{192}-\frac{5f}{16}-\frac{17h}{24}\right)x^5+\left(\frac{35d}{384}+\frac{21f}{32}+\frac{35h}{24}\right)x^3+\left(\frac{35d}{3456}+\frac{35f}{864}+\frac{5h}{54}\right)x^7+\left(\frac{43d}{864}-\frac{65f}{216}-\frac{41h}{54}\right)x+\left(-\frac{5e}{18}-\frac{25g}{36}-\frac{55i}{36}\right)x^4+\left(\frac{5e}{9}+\frac{25g}{18}+\frac{26}{9}\right.}{(x^4-5x^2+4)^2}$
default	$\left(-\frac{313d}{41472}+\frac{e}{81}-\frac{205f}{10368}+\frac{5g}{162}-\frac{121h}{2592}+\frac{11i}{162}\right)\ln(x+2)-\frac{-\frac{19d}{6912}+\frac{17e}{3456}-\frac{5f}{576}+\frac{13g}{864}-\frac{11h}{432}+\frac{i}{24}}{x+2}-\frac{-\frac{d}{1728}+\frac{e}{864}}{}$
risch	$-\frac{11\ln(x+1)i}{162}-\frac{205\ln(x+2)f}{10368}-\frac{25\ln(1-x)f}{1296}+\frac{25\ln(x+1)f}{1296}+\frac{205\ln(2-x)f}{10368}+\frac{313\ln(2-x)d}{41472}+\frac{\ln(2-x)e}{81}-\frac{313\,\mathrm{l}}{4}$
parallelrisch	Expression too large to display

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x,method=_RETURNVERBOSE
)
```

```
[Out] ((-13/192*d-5/16*f-17/24*h)*x^5+(35/384*d+21/32*f+35/24*h)*x^3+(35/3456*d+3
5/864*f+5/54*h)*x^7+(43/864*d-65/216*f-41/54*h)*x+(-5/18*e-25/36*g-55/36*i)
*x^4+(5/9*e+25/18*g+26/9*i)*x^2+(1/27*e+5/54*g+11/54*i)*x^6-25/108*e-19/27*
g-40/27*i)/(x^4-5*x^2+4)^2+(-313/41472*d+1/81*e-205/10368*f+5/162*g-121/259
2*h+11/162*i)*ln(x+2)+(-13/1296*d-1/81*e-25/1296*f-5/162*g-61/1296*h-11/162
*i)*ln(x-1)+(13/1296*d-1/81*e+25/1296*f-5/162*g+61/1296*h-11/162*i)*ln(x+1)
+(313/41472*d+1/81*e+205/10368*f+5/162*g+121/2592*h+11/162*i)*ln(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 616 vs. $2(219) = 438$.

Time = 6.60 (sec) , antiderivative size = 616, normalized size of antiderivative = 2.58

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx = \frac{60\,(7\,d+28\,f+64\,h)x^7+768\,(2\,e+5\,g+11\,i)x^6-216\,(13\,d+60\,f+136\,h)x^5-5760\,(2\,e+5\,g+11\,i)x^4}{}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="fri
cas")
```

```
[Out] 1/41472*(60*(7*d + 28*f + 64*h)*x^7 + 768*(2*e + 5*g + 11*i)*x^6 - 216*(13*
d + 60*f + 136*h)*x^5 - 5760*(2*e + 5*g + 11*i)*x^4 + 756*(5*d + 36*f + 80*
h)*x^3 + 2304*(10*e + 25*g + 52*i)*x^2 + 48*(43*d - 260*f - 656*h)*x - ((31
3*d - 512*e + 820*f - 1280*g + 1936*h - 2816*i)*x^8 - 10*(313*d - 512*e + 8
20*f - 1280*g + 1936*h - 2816*i)*x^6 + 33*(313*d - 512*e + 820*f - 1280*g +
 1936*h - 2816*i)*x^4 - 40*(313*d - 512*e + 820*f - 1280*g + 1936*h - 2816*
i)*x^2 + 5008*d - 8192*e + 13120*f - 20480*g + 30976*h - 45056*i)*log(x + 2
) + 32*((13*d - 16*e + 25*f - 40*g + 61*h - 88*i)*x^8 - 10*(13*d - 16*e + 2
5*f - 40*g + 61*h - 88*i)*x^6 + 33*(13*d - 16*e + 25*f - 40*g + 61*h - 88*i
)*x^4 - 40*(13*d - 16*e + 25*f - 40*g + 61*h - 88*i)*x^2 + 208*d - 256*e +
400*f - 640*g + 976*h - 1408*i)*log(x + 1) - 32*((13*d + 16*e + 25*f + 40*g
 + 61*h + 88*i)*x^8 - 10*(13*d + 16*e + 25*f + 40*g + 61*h + 88*i)*x^6 + 33
*(13*d + 16*e + 25*f + 40*g + 61*h + 88*i)*x^4 - 40*(13*d + 16*e + 25*f + 4
0*g + 61*h + 88*i)*x^2 + 208*d + 256*e + 400*f + 640*g + 976*h + 1408*i)*lo
g(x - 1) + ((313*d + 512*e + 820*f + 1280*g + 1936*h + 2816*i)*x^8 - 10*(31
3*d + 512*e + 820*f + 1280*g + 1936*h + 2816*i)*x^6 + 33*(313*d + 512*e + 8
20*f + 1280*g + 1936*h + 2816*i)*x^4 - 40*(313*d + 512*e + 820*f + 1280*g +
 1936*h + 2816*i)*x^2 + 5008*d + 8192*e + 13120*f + 20480*g + 30976*h + 450
56*i)*log(x - 2) - 9600*e - 29184*g - 61440*i)/(x^8 - 10*x^6 + 33*x^4 - 40*
x^2 + 16)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 238, normalized size of antiderivative = 1.00

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx \\
&= -\frac{1}{41472}\,(313\,d-512\,e+820\,f-1280\,g+1936\,h-2816\,i)\log\left(x+2\right) \\
&\quad+\frac{1}{1296}\,(13\,d-16\,e+25\,f-40\,g+61\,h-88\,i)\log\left(x+1\right) \\
&\quad-\frac{1}{1296}\,(13\,d+16\,e+25\,f+40\,g+61\,h+88\,i)\log\left(x-1\right) \\
&\quad+\frac{1}{41472}\,(313\,d+512\,e+820\,f+1280\,g+1936\,h+2816\,i)\log\left(x-2\right) \\
&\quad+\frac{5\,(7\,d+28\,f+64\,h)x^7+64\,(2\,e+5\,g+11\,i)x^6-18\,(13\,d+60\,f+136\,h)x^5-480\,(2\,e+5\,g+11\,i)x^4}{3456\,(x^8-10\,x^6}
\end{aligned}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="max
ima")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g + 1936*h - 2816*i)*log(x + 2) + 1/
1296*(13*d - 16*e + 25*f - 40*g + 61*h - 88*i)*log(x + 1) - 1/1296*(13*d +
16*e + 25*f + 40*g + 61*h + 88*i)*log(x - 1) + 1/41472*(313*d + 512*e + 820
*f + 1280*g + 1936*h + 2816*i)*log(x - 2) + 1/3456*(5*(7*d + 28*f + 64*h)*x
^7 + 64*(2*e + 5*g + 11*i)*x^6 - 18*(13*d + 60*f + 136*h)*x^5 - 480*(2*e +
5*g + 11*i)*x^4 + 63*(5*d + 36*f + 80*h)*x^3 + 192*(10*e + 25*g + 52*i)*x^2
 + 4*(43*d - 260*f - 656*h)*x - 800*e - 2432*g - 5120*i)/(x^8 - 10*x^6 + 33
*x^4 - 40*x^2 + 16)
```

Giac [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 249, normalized size of antiderivative = 1.04

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx$$
$$= -\frac{1}{41472}(313\,d-512\,e+820\,f-1280\,g+1936\,h-2816\,i)\log(|x+2|)$$
$$+\frac{1}{1296}(13\,d-16\,e+25\,f-40\,g+61\,h-88\,i)\log(|x+1|)$$
$$-\frac{1}{1296}(13\,d+16\,e+25\,f+40\,g+61\,h+88\,i)\log(|x-1|)$$
$$+\frac{1}{41472}(313\,d+512\,e+820\,f+1280\,g+1936\,h+2816\,i)\log(|x-2|)$$
$$+\frac{35\,dx^7+140\,fx^7+320\,hx^7+128\,ex^6+320\,gx^6+704\,ix^6-234\,dx^5-1080\,fx^5-2448\,hx^5-960\,ex^4-}{}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^3,x, algorithm="gia
c")
```

```
[Out] -1/41472*(313*d - 512*e + 820*f - 1280*g + 1936*h - 2816*i)*log(abs(x + 2))
 + 1/1296*(13*d - 16*e + 25*f - 40*g + 61*h - 88*i)*log(abs(x + 1)) - 1/129
6*(13*d + 16*e + 25*f + 40*g + 61*h + 88*i)*log(abs(x - 1)) + 1/41472*(313*
d + 512*e + 820*f + 1280*g + 1936*h + 2816*i)*log(abs(x - 2)) + 1/3456*(35*
d*x^7 + 140*f*x^7 + 320*h*x^7 + 128*e*x^6 + 320*g*x^6 + 704*i*x^6 - 234*d*x
^5 - 1080*f*x^5 - 2448*h*x^5 - 960*e*x^4 - 2400*g*x^4 - 5280*i*x^4 + 315*d*
x^3 + 2268*f*x^3 + 5040*h*x^3 + 1920*e*x^2 + 4800*g*x^2 + 9984*i*x^2 + 172*
d*x - 1040*f*x - 2624*h*x - 800*e - 2432*g - 5120*i)/(x^4 - 5*x^2 + 4)^2
```

Mupad [B] (verification not implemented)

Time = 0.37 (sec) , antiderivative size = 233, normalized size of antiderivative = 0.97

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(4-5x^2+x^4)^3}\,dx = \ln(x+1)\left(\frac{13\,d}{1296}-\frac{e}{81}+\frac{25\,f}{1296}-\frac{5\,g}{162}+\frac{61\,h}{1296}-\frac{11\,i}{162}\right)$$
$$-\ln(x-1)\left(\frac{13\,d}{1296}+\frac{e}{81}+\frac{25\,f}{1296}+\frac{5\,g}{162}+\frac{61\,h}{1296}+\frac{11\,i}{162}\right)$$
$$-\frac{\left(-\frac{35\,d}{3456}-\frac{35\,f}{864}-\frac{5\,h}{54}\right)x^7+\left(-\frac{e}{27}-\frac{5\,g}{54}-\frac{11\,i}{54}\right)x^6+\left(\frac{13\,d}{192}+\frac{5\,f}{16}+\frac{17\,h}{24}\right)x^5+\left(\frac{5\,e}{18}+\frac{25\,g}{36}+\frac{55\,i}{36}\right)x^4+\left(-\frac{35\,d}{384}-\right.}{x^8-10\,x^6+33\,x^4-40\,x^2}$$
$$+\ln(x-2)\left(\frac{313\,d}{41472}+\frac{e}{81}+\frac{205\,f}{10368}+\frac{5\,g}{162}+\frac{121\,h}{2592}+\frac{11\,i}{162}\right)$$
$$-\ln(x+2)\left(\frac{313\,d}{41472}-\frac{e}{81}+\frac{205\,f}{10368}-\frac{5\,g}{162}+\frac{121\,h}{2592}-\frac{11\,i}{162}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^4 - 5*x^2 + 4)^3,x)
```

```
[Out] log(x + 1)*((13*d)/1296 - e/81 + (25*f)/1296 - (5*g)/162 + (61*h)/1296 - (1
1*i)/162) - log(x - 1)*((13*d)/1296 + e/81 + (25*f)/1296 + (5*g)/162 + (61*
h)/1296 + (11*i)/162) - ((25*e)/108 + (19*g)/27 + (40*i)/27 + x*((65*f)/216
 - (43*d)/864 + (41*h)/54) + x^5*((13*d)/192 + (5*f)/16 + (17*h)/24) - x^3*
((35*d)/384 + (21*f)/32 + (35*h)/24) - x^7*((35*d)/3456 + (35*f)/864 + (5*h
)/54) - x^2*((5*e)/9 + (25*g)/18 + (26*i)/9) - x^6*(e/27 + (5*g)/54 + (11*i
)/54) + x^4*((5*e)/18 + (25*g)/36 + (55*i)/36))/(33*x^4 - 40*x^2 - 10*x^6 +
 x^8 + 16) + log(x - 2)*((313*d)/41472 + e/81 + (205*f)/10368 + (5*g)/162 +
 (121*h)/2592 + (11*i)/162) - log(x + 2)*((313*d)/41472 - e/81 + (205*f)/10
368 - (5*g)/162 + (121*h)/2592 - (11*i)/162)
```

3.47 $\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx$

Optimal result

Integrand size = 16, antiderivative size = 185

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = \frac{dx(1-x^2)}{12(1+x^2+x^4)^2} + \frac{e(1+2x^2)}{12(1+x^2+x^4)^2} + \frac{dx(2-7x^2)}{24(1+x^2+x^4)} + \frac{e(1+2x^2)}{6(1+x^2+x^4)} - \frac{13d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{13d\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{2e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{9}{32}d\log\left(1-x+x^2\right) + \frac{9}{32}d\log\left(1+x+x^2\right)$$

```
[Out] 1/12*d*x*(-x^2+1)/(x^4+x^2+1)^2+1/12*e*(2*x^2+1)/(x^4+x^2+1)^2+1/24*d*x*(-7
*x^2+2)/(x^4+x^2+1)+1/6*e*(2*x^2+1)/(x^4+x^2+1)-9/32*d*ln(x^2-x+1)+9/32*d*l
n(x^2+x+1)-13/144*d*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+13/144*d*arctan(1/3
*(1+2*x)*3^(1/2))*3^(1/2)+2/9*e*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 185, normalized size of antiderivative = 1.00, number of steps used = 19, number of rules used = 11, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.688, Rules used = {1687, 12, 1106, 1192, 1183, 648, 632, 210, 642, 1121, 628}

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = -\frac{13d\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{13d\arctan\left(\frac{2x+1}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{2e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{9}{32}d\log\left(x^2-x+1\right) + \frac{9}{32}d\log\left(x^2+x+1\right) + \frac{dx(2-7x^2)}{24(x^4+x^2+1)} + \frac{dx(1-x^2)}{12(x^4+x^2+1)^2} + \frac{e(2x^2+1)}{6(x^4+x^2+1)} + \frac{e(2x^2+1)}{12(x^4+x^2+1)^2}$$

```
[In] Int[(d + e*x)/(1 + x^2 + x^4)^3,x]
```

```
[Out] (d*x*(1 - x^2))/(12*(1 + x^2 + x^4)^2) + (e*(1 + 2*x^2))/(12*(1 + x^2 + x^4
)^2) + (d*x*(2 - 7*x^2))/(24*(1 + x^2 + x^4)) + (e*(1 + 2*x^2))/(6*(1 + x^2
 + x^4)) - (13*d*ArcTan[(1 - 2*x)/Sqrt[3]])/(48*Sqrt[3]) + (13*d*ArcTan[(1
+ 2*x)/Sqrt[3]])/(48*Sqrt[3]) + (2*e*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3
]) - (9*d*Log[1 - x + x^2])/32 + (9*d*Log[1 + x + x^2])/32
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d}{\left(1+x^2+x^4\right)^3}\,dx + \int \frac{ex}{\left(1+x^2+x^4\right)^3}\,dx \\
&= d\int \frac{1}{\left(1+x^2+x^4\right)^3}\,dx + e\int \frac{x}{\left(1+x^2+x^4\right)^3}\,dx \\
&= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{1}{12}d\int \frac{11-5x^2}{\left(1+x^2+x^4\right)^2}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{\left(1+x+x^2\right)^3}\,dx, x, x^2\right)
\end{aligned}
$$

$$= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{dx(2-7x^2)}{24\left(1+x^2+x^4\right)} + \frac{1}{72} d \int \frac{60-21x^2}{1+x^2+x^4}\,dx + \frac{1}{2} e\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right)$$

$$= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{dx(2-7x^2)}{24\left(1+x^2+x^4\right)} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} + \frac{1}{144} d \int \frac{60-81x}{1-x+x^2}\,dx + \frac{1}{144} d \int \frac{60+81x}{1+x+x^2}\,dx + \frac{1}{3} e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right)$$

$$= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{dx(2-7x^2)}{24\left(1+x^2+x^4\right)} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} + \frac{1}{96}(13d) \int \frac{1}{1-x+x^2}\,dx + \frac{1}{96}(13d) \int \frac{1}{1+x+x^2}\,dx - \frac{1}{32}(9d) \int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{32}(9d) \int \frac{1+2x}{1+x+x^2}\,dx - \frac{1}{3}(2e)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right)$$

$$= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{dx(2-7x^2)}{24\left(1+x^2+x^4\right)} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{9}{32} d \log\left(1-x+x^2\right) + \frac{9}{32} d \log\left(1+x+x^2\right) - \frac{1}{48}(13d)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) - \frac{1}{48}(13d)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)$$

$$= \frac{dx(1-x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{dx(2-7x^2)}{24\left(1+x^2+x^4\right)} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} - \frac{13d\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{13d\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{9}{32} d \log\left(1-x+x^2\right) + \frac{9}{32} d \log\left(1+x+x^2\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.53 (sec) , antiderivative size = 186, normalized size of antiderivative = 1.01

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\left(\frac{12(e+dx+2ex^2-dx^3)}{(1+x^2+x^4)^2}+\frac{6(dx(2-7x^2)+e(4+8x^2))}{1+x^2+x^4} - \frac{\left(-47i+7\sqrt{3}\right)d\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(47i+7\sqrt{3}\right)d\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 32\sqrt{3}e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x)/(1 + x^2 + x^4)^3,x]
```

```
[Out] ((12*(e + d*x + 2*e*x^2 - d*x^3))/(1 + x^2 + x^4)^2 + (6*(d*x*(2 - 7*x^2) +
 e*(4 + 8*x^2)))/(1 + x^2 + x^4) - ((-47*I + 7*Sqrt[3])*d*ArcTan[((-I + Sqr
t[3])*x)/2])/Sqrt[(1 + I*Sqrt[3])/6] - ((47*I + 7*Sqrt[3])*d*ArcTan[((I + S
qrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 32*Sqrt[3]*e*ArcTan[Sqrt[3]/(1 + 2
*x^2)])/144
```

Maple [A] (verified)

Time = 0.18 (sec) , antiderivative size = 158, normalized size of antiderivative = 0.85

method	result
default	$-\frac{\left(\frac{7d}{3}-\frac{4e}{3}\right)x^3-6d\,x^2+\left(\frac{20d}{3}+\frac{e}{3}\right)x-4d-2e}{16(x^2-x+1)^2}-\frac{9d\ln\left(x^2-x+1\right)}{32}-\frac{\left(-\frac{13d}{2}-16e\right)\sqrt{3}\,\arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)}{72}+\frac{\left(-\frac{7d}{3}-\frac{4e}{3}\right)x^3-6d\,x^2+}{16(x^2+x}$
risch	$-\frac{9d\ln\left(15457716d^2x^2+7237632e^2x^2-15457716d^2x-7237632e^2x+15457716d^2+7237632e^2\right)}{32}+\frac{13\sqrt{3}\,d\arctan\left(\frac{1458d^2x\sqrt{3}}{2187d^2+1024e^2}+\frac{}{3(2}\right.}{}$

```
[In] int((e*x+d)/(x^4+x^2+1)^3,x,method=_RETURNVERBOSE)
```

```
[Out] -1/16*((7/3*d-4/3*e)*x^3-6*d*x^2+(20/3*d+1/3*e)*x-4*d-2*e)/(x^2-x+1)^2-9/32
*d*ln(x^2-x+1)-1/72*(-13/2*d-16*e)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/16
*((-7/3*d-4/3*e)*x^3-6*d*x^2+(-20/3*d+1/3*e)*x-4*d+2*e)/(x^2+x+1)^2+9/32*d*
ln(x^2+x+1)+1/72*(13/2*d-16*e)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 278, normalized size of antiderivative = 1.50

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = -\frac{84\,dx^7-96\,ex^6+60\,dx^5-144\,ex^4+84\,dx^3-192\,ex^2-2\,\sqrt{3}((13\,d-32\,e)x^8+2\,(13\,d-32\,e)x^6+3\,(}{}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^3,x, algorithm="fricas")
```

```
[Out] -1/288*(84*d*x^7 - 96*e*x^6 + 60*d*x^5 - 144*e*x^4 + 84*d*x^3 - 192*e*x^2 -
 2*sqrt(3)*((13*d - 32*e)*x^8 + 2*(13*d - 32*e)*x^6 + 3*(13*d - 32*e)*x^4 +
 2*(13*d - 32*e)*x^2 + 13*d - 32*e)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(
3)*((13*d + 32*e)*x^8 + 2*(13*d + 32*e)*x^6 + 3*(13*d + 32*e)*x^4 + 2*(13*d
 + 32*e)*x^2 + 13*d + 32*e)*arctan(1/3*sqrt(3)*(2*x - 1)) - 48*d*x - 81*(d*
x^8 + 2*d*x^6 + 3*d*x^4 + 2*d*x^2 + d)*log(x^2 + x + 1) + 81*(d*x^8 + 2*d*x
^6 + 3*d*x^4 + 2*d*x^2 + d)*log(x^2 - x + 1) - 72*e)/(x^8 + 2*x^6 + 3*x^4 +
 2*x^2 + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.19 (sec) , antiderivative size = 1103, normalized size of antiderivative = 5.96

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)/(x**4+x**2+1)**3,x)
```

```
[Out] (-9*d/32 - sqrt(3)*I*(13*d + 32*e)/288)*log(x + (-1025428432*d**4*e - 33475
2912*d**4*(-9*d/32 - sqrt(3)*I*(13*d + 32*e)/288) - 431308800*d**2*e**3 - 3
143688192*d**2*e**2*(-9*d/32 - sqrt(3)*I*(13*d + 32*e)/288) + 9917005824*d*
*2*e*(-9*d/32 - sqrt(3)*I*(13*d + 32*e)/288)**2 + 11878244352*d**2*(-9*d/32
 - sqrt(3)*I*(13*d + 32*e)/288)**3 + 142606336*e**5 + 754974720*e**4*(-9*d/
32 - sqrt(3)*I*(13*d + 32*e)/288) + 3850371072*e**3*(-9*d/32 - sqrt(3)*I*(1
3*d + 32*e)/288)**2 + 20384317440*e**2*(-9*d/32 - sqrt(3)*I*(13*d + 32*e)/2
88)**3)/(217696167*d**5 - 1217128448*d**3*e**2 - 617611264*d*e**4)) + (-9*d
/32 + sqrt(3)*I*(13*d + 32*e)/288)*log(x + (-1025428432*d**4*e - 334752912*
d**4*(-9*d/32 + sqrt(3)*I*(13*d + 32*e)/288) - 431308800*d**2*e**3 - 314368
8192*d**2*e**2*(-9*d/32 + sqrt(3)*I*(13*d + 32*e)/288) + 9917005824*d**2*e*
(-9*d/32 + sqrt(3)*I*(13*d + 32*e)/288)**2 + 11878244352*d**2*(-9*d/32 + sq
rt(3)*I*(13*d + 32*e)/288)**3 + 142606336*e**5 + 754974720*e**4*(-9*d/32 +
```

```
sqrt(3)*I*(13*d + 32*e)/288) + 3850371072*e**3*(-9*d/32 + sqrt(3)*I*(13*d +
 32*e)/288)**2 + 20384317440*e**2*(-9*d/32 + sqrt(3)*I*(13*d + 32*e)/288)**
3)/(217696167*d**5 - 1217128448*d**3*e**2 - 617611264*d*e**4)) + (9*d/32 -
sqrt(3)*I*(13*d - 32*e)/288)*log(x + (-1025428432*d**4*e - 334752912*d**4*(
9*d/32 - sqrt(3)*I*(13*d - 32*e)/288) - 431308800*d**2*e**3 - 3143688192*d*
*2*e**2*(9*d/32 - sqrt(3)*I*(13*d - 32*e)/288) + 9917005824*d**2*e*(9*d/32
- sqrt(3)*I*(13*d - 32*e)/288)**2 + 11878244352*d**2*(9*d/32 - sqrt(3)*I*(1
3*d - 32*e)/288)**3 + 142606336*e**5 + 754974720*e**4*(9*d/32 - sqrt(3)*I*(
13*d - 32*e)/288) + 3850371072*e**3*(9*d/32 - sqrt(3)*I*(13*d - 32*e)/288)*
*2 + 20384317440*e**2*(9*d/32 - sqrt(3)*I*(13*d - 32*e)/288)**3)/(217696167
*d**5 - 1217128448*d**3*e**2 - 617611264*d*e**4)) + (9*d/32 + sqrt(3)*I*(13
*d - 32*e)/288)*log(x + (-1025428432*d**4*e - 334752912*d**4*(9*d/32 + sqrt
(3)*I*(13*d - 32*e)/288) - 431308800*d**2*e**3 - 3143688192*d**2*e**2*(9*d/
32 + sqrt(3)*I*(13*d - 32*e)/288) + 9917005824*d**2*e*(9*d/32 + sqrt(3)*I*(
13*d - 32*e)/288)**2 + 11878244352*d**2*(9*d/32 + sqrt(3)*I*(13*d - 32*e)/2
88)**3 + 142606336*e**5 + 754974720*e**4*(9*d/32 + sqrt(3)*I*(13*d - 32*e)/
288) + 3850371072*e**3*(9*d/32 + sqrt(3)*I*(13*d - 32*e)/288)**2 + 20384317
440*e**2*(9*d/32 + sqrt(3)*I*(13*d - 32*e)/288)**3)/(217696167*d**5 - 12171
28448*d**3*e**2 - 617611264*d*e**4)) + (-7*d*x**7 - 5*d*x**5 - 7*d*x**3 + 4
*d*x + 8*e*x**6 + 12*e*x**4 + 16*e*x**2 + 6*e)/(24*x**8 + 48*x**6 + 72*x**4
 + 48*x**2 + 24)
```

Maxima [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 137, normalized size of antiderivative = 0.74

$$\begin{aligned}\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx =& \frac{1}{144}\sqrt{3}(13\,d-32\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)\\&+\frac{1}{144}\sqrt{3}(13\,d+32\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)\\&+\frac{9}{32}\,d\log\left(x^2+x+1\right)-\frac{9}{32}\,d\log\left(x^2-x+1\right)\\&-\frac{7\,dx^7-8\,ex^6+5\,dx^5-12\,ex^4+7\,dx^3-16\,ex^2-4\,dx-6\,e}{24\,(x^8+2\,x^6+3\,x^4+2\,x^2+1)}\end{aligned}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^3,x, algorithm="maxima")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/144*sqrt(3)*(
13*d + 32*e)*arctan(1/3*sqrt(3)*(2*x - 1)) + 9/32*d*log(x^2 + x + 1) - 9/32
*d*log(x^2 - x + 1) - 1/24*(7*d*x^7 - 8*e*x^6 + 5*d*x^5 - 12*e*x^4 + 7*d*x^
3 - 16*e*x^2 - 4*d*x - 6*e)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 125, normalized size of antiderivative = 0.68

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\sqrt{3}(13\,d-32\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{144}\sqrt{3}(13\,d+32\,e)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{9}{32}\,d\log\left(x^2+x+1\right) - \frac{9}{32}\,d\log\left(x^2-x+1\right) - \frac{7\,dx^7-8\,ex^6+5\,dx^5-12\,ex^4+7\,dx^3-16\,ex^2-4\,dx-6\,e}{24\,(x^4+x^2+1)^2}$$

```
[In] integrate((e*x+d)/(x^4+x^2+1)^3,x, algorithm="giac")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/144*sqrt(3)*(
13*d + 32*e)*arctan(1/3*sqrt(3)*(2*x - 1)) + 9/32*d*log(x^2 + x + 1) - 9/32
*d*log(x^2 - x + 1) - 1/24*(7*d*x^7 - 8*e*x^6 + 5*d*x^5 - 12*e*x^4 + 7*d*x^
3 - 16*e*x^2 - 4*d*x - 6*e)/(x^4 + x^2 + 1)^2
```

Mupad [B] (verification not implemented)

Time = 0.15 (sec) , antiderivative size = 185, normalized size of antiderivative = 1.00

$$\int \frac{d+ex}{(1+x^2+x^4)^3}\,dx = \frac{-\frac{7\,d\,x^7}{24}+\frac{e\,x^6}{3}-\frac{5\,d\,x^5}{24}+\frac{e\,x^4}{2}-\frac{7\,d\,x^3}{24}+\frac{2\,e\,x^2}{3}+\frac{d\,x}{6}+\frac{e}{4}}{x^8+2\,x^6+3\,x^4+2\,x^2+1} - \ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right) + \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{9\,d}{32}-\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(-\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\left(\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}\right)$$

```
[In] int((d + e*x)/(x^2 + x^4 + 1)^3,x)
```

```
[Out] (e/4 + (d*x)/6 - (7*d*x^3)/24 - (5*d*x^5)/24 - (7*d*x^7)/24 + (2*e*x^2)/3 +
 (e*x^4)/2 + (e*x^6)/3)/(2*x^2 + 3*x^4 + 2*x^6 + x^8 + 1) - log(x - (3^(1/2
)*1i)/2 - 1/2)*((9*d)/32 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9) + log(x
```

```
- (3^(1/2)*1i)/2 + 1/2)*((9*d)/32 - (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9)
 + log(x + (3^(1/2)*1i)/2 - 1/2)*((3^(1/2)*d*13i)/288 - (9*d)/32 + (3^(1/2)
*e*1i)/9) + log(x + (3^(1/2)*1i)/2 + 1/2)*((9*d)/32 + (3^(1/2)*d*13i)/288 -
 (3^(1/2)*e*1i)/9)
```

3.48 $\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3} dx$

Optimal result

Integrand size = 21, antiderivative size = 223

$$\begin{aligned}\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3} dx &= \frac{e(1+2x^2)}{12(1+x^2+x^4)^2} + \frac{x(d+f-(d-2f)x^2)}{12(1+x^2+x^4)^2} + \frac{e(1+2x^2)}{6(1+x^2+x^4)} \\ &+ \frac{x(2d+3f-7(d-f)x^2)}{24(1+x^2+x^4)} - \frac{(13d+2f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} \\ &+ \frac{(13d+2f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{2e\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} \\ &- \frac{1}{32}(9d-4f)\log\left(1-x+x^2\right) + \frac{1}{32}(9d-4f)\log\left(1+x+x^2\right)\end{aligned}$$

```
[Out] 1/12*e*(2*x^2+1)/(x^4+x^2+1)^2+1/12*x*(d+f-(d-2*f)*x^2)/(x^4+x^2+1)^2+1/6*e
*(2*x^2+1)/(x^4+x^2+1)+1/24*x*(2*d+3*f-7*(d-f)*x^2)/(x^4+x^2+1)-1/32*(9*d-4
*f)*ln(x^2-x+1)+1/32*(9*d-4*f)*ln(x^2+x+1)-1/144*(13*d+2*f)*arctan(1/3*(1-2
*x)*3^(1/2))*3^(1/2)+1/144*(13*d+2*f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+2
/9*e*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 223, normalized size of antiderivative = 1.00, number of steps used = 18, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}} = 0.476$, Rules

used = {1687, 1192, 1183, 648, 632, 210, 642, 12, 1121, 628}

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(13d+2f)}{48\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(13d+2f)}{48\sqrt{3}} + \frac{2e\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f)\log\left(x^2-x+1\right) + \frac{1}{32}(9d-4f)\log\left(x^2+x+1\right) + \frac{x(-7x^2(d-f)+2d+3f)}{24\left(x^4+x^2+1\right)} + \frac{x(-(x^2(d-2f))+d+f)}{12\left(x^4+x^2+1\right)^2} + \frac{e(2x^2+1)}{6\left(x^4+x^2+1\right)} + \frac{e(2x^2+1)}{12\left(x^4+x^2+1\right)^2}$$

```
[In] Int[(d + e*x + f*x^2)/(1 + x^2 + x^4)^3,x]
```

```
[Out] (e*(1 + 2*x^2))/(12*(1 + x^2 + x^4)^2) + (x*(d + f - (d - 2*f)*x^2))/(12*(1
 + x^2 + x^4)^2) + (e*(1 + 2*x^2))/(6*(1 + x^2 + x^4)) + (x*(2*d + 3*f - 7*
(d - f)*x^2))/(24*(1 + x^2 + x^4)) - ((13*d + 2*f)*ArcTan[(1 - 2*x)/Sqrt[3]
])/(48*Sqrt[3]) + ((13*d + 2*f)*ArcTan[(1 + 2*x)/Sqrt[3]])/(48*Sqrt[3]) + (
2*e*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((9*d - 4*f)*Log[1 - x + x^2
])/32 + ((9*d - 4*f)*Log[1 + x + x^2])/32
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{ex}{(1+x^2+x^4)^3}\,dx + \int \frac{d+fx^2}{(1+x^2+x^4)^3}\,dx \\
&= \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{1}{12}\int \frac{11d-f-5(d-2f)x^2}{(1+x^2+x^4)^2}\,dx + e\int \frac{x}{(1+x^2+x^4)^3}\,dx \\
&= \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} \\
&\quad + \frac{1}{72}\int \frac{15(4d-f)-21(d-f)x^2}{1+x^2+x^4}\,dx + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(1+x+x^2)^3}\,dx, x, x^2\right) \\
&= \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} \\
&\quad + \frac{1}{144}\int \frac{15(4d-f)-(21(d-f)+15(4d-f))x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{144}\int \frac{15(4d-f)+(21(d-f)+15(4d-f))x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} + \frac{1}{3}e\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&\quad + \frac{1}{32}(9d-4f)\int \frac{1+2x}{1+x+x^2}\,dx + \frac{1}{96}(13d+2f)\int \frac{1}{1-x+x^2}\,dx \\
&\quad + \frac{1}{96}(13d+2f)\int \frac{1}{1+x+x^2}\,dx + \frac{1}{32}(-9d+4f)\int \frac{-1+2x}{1-x+x^2}\,dx \\
&= \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} - \frac{1}{32}(9d-4f)\log\left(1-x+x^2\right) \\
&\quad + \frac{1}{32}(9d-4f)\log\left(1+x+x^2\right) - \frac{1}{3}(2e)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) \\
&\quad + \frac{1}{48}(-13d-2f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) \\
&\quad + \frac{1}{48}(-13d-2f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)
\end{aligned}
$$

$$= \frac{e(1+2x^2)}{12\left(1+x^2+x^4\right)^2}+\frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2}+\frac{e(1+2x^2)}{6\left(1+x^2+x^4\right)} + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)}-\frac{(13d+2f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}}+\frac{(13d+2f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{2e\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{32}(9d-4f)\log\left(1-x+x^2\right)+\frac{1}{32}(9d-4f)\log\left(1+x+x^2\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.39 (sec) , antiderivative size = 235, normalized size of antiderivative = 1.05

$$\int \frac{d+ex+fx^2}{\left(1+x^2+x^4\right)^3}\,dx = \frac{1}{144}\left(\frac{6(2dx+3fx-7dx^3+7fx^3+e(4+8x^2))}{1+x^2+x^4} + \frac{12(e+2ex^2+x(d+f-dx^2+2fx^2))}{\left(1+x^2+x^4\right)^2} - \frac{\left(\left(-47i+7\sqrt{3}\right)d+\left(17i-7\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(\left(47i+7\sqrt{3}\right)d-\left(17i+7\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 32\sqrt{3}e\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2)/(1 + x^2 + x^4)^3,x]
```

```
[Out] ((6*(2*d*x + 3*f*x - 7*d*x^3 + 7*f*x^3 + e*(4 + 8*x^2)))/(1 + x^2 + x^4) +
(12*(e + 2*e*x^2 + x*(d + f - d*x^2 + 2*f*x^2)))/(1 + x^2 + x^4)^2 - (((-47
*I + 7*Sqrt[3])*d + (17*I - 7*Sqrt[3])*f)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqr
t[(1 + I*Sqrt[3])/6] - (((47*I + 7*Sqrt[3])*d - (17*I + 7*Sqrt[3])*f)*ArcTa
n[((I + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 32*Sqrt[3]*e*ArcTan[Sqrt[
3]/(1 + 2*x^2)])/144
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 202, normalized size of antiderivative = 0.91

method	result
default	$-\frac{\left(\frac{7d}{3}-\frac{7f}{3}-\frac{4e}{3}\right)x^3+(-6d+4f)x^2+\left(\frac{20d}{3}-\frac{13f}{3}+\frac{e}{3}\right)x-4d+\frac{4f}{3}-2e}{16(x^2-x+1)^2}-\frac{(27d-12f)\ln(x^2-x+1)}{96}-\frac{\left(-\frac{13d}{2}-16e-f\right)\sqrt{3}\arctan\left(\frac{(2x-}{3}\right.}{72}$
risch	Expression too large to display

```
[In] int((f*x^2+e*x+d)/(x^4+x^2+1)^3,x,method=_RETURNVERBOSE)
```

```
[Out] -1/16*((7/3*d-7/3*f-4/3*e)*x^3+(-6*d+4*f)*x^2+(20/3*d-13/3*f+1/3*e)*x-4*d+4
/3*f-2*e)/(x^2-x+1)^2-1/96*(27*d-12*f)*ln(x^2-x+1)-1/72*(-13/2*d-16*e-f)*3^
(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/16*((-7/3*d+7/3*f-4/3*e)*x^3+(-6*d+4*f)
*x^2+(-20/3*d+13/3*f+1/3*e)*x-4*d+4/3*f+2*e)/(x^2+x+1)^2+1/96*(27*d-12*f)*l
n(x^2+x+1)+1/72*(13/2*d-16*e+f)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 384, normalized size of antiderivative = 1.72

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = -\frac{84\,(d-f)x^7-96\,ex^6+60\,(d-2\,f)x^5-144\,ex^4+84\,(d-2\,f)x^3-192\,ex^2-2\,\sqrt{3}((13\,d-32\,e+2\,f)x}{}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="fricas")
```

```
[Out] -1/288*(84*(d - f)*x^7 - 96*e*x^6 + 60*(d - 2*f)*x^5 - 144*e*x^4 + 84*(d -
2*f)*x^3 - 192*e*x^2 - 2*sqrt(3)*((13*d - 32*e + 2*f)*x^8 + 2*(13*d - 32*e
+ 2*f)*x^6 + 3*(13*d - 32*e + 2*f)*x^4 + 2*(13*d - 32*e + 2*f)*x^2 + 13*d -
 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*((13*d + 32*e + 2*f)
*x^8 + 2*(13*d + 32*e + 2*f)*x^6 + 3*(13*d + 32*e + 2*f)*x^4 + 2*(13*d + 32
*e + 2*f)*x^2 + 13*d + 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x - 1)) - 12*(4*d
+ 5*f)*x - 9*((9*d - 4*f)*x^8 + 2*(9*d - 4*f)*x^6 + 3*(9*d - 4*f)*x^4 + 2*(
9*d - 4*f)*x^2 + 9*d - 4*f)*log(x^2 + x + 1) + 9*((9*d - 4*f)*x^8 + 2*(9*d
- 4*f)*x^6 + 3*(9*d - 4*f)*x^4 + 2*(9*d - 4*f)*x^2 + 9*d - 4*f)*log(x^2 - x
 + 1) - 72*e)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 138.76 (sec) , antiderivative size = 4496, normalized size of antiderivative = 20.16

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x**2+e*x+d)/(x**4+x**2+1)**3,x)
```

```
[Out] (-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)*log(x + (-1025428432*d*
*5*e - 334752912*d**5*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) +
 2008961360*d**4*e*f + 1151575920*d**4*f*(-9*d/32 + f/8 - sqrt(3)*I*(13*d +
 32*e + 2*f)/288) - 431308800*d**3*e**3 - 3143688192*d**3*e**2*(-9*d/32 + f
/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 1598857120*d**3*e*f**2 + 99170058
24*d**3*e*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 - 94430016
0*d**3*f**2*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) + 118782443
52*d**3*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 233164800*
d**2*e**3*f + 4409634816*d**2*e**2*f*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*
e + 2*f)/288) + 662937520*d**2*e*f**3 - 13004623872*d**2*e*f*(-9*d/32 + f/8
 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 + 231796080*d**2*f**3*(-9*d/32 + f
/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 10089639936*d**2*f*(-9*d/32 + f/8
 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 142606336*d*e**5 + 754974720*d*e
**4*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 1843200*d*e**3*f*
*2 + 3850371072*d*e**3*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)*
*2 - 1926291456*d*e**2*f**2*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/
288) + 20384317440*d*e**2*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/28
8)**3 - 146756960*d*e*f**4 + 5813379072*d*e*f**2*(-9*d/32 + f/8 - sqrt(3)*I
*(13*d + 32*e + 2*f)/288)**2 + 12679200*d*f**4*(-9*d/32 + f/8 - sqrt(3)*I*(
13*d + 32*e + 2*f)/288) + 1116758016*d*f**2*(-9*d/32 + f/8 - sqrt(3)*I*(13*
d + 32*e + 2*f)/288)**3 - 79691776*e**5*f - 188743680*e**4*f*(-9*d/32 + f/8
 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 7372800*e**3*f**3 - 2151677952*e**3
*f*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 + 287096832*e**2*
f**3*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 5096079360*e**2*
f*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 14093632*e*f**5
- 859521024*e*f**3*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 -
 7648128*f**5*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288) + 4538695
68*f**3*(-9*d/32 + f/8 - sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3)/(217696167*
d**6 - 301346487*d**5*f - 1217128448*d**4*e**2 + 130506255*d**4*f**2 + 2181
281792*d**3*e**2*f - 5619240*d**3*f**3 - 617611264*d**2*e**4 - 1450149888*d
**2*e**2*f**2 - 8036820*d**2*f**4 + 495976448*d*e**4*f + 430088192*d*e**2*f
**3 + 783648*d*f**5 - 114294784*e**4*f**2 - 47771648*e**2*f**4 + 188352*f**
6)) + (-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)*log(x + (-1025428
432*d**5*e - 334752912*d**5*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/
288) + 2008961360*d**4*e*f + 1151575920*d**4*f*(-9*d/32 + f/8 + sqrt(3)*I*(
```

```
13*d + 32*e + 2*f)/288) - 431308800*d**3*e**3 - 3143688192*d**3*e**2*(-9*d/
32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 1598857120*d**3*e*f**2 + 99
17005824*d**3*e*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 - 94
4300160*d**3*f**2*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) + 118
78244352*d**3*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 2331
64800*d**2*e**3*f + 4409634816*d**2*e**2*f*(-9*d/32 + f/8 + sqrt(3)*I*(13*d
 + 32*e + 2*f)/288) + 662937520*d**2*e*f**3 - 13004623872*d**2*e*f*(-9*d/32
 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 + 231796080*d**2*f**3*(-9*d/
32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 10089639936*d**2*f*(-9*d/32
 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 142606336*d*e**5 + 7549747
20*d*e**4*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 1843200*d*e
**3*f**2 + 3850371072*d*e**3*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)
/288)**2 - 1926291456*d*e**2*f**2*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e +
 2*f)/288) + 20384317440*d*e**2*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2
*f)/288)**3 - 146756960*d*e*f**4 + 5813379072*d*e*f**2*(-9*d/32 + f/8 + sqr
t(3)*I*(13*d + 32*e + 2*f)/288)**2 + 12679200*d*f**4*(-9*d/32 + f/8 + sqrt(
3)*I*(13*d + 32*e + 2*f)/288) + 1116758016*d*f**2*(-9*d/32 + f/8 + sqrt(3)*
I*(13*d + 32*e + 2*f)/288)**3 - 79691776*e**5*f - 188743680*e**4*f*(-9*d/32
 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 7372800*e**3*f**3 - 215167795
2*e**3*f*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**2 + 287096832
*e**2*f**3*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) - 5096079360
*e**2*f*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3 + 14093632*e
*f**5 - 859521024*e*f**3*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288
)**2 - 7648128*f**5*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288) + 4
53869568*f**3*(-9*d/32 + f/8 + sqrt(3)*I*(13*d + 32*e + 2*f)/288)**3)/(2176
96167*d**6 - 301346487*d**5*f - 1217128448*d**4*e**2 + 130506255*d**4*f**2
+ 2181281792*d**3*e**2*f - 5619240*d**3*f**3 - 617611264*d**2*e**4 - 145014
9888*d**2*e**2*f**2 - 8036820*d**2*f**4 + 495976448*d*e**4*f + 430088192*d*
e**2*f**3 + 783648*d*f**5 - 114294784*e**4*f**2 - 47771648*e**2*f**4 + 1883
52*f**6)) + (9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)*log(x + (-10
25428432*d**5*e - 334752912*d**5*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2
*f)/288) + 2008961360*d**4*e*f + 1151575920*d**4*f*(9*d/32 - f/8 - sqrt(3)*
I*(13*d - 32*e + 2*f)/288) - 431308800*d**3*e**3 - 3143688192*d**3*e**2*(9*
d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) - 1598857120*d**3*e*f**2 +
9917005824*d**3*e*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 - 9
44300160*d**3*f**2*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 118
78244352*d**3*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 + 23316
4800*d**2*e**3*f + 4409634816*d**2*e**2*f*(9*d/32 - f/8 - sqrt(3)*I*(13*d -
 32*e + 2*f)/288) + 662937520*d**2*e*f**3 - 13004623872*d**2*e*f*(9*d/32 -
f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 + 231796080*d**2*f**3*(9*d/32 -
 f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) - 10089639936*d**2*f*(9*d/32 - f/
8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 + 142606336*d*e**5 + 754974720*d*
e**4*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) - 1843200*d*e**3*f*
*2 + 3850371072*d*e**3*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**
2 - 1926291456*d*e**2*f**2*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/28
```

8) + 20384317440*d*e**2*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)*
*3 - 146756960*d*e*f**4 + 5813379072*d*e*f**2*(9*d/32 - f/8 - sqrt(3)*I*(13
*d - 32*e + 2*f)/288)**2 + 12679200*d*f**4*(9*d/32 - f/8 - sqrt(3)*I*(13*d
- 32*e + 2*f)/288) + 1116758016*d*f**2*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32
*e + 2*f)/288)**3 - 79691776*e**5*f - 188743680*e**4*f*(9*d/32 - f/8 - sqrt
(3)*I*(13*d - 32*e + 2*f)/288) - 7372800*e**3*f**3 - 2151677952*e**3*f*(9*d
/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 + 287096832*e**2*f**3*(9*
d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) - 5096079360*e**2*f*(9*d/32
- f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 + 14093632*e*f**5 - 85952102
4*e*f**3*(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 - 7648128*f*
5(9*d/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 453869568*f**3*(9*d
/32 - f/8 - sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3)/(217696167*d**6 - 301346
487*d**5*f - 1217128448*d**4*e**2 + 130506255*d**4*f**2 + 2181281792*d**3*e
2*f - 5619240*d3*f**3 - 617611264*d**2*e**4 - 1450149888*d**2*e**2*f**2
- 8036820*d**2*f**4 + 495976448*d*e**4*f + 430088192*d*e**2*f**3 + 783648*
d*f**5 - 114294784*e**4*f**2 - 47771648*e**2*f**4 + 188352*f**6)) + (9*d/32
- f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)*log(x + (-1025428432*d**5*e - 3
34752912*d**5*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 20089613
60*d**4*e*f + 1151575920*d**4*f*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*
f)/288) - 431308800*d**3*e**3 - 3143688192*d**3*e**2*(9*d/32 - f/8 + sqrt(3
)*I*(13*d - 32*e + 2*f)/288) - 1598857120*d**3*e*f**2 + 9917005824*d**3*e*(
9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 - 944300160*d**3*f**2*
(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 11878244352*d**3*(9*d/
32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 + 233164800*d**2*e**3*f +
4409634816*d**2*e**2*f*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288) +
662937520*d**2*e*f**3 - 13004623872*d**2*e*f*(9*d/32 - f/8 + sqrt(3)*I*(13
*d - 32*e + 2*f)/288)**2 + 231796080*d**2*f**3*(9*d/32 - f/8 + sqrt(3)*I*(1
3*d - 32*e + 2*f)/288) - 10089639936*d**2*f*(9*d/32 - f/8 + sqrt(3)*I*(13*d
- 32*e + 2*f)/288)**3 + 142606336*d*e**5 + 754974720*d*e**4*(9*d/32 - f/8
+ sqrt(3)*I*(13*d - 32*e + 2*f)/288) - 1843200*d*e**3*f**2 + 3850371072*d*e
3*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)2 - 1926291456*d*e*
*2*f**2*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 20384317440*d*
e**2*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 - 146756960*d*e*
f**4 + 5813379072*d*e*f**2*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/28
8)**2 + 12679200*d*f**4*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)
+ 1116758016*d*f**2*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)**3 -
79691776*e**5*f - 188743680*e**4*f*(9*d/32 - f/8 + sqrt(3)*I*(13*d - 32*e
+ 2*f)/288) - 7372800*e**3*f**3 - 2151677952*e**3*f*(9*d/32 - f/8 + sqrt(3)
I(13*d - 32*e + 2*f)/288)**2 + 287096832*e**2*f**3*(9*d/32 - f/8 + sqrt(3
)*I*(13*d - 32*e + 2*f)/288) - 5096079360*e**2*f*(9*d/32 - f/8 + sqrt(3)*I*
(13*d - 32*e + 2*f)/288)**3 + 14093632*e*f**5 - 859521024*e*f**3*(9*d/32 -
f/8 + sqrt(3)*I*(13*d - 32*e + 2*f)/288)**2 - 7648128*f**5*(9*d/32 - f/8 +
sqrt(3)*I*(13*d - 32*e + 2*f)/288) + 453869568*f**3*(9*d/32 - f/8 + sqrt(3)
I(13*d - 32*e + 2*f)/288)**3)/(217696167*d**6 - 301346487*d**5*f - 121712
8448*d**4*e**2 + 130506255*d**4*f**2 + 2181281792*d**3*e**2*f - 5619240*d**

```
3*f**3 - 617611264*d**2*e**4 - 1450149888*d**2*e**2*f**2 - 8036820*d**2*f**
4 + 495976448*d*e**4*f + 430088192*d*e**2*f**3 + 783648*d*f**5 - 114294784*
e**4*f**2 - 47771648*e**2*f**4 + 188352*f**6)) + (8*e*x**6 + 12*e*x**4 + 16
*e*x**2 + 6*e + x**7*(-7*d + 7*f) + x**5*(-5*d + 10*f) + x**3*(-7*d + 14*f)
 + x*(4*d + 5*f))/(24*x**8 + 48*x**6 + 72*x**4 + 48*x**2 + 24)
```

Maxima [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 173, normalized size of antiderivative = 0.78

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{1}{32}(9\,d-4\,f)\log\left(x^2+x+1\right) - \frac{1}{32}(9\,d-4\,f)\log\left(x^2-x+1\right) - \frac{7\,(d-f)x^7-8\,ex^6+5\,(d-2\,f)x^5-12\,ex^4+7\,(d-2\,f)x^3-16\,ex^2-(4\,d+5\,f)x-6\,e}{24\,(x^8+2\,x^6+3\,x^4+2\,x^2+1)}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="maxima")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/144*sqr
t(3)*(13*d + 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/32*(9*d - 4*f)*l
og(x^2 + x + 1) - 1/32*(9*d - 4*f)*log(x^2 - x + 1) - 1/24*(7*(d - f)*x^7 -
 8*e*x^6 + 5*(d - 2*f)*x^5 - 12*e*x^4 + 7*(d - 2*f)*x^3 - 16*e*x^2 - (4*d +
 5*f)*x - 6*e)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 165, normalized size of antiderivative = 0.74

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{1}{32}(9\,d-4\,f)\log\left(x^2+x+1\right) - \frac{1}{32}(9\,d-4\,f)\log\left(x^2-x+1\right) - \frac{7\,dx^7-7\,fx^7-8\,ex^6+5\,dx^5-10\,fx^5-12\,ex^4+7\,dx^3-14\,fx^3-16\,ex^2-4\,dx-5\,fx-6\,e}{24\,(x^4+x^2+1)^2}$$

```
[In] integrate((f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="giac")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/144*sqr
t(3)*(13*d + 32*e + 2*f)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/32*(9*d - 4*f)*l
og(x^2 + x + 1) - 1/32*(9*d - 4*f)*log(x^2 - x + 1) - 1/24*(7*d*x^7 - 7*f*x
^7 - 8*e*x^6 + 5*d*x^5 - 10*f*x^5 - 12*e*x^4 + 7*d*x^3 - 14*f*x^3 - 16*e*x^
2 - 4*d*x - 5*f*x - 6*e)/(x^4 + x^2 + 1)^2
```

Mupad [B] (verification not implemented)

Time = 7.97 (sec) , antiderivative size = 249, normalized size of antiderivative = 1.12

$$\int \frac{d+ex+fx^2}{(1+x^2+x^4)^3}\,dx = \frac{\left(\frac{7\,f}{24}-\frac{7\,d}{24}\right)\,x^7+\frac{e\,x^6}{3}+\left(\frac{5\,f}{12}-\frac{5\,d}{24}\right)\,x^5+\frac{e\,x^4}{2}+\left(\frac{7\,f}{12}-\frac{7\,d}{24}\right)\,x^3+\frac{2\,e\,x^2}{3}+\left(\frac{d}{6}+\frac{5\,f}{24}\right)\,x+\frac{e}{4}}{x^8+2\,x^6+3\,x^4+2\,x^2+1} - \ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{9\,d}{32}-\frac{f}{8}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}\right) - \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{9\,d}{32}-\frac{f}{8}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}\right)$$

```
[In] int((d + e*x + f*x^2)/(x^2 + x^4 + 1)^3,x)
```

```
[Out] (e/4 - x^5*((5*d)/24 - (5*f)/12) - x^3*((7*d)/24 - (7*f)/12) - x^7*((7*d)/2
4 - (7*f)/24) + (2*e*x^2)/3 + (e*x^4)/2 + (e*x^6)/3 + x*(d/6 + (5*f)/24))/(
2*x^2 + 3*x^4 + 2*x^6 + x^8 + 1) - log(x - (3^(1/2)*1i)/2 - 1/2)*((9*d)/32
- f/8 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144) - log(
x - (3^(1/2)*1i)/2 + 1/2)*(f/8 - (9*d)/32 + (3^(1/2)*d*13i)/288 - (3^(1/2)*
e*1i)/9 + (3^(1/2)*f*1i)/144) + log(x + (3^(1/2)*1i)/2 - 1/2)*(f/8 - (9*d)/
32 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144) + log(x +
 (3^(1/2)*1i)/2 + 1/2)*((9*d)/32 - f/8 + (3^(1/2)*d*13i)/288 - (3^(1/2)*e*1
i)/9 + (3^(1/2)*f*1i)/144)
```

3.49 $\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3} dx$

Optimal result

Integrand size = 26, antiderivative size = 243

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx &= \frac{x(d+f-(d-2f)x^2)}{12\,(1+x^2+x^4)^2}+\frac{e-2g+(2e-g)x^2}{12\,(1+x^2+x^4)^2}\\ &\quad+\frac{(2e-g)\,(1+2x^2)}{12\,(1+x^2+x^4)}+\frac{x(2d+3f-7(d-f)x^2)}{24\,(1+x^2+x^4)}\\ &\quad-\frac{(13d+2f)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}}+\frac{(13d+2f)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}}\\ &\quad+\frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}}-\frac{1}{32}(9d-4f)\log\left(1-x+x^2\right)\\ &\quad+\frac{1}{32}(9d-4f)\log\left(1+x+x^2\right)\end{aligned}$$

```
[Out] 1/12*x*(d+f-(d-2*f)*x^2)/(x^4+x^2+1)^2+1/12*(e-2*g+(2*e-g)*x^2)/(x^4+x^2+1)
^2+1/12*(2*e-g)*(2*x^2+1)/(x^4+x^2+1)+1/24*x*(2*d+3*f-7*(d-f)*x^2)/(x^4+x^2
+1)-1/32*(9*d-4*f)*ln(x^2-x+1)+1/32*(9*d-4*f)*ln(x^2+x+1)-1/144*(13*d+2*f)*
arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/144*(13*d+2*f)*arctan(1/3*(1+2*x)*3^(
1/2))*3^(1/2)+1/9*(2*e-g)*arctan(1/3*(2*x^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.15 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.00, number of steps used = 17, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.385, Rules used = {1687, 1192, 1183, 648, 632, 210, 642, 1261, 652, 628}

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(13d+2f)}{48\sqrt{3}} + \frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(13d+2f)}{48\sqrt{3}} + \frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{3\sqrt{3}} - \frac{1}{32}(9d-4f)\log\left(x^2-x+1\right) + \frac{1}{32}(9d-4f)\log\left(x^2+x+1\right) + \frac{x(-7x^2(d-f)+2d+3f)}{24\left(x^4+x^2+1\right)} + \frac{x(-(x^2(d-2f))+d+f)}{12\left(x^4+x^2+1\right)^2} + \frac{(2x^2+1)(2e-g)}{12\left(x^4+x^2+1\right)} + \frac{x^2(2e-g)+e-2g}{12\left(x^4+x^2+1\right)^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4)^3,x]
```

```
[Out] (x*(d + f - (d - 2*f)*x^2))/(12*(1 + x^2 + x^4)^2) + (e - 2*g + (2*e - g)*x
^2)/(12*(1 + x^2 + x^4)^2) + ((2*e - g)*(1 + 2*x^2))/(12*(1 + x^2 + x^4)) +
 (x*(2*d + 3*f - 7*(d - f)*x^2))/(24*(1 + x^2 + x^4)) - ((13*d + 2*f)*ArcTa
n[(1 - 2*x)/Sqrt[3]])/(48*Sqrt[3]) + ((13*d + 2*f)*ArcTan[(1 + 2*x)/Sqrt[3]
])/(48*Sqrt[3]) + ((2*e - g)*ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((9
*d - 4*f)*Log[1 - x + x^2])/32 + ((9*d - 4*f)*Log[1 + x + x^2])/32
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
```

```
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+fx^2}{(1+x^2+x^4)^3}\,dx + \int \frac{x(e+gx^2)}{(1+x^2+x^4)^3}\,dx \\
&= \frac{x(d+f-(d-2f)x^2)}{12(1+x^2+x^4)^2} + \frac{1}{12}\int \frac{11d-f-5(d-2f)x^2}{(1+x^2+x^4)^2}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(1+x+x^2)^3}\,dx, x, x^2\right) \\
&= \frac{x(d+f-(d-2f)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+(2e-g)x^2}{12(1+x^2+x^4)^2} + \frac{x(2d+3f-7(d-f)x^2)}{24(1+x^2+x^4)} \\
&\quad + \frac{1}{72}\int \frac{15(4d-f)-21(d-f)x^2}{1+x^2+x^4}\,dx + \frac{1}{4}(2e-g)\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-(d-2f)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+(2e-g)x^2}{12(1+x^2+x^4)^2} + \frac{(2e-g)(1+2x^2)}{12(1+x^2+x^4)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24(1+x^2+x^4)} + \frac{1}{144}\int \frac{15(4d-f)-(21(d-f)+15(4d-f))x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{144}\int \frac{15(4d-f)+(21(d-f)+15(4d-f))x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{6}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-(d-2f)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+(2e-g)x^2}{12(1+x^2+x^4)^2} + \frac{(2e-g)(1+2x^2)}{12(1+x^2+x^4)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24(1+x^2+x^4)} + \frac{1}{32}(9d-4f)\int \frac{1+2x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{96}(13d+2f)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{96}(13d+2f)\int \frac{1}{1+x+x^2}\,dx \\
&\quad + \frac{1}{32}(-9d+4f)\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{3}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f)\log\left(1-x+x^2\right) \\
&\quad + \frac{1}{32}(9d-4f)\log\left(1+x+x^2\right) + \frac{1}{48}(-13d-2f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) \\
&\quad + \frac{1}{48}(-13d-2f)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) \\
&= \frac{x(d+f-(d-2f)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} \\
&\quad + \frac{x(2d+3f-7(d-f)x^2)}{24\left(1+x^2+x^4\right)} - \frac{(13d+2f)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} \\
&\quad + \frac{(13d+2f)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} \\
&\quad - \frac{1}{32}(9d-4f)\log\left(1-x+x^2\right) + \frac{1}{32}(9d-4f)\log\left(1+x+x^2\right)
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.44 (sec) , antiderivative size = 259, normalized size of antiderivative = 1.07

$$
\begin{aligned}
\int \frac{d+ex+fx^2+gx^3}{\left(1+x^2+x^4\right)^3}\,dx = \frac{1}{144}\Bigg(& \frac{6(2dx+3fx-7dx^3+7fx^3-2g(1+2x^2)+e(4+8x^2))}{1+x^2+x^4} \\
& + \frac{12(e+2ex^2-g(2+x^2)+x(d+f-dx^2+2fx^2))}{\left(1+x^2+x^4\right)^2} \\
& - \frac{\left(\left(-47i+7\sqrt{3}\right)d+\left(17i-7\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} \\
& - \frac{\left(\left(47i+7\sqrt{3}\right)d-\left(17i+7\sqrt{3}\right)f\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} \\
& - 16\sqrt{3}(2e-g)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\Bigg)
\end{aligned}
$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(1 + x^2 + x^4)^3,x]
```

```
[Out] ((6*(2*d*x + 3*f*x - 7*d*x^3 + 7*f*x^3 - 2*g*(1 + 2*x^2) + e*(4 + 8*x^2)))/
(1 + x^2 + x^4) + (12*(e + 2*e*x^2 - g*(2 + x^2) + x*(d + f - d*x^2 + 2*f*x
```

```
^2)))/(1 + x^2 + x^4)^2 - (((-47*I + 7*Sqrt[3])*d + (17*I - 7*Sqrt[3])*f)*A
rcTan[((-I + Sqrt[3])*x)/2])/Sqrt[(1 + I*Sqrt[3])/6] - (((47*I + 7*Sqrt[3])
*d - (17*I + 7*Sqrt[3])*f)*ArcTan[((I + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3]
)/6] - 16*Sqrt[3]*(2*e - g)*ArcTan[Sqrt[3]/(1 + 2*x^2)])/144
```

Maple [A] (verified)

Time = 0.30 (sec) , antiderivative size = 232, normalized size of antiderivative = 0.95

method	result
default	$-\frac{\left(\frac{7d}{3}-\frac{7f}{3}-\frac{4e}{3}-\frac{g}{3}\right)x^3+(-6d+4f+2g)x^2+\left(\frac{20d}{3}-\frac{13f}{3}+\frac{e}{3}-\frac{8g}{3}\right)x-4d+\frac{4f}{3}-2e+2g}{16(x^2-x+1)^2}-\frac{(27d-12f)\ln(x^2-x+1)}{96}-\frac{\left(-\frac{13d}{2}-16e-f\right.}{}$
risch	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x,method=_RETURNVERBOSE)
```

```
[Out] -1/16*((7/3*d-7/3*f-4/3*e-1/3*g)*x^3+(-6*d+4*f+2*g)*x^2+(20/3*d-13/3*f+1/3*
e-8/3*g)*x-4*d+4/3*f-2*e+2*g)/(x^2-x+1)^2-1/96*(27*d-12*f)*ln(x^2-x+1)-1/72
*(-13/2*d-16*e-f+8*g)*3^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/16*((-7/3*d+7/3
*f-4/3*e-1/3*g)*x^3+(-6*d+4*f-2*g)*x^2+(-20/3*d+13/3*f+1/3*e-8/3*g)*x-4*d+4
/3*f+2*e-2*g)/(x^2+x+1)^2+1/96*(27*d-12*f)*ln(x^2+x+1)+1/72*(13/2*d-16*e+f+
8*g)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.49 (sec) , antiderivative size = 435, normalized size of antiderivative = 1.79

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = -\frac{84\,(d-f)x^7-48\,(2\,e-g)x^6+60\,(d-2\,f)x^5-72\,(2\,e-g)x^4+84\,(d-2\,f)x^3-96\,(2\,e-g)x^2-2\,\sqrt{}}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="fricas")
```

```
[Out] -1/288*(84*(d - f)*x^7 - 48*(2*e - g)*x^6 + 60*(d - 2*f)*x^5 - 72*(2*e - g)
*x^4 + 84*(d - 2*f)*x^3 - 96*(2*e - g)*x^2 - 2*sqrt(3)*((13*d - 32*e + 2*f
+ 16*g)*x^8 + 2*(13*d - 32*e + 2*f + 16*g)*x^6 + 3*(13*d - 32*e + 2*f + 16*
g)*x^4 + 2*(13*d - 32*e + 2*f + 16*g)*x^2 + 13*d - 32*e + 2*f + 16*g)*arcta
n(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*((13*d + 32*e + 2*f - 16*g)*x^8 + 2*(1
3*d + 32*e + 2*f - 16*g)*x^6 + 3*(13*d + 32*e + 2*f - 16*g)*x^4 + 2*(13*d +
 32*e + 2*f - 16*g)*x^2 + 13*d + 32*e + 2*f - 16*g)*arctan(1/3*sqrt(3)*(2*x
 - 1)) - 12*(4*d + 5*f)*x - 9*((9*d - 4*f)*x^8 + 2*(9*d - 4*f)*x^6 + 3*(9*d
 - 4*f)*x^4 + 2*(9*d - 4*f)*x^2 + 9*d - 4*f)*log(x^2 + x + 1) + 9*((9*d - 4
```

```
*f)*x^8 + 2*(9*d - 4*f)*x^6 + 3*(9*d - 4*f)*x^4 + 2*(9*d - 4*f)*x^2 + 9*d -
 4*f)*log(x^2 - x + 1) - 72*e + 72*g)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**3,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 200, normalized size of antiderivative = 0.82

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{1}{32}(9\,d-4\,f)\log\left(x^2+x+1\right) - \frac{1}{32}(9\,d-4\,f)\log\left(x^2-x+1\right) - \frac{7\,(d-f)x^7-4\,(2\,e-g)x^6+5\,(d-2\,f)x^5-6\,(2\,e-g)x^4+7\,(d-2\,f)x^3-8\,(2\,e-g)x^2-(4\,d+5\,f)x}{24\,(x^8+2\,x^6+3\,x^4+2\,x^2+1)}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="maxima")
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/
144*sqrt(3)*(13*d + 32*e + 2*f - 16*g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/32
*(9*d - 4*f)*log(x^2 + x + 1) - 1/32*(9*d - 4*f)*log(x^2 - x + 1) - 1/24*(7
*(d - f)*x^7 - 4*(2*e - g)*x^6 + 5*(d - 2*f)*x^5 - 6*(2*e - g)*x^4 + 7*(d -
 2*f)*x^3 - 8*(2*e - g)*x^2 - (4*d + 5*f)*x - 6*e + 6*g)/(x^8 + 2*x^6 + 3*x
^4 + 2*x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 192, normalized size of antiderivative = 0.79

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right) + \frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right) + \frac{1}{32}(9\,d-4\,f)\log\left(x^2+x+1\right) - \frac{1}{32}(9\,d-4\,f)\log\left(x^2-x+1\right) - \frac{7\,dx^7-7\,fx^7-8\,ex^6+4\,gx^6+5\,dx^5-10\,fx^5-12\,ex^4+6\,gx^4+7\,dx^3-14\,fx^3-16\,ex^2+8\,gx^2-4}{24\,(x^4+x^2+1)^2}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="giac")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g)*arctan(1/3*sqrt(3)*(2*x + 1)) + 1/
144*sqrt(3)*(13*d + 32*e + 2*f - 16*g)*arctan(1/3*sqrt(3)*(2*x - 1)) + 1/32
*(9*d - 4*f)*log(x^2 + x + 1) - 1/32*(9*d - 4*f)*log(x^2 - x + 1) - 1/24*(7
*d*x^7 - 7*f*x^7 - 8*e*x^6 + 4*g*x^6 + 5*d*x^5 - 10*f*x^5 - 12*e*x^4 + 6*g*
x^4 + 7*d*x^3 - 14*f*x^3 - 16*e*x^2 + 8*g*x^2 - 4*d*x - 5*f*x - 6*e + 6*g)/
(x^4 + x^2 + 1)^2
```

Mupad [B] (verification not implemented)

Time = 8.07 (sec) , antiderivative size = 295, normalized size of antiderivative = 1.21

$$\int \frac{d+ex+fx^2+gx^3}{(1+x^2+x^4)^3}\,dx = \frac{\left(\frac{7\,f}{24}-\frac{7\,d}{24}\right)\,x^7+\left(\frac{e}{3}-\frac{g}{6}\right)\,x^6+\left(\frac{5\,f}{12}-\frac{5\,d}{24}\right)\,x^5+\left(\frac{e}{2}-\frac{g}{4}\right)\,x^4+\left(\frac{7\,f}{12}-\frac{7\,d}{24}\right)\,x^3+\left(\frac{2\,e}{3}-\frac{g}{3}\right)\,x^2+\left(\frac{d}{6}+\frac{5\,f}{24}\right)\,x+}{x^8+2\,x^6+3\,x^4+2\,x^2+1} - \ln\left(x-\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{9\,d}{32}-\frac{f}{8}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right) - \ln\left(x+\frac{1}{2}-\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right) + \ln\left(x-\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{f}{8}-\frac{9\,d}{32}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}+\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}-\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right) + \ln\left(x+\frac{1}{2}+\frac{\sqrt{3}\,1\mathrm{i}}{2}\right)\,\left(\frac{9\,d}{32}-\frac{f}{8}+\frac{\sqrt{3}\,d\,13\mathrm{i}}{288}-\frac{\sqrt{3}\,e\,1\mathrm{i}}{9}+\frac{\sqrt{3}\,f\,1\mathrm{i}}{144}+\frac{\sqrt{3}\,g\,1\mathrm{i}}{18}\right)$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(x^2 + x^4 + 1)^3,x)
```

```
[Out] (e/4 - g/4 - x^5*((5*d)/24 - (5*f)/12) - x^3*((7*d)/24 - (7*f)/12) - x^7*((
7*d)/24 - (7*f)/24) + x^2*((2*e)/3 - g/3) + x^4*(e/2 - g/4) + x^6*(e/3 - g/
6) + x*(d/6 + (5*f)/24))/(2*x^2 + 3*x^4 + 2*x^6 + x^8 + 1) - log(x - (3^(1/
2)*1i)/2 - 1/2)*((9*d)/32 - f/8 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9 +
(3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18) - log(x - (3^(1/2)*1i)/2 + 1/2)*(f/
8 - (9*d)/32 + (3^(1/2)*d*13i)/288 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144
+ (3^(1/2)*g*1i)/18) + log(x + (3^(1/2)*1i)/2 - 1/2)*(f/8 - (9*d)/32 + (3^(
1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18
) + log(x + (3^(1/2)*1i)/2 + 1/2)*((9*d)/32 - f/8 + (3^(1/2)*d*13i)/288 - (
3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144 + (3^(1/2)*g*1i)/18)
```

3.50 $\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 263

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx &= \frac{e-2g+(2e-g)x^2}{12(1+x^2+x^4)^2}+\frac{x(d+f-2h-(d-2f+h)x^2)}{12(1+x^2+x^4)^2}\\ &\quad+\frac{(2e-g)(1+2x^2)}{12(1+x^2+x^4)}\\ &\quad+\frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24(1+x^2+x^4)}\\ &\quad-\frac{(13d+2f+h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}}\\ &\quad+\frac{(13d+2f+h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}}\\ &\quad+\frac{(2e-g)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}}\\ &\quad-\frac{1}{32}(9d-4f+3h)\log\left(1-x+x^2\right)\\ &\quad+\frac{1}{32}(9d-4f+3h)\log\left(1+x+x^2\right)\end{aligned}$$

```
[Out] 1/12*(e-2*g+(2*e-g)*x^2)/(x^4+x^2+1)^2+1/12*x*(d+f-2*h-(d-2*f+h)*x^2)/(x^4+
x^2+1)^2+1/12*(2*e-g)*(2*x^2+1)/(x^4+x^2+1)+1/24*x*(2*d+3*f-h-(7*d-7*f+4*h)
*x^2)/(x^4+x^2+1)-1/32*(9*d-4*f+3*h)*ln(x^2-x+1)+1/32*(9*d-4*f+3*h)*ln(x^2+
x+1)-1/144*(13*d+2*f+h)*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/144*(13*d+2*f
+h)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/9*(2*e-g)*arctan(1/3*(2*x^2+1)*3^
(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.18 (sec) , antiderivative size = 263, normalized size of antiderivative = 1.00, number of steps used = 17, number of rules used = 11, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.355, Rules used = {1687, 1692, 1192, 1183, 648, 632, 210, 642, 1261, 652, 628}

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx = & -\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(13d+2f+h)}{48\sqrt{3}} \\ & +\frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(13d+2f+h)}{48\sqrt{3}} \\ & +\frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g)}{3\sqrt{3}} \\ & -\frac{1}{32}\log\left(x^2-x+1\right)(9d-4f+3h) \\ & +\frac{1}{32}\log\left(x^2+x+1\right)(9d-4f+3h) \\ & +\frac{x(-(x^2(7d-7f+4h))+2d+3f-h)}{24\left(x^4+x^2+1\right)} \\ & +\frac{x(-(x^2(d-2f+h))+d+f-2h)}{12\left(x^4+x^2+1\right)^2} \\ & +\frac{(2x^2+1)(2e-g)}{12\left(x^4+x^2+1\right)}+\frac{x^2(2e-g)+e-2g}{12\left(x^4+x^2+1\right)^2}\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4)^3,x]

[Out] (e - 2*g + (2*e - g)*x^2)/(12*(1 + x^2 + x^4)^2) + (x*(d + f - 2*h - (d - 2
*f + h)*x^2))/(12*(1 + x^2 + x^4)^2) + ((2*e - g)*(1 + 2*x^2))/(12*(1 + x^2
 + x^4)) + (x*(2*d + 3*f - h - (7*d - 7*f + 4*h)*x^2))/(24*(1 + x^2 + x^4))
 - ((13*d + 2*f + h)*ArcTan[(1 - 2*x)/Sqrt[3]])/(48*Sqrt[3]) + ((13*d + 2*f
 + h)*ArcTan[(1 + 2*x)/Sqrt[3]])/(48*Sqrt[3]) + ((2*e - g)*ArcTan[(1 + 2*x^
2)/Sqrt[3]])/(3*Sqrt[3]) - ((9*d - 4*f + 3*h)*Log[1 - x + x^2])/32 + ((9*d
- 4*f + 3*h)*Log[1 + x + x^2])/32
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
```

```
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
```

```
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{(1+x^2+x^4)^3}\,dx + \int \frac{d+fx^2+hx^4}{(1+x^2+x^4)^3}\,dx \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{12\,(1+x^2+x^4)^2} + \frac{1}{12}\int \frac{11d-f+2h-5(d-2f+h)x^2}{(1+x^2+x^4)^2}\,dx \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(1+x+x^2)^3}\,dx, x, x^2\right) \\
&= \frac{e-2g+(2e-g)x^2}{12\,(1+x^2+x^4)^2} + \frac{x(d+f-2h-(d-2f+h)x^2)}{12\,(1+x^2+x^4)^2} \\
&\quad + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\,(1+x^2+x^4)} \\
&\quad + \frac{1}{72}\int \frac{15(4d-f+h)-3(7d-7f+4h)x^2}{1+x^2+x^4}\,dx \\
&\quad + \frac{1}{4}(2e-g)\text{Subst}\left(\int \frac{1}{(1+x+x^2)^2}\,dx, x, x^2\right)
\end{aligned}$$

$$= \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} + \frac{1}{144}\int \frac{15(4d-f+h)-(15(4d-f+h)+3(7d-7f+4h))x}{1-x+x^2}\,dx + \frac{1}{144}\int \frac{15(4d-f+h)+(15(4d-f+h)+3(7d-7f+4h))x}{1+x+x^2}\,dx + \frac{1}{6}(2e-g)\text{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right)$$

$$= \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} + \frac{1}{3}(-2e+g)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right) + \frac{1}{32}(-9d+4f-3h)\int \frac{-1+2x}{1-x+x^2}\,dx + \frac{1}{96}(13d+2f+h)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{96}(13d+2f+h)\int \frac{1}{1+x+x^2}\,dx + \frac{1}{32}(9d-4f+3h)\int \frac{1+2x}{1+x+x^2}\,dx$$

$$= \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f+3h)\log\left(1-x+x^2\right) + \frac{1}{32}(9d-4f+3h)\log\left(1+x+x^2\right) + \frac{1}{48}(-13d-2f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) + \frac{1}{48}(-13d-2f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right)$$

$$= \frac{e-2g+(2e-g)x^2}{12\left(1+x^2+x^4\right)^2} + \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{(2e-g)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} - \frac{(13d+2f+h)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(13d+2f+h)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(2e-g)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f+3h)\log\left(1-x+x^2\right) + \frac{1}{32}(9d-4f+3h)\log\left(1+x+x^2\right)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.57 (sec) , antiderivative size = 303, normalized size of antiderivative = 1.15

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx$$
$$= \frac{1}{144}\left(-\frac{6(-4e(1+2x^2)+g(2+4x^2)+x(-2d-3f+h+7dx^2-7fx^2+4hx^2))}{1+x^2+x^4} + \frac{12(e+2ex^2-g(2+x^2)+x(d+f-dx^2+2fx^2-h(2+x^2)))}{(1+x^2+x^4)^2} - \frac{\left(\left(-47i+7\sqrt{3}\right)d+\left(17i-7\sqrt{3}\right)f+2\left(-7i+2\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}} - \frac{\left(\left(47i+7\sqrt{3}\right)d-\left(17i+7\sqrt{3}\right)f+2\left(7i+2\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}} - 16\sqrt{3}(2e-g)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(1 + x^2 + x^4)^3,x]
```

```
[Out] ((-6*(-4*e*(1 + 2*x^2) + g*(2 + 4*x^2) + x*(-2*d - 3*f + h + 7*d*x^2 - 7*f*
x^2 + 4*h*x^2)))/(1 + x^2 + x^4) + (12*(e + 2*e*x^2 - g*(2 + x^2) + x*(d +
f - d*x^2 + 2*f*x^2 - h*(2 + x^2))))/(1 + x^2 + x^4)^2 - (((-47*I + 7*Sqrt[
3])*d + (17*I - 7*Sqrt[3])*f + 2*(-7*I + 2*Sqrt[3])*h)*ArcTan[((-I + Sqrt[3
])*x)/2])/Sqrt[(1 + I*Sqrt[3])/6] - (((47*I + 7*Sqrt[3])*d - (17*I + 7*Sqrt
[3])*f + 2*(7*I + 2*Sqrt[3])*h)*ArcTan[((I + Sqrt[3])*x)/2])/Sqrt[(1 - I*Sq
rt[3])/6] - 16*Sqrt[3]*(2*e - g)*ArcTan[Sqrt[3]/(1 + 2*x^2)])/144
```

Maple [A] (verified)

Time = 0.43 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.00

method	result
default	$-\frac{\left(\frac{7d}{3}-\frac{7f}{3}+\frac{4h}{3}-\frac{4e}{3}-\frac{g}{3}\right)x^3+(-6d+4f-2h+2g)x^2+\left(\frac{20d}{3}-\frac{13f}{3}+\frac{5h}{3}+\frac{e}{3}-\frac{8g}{3}\right)x-4d+\frac{4f}{3}-2e+2g}{16(x^2-x+1)^2}-\frac{(27d-12f+9h)\ln(x^2-x+1)}{96}-$
risch	Expression too large to display

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x,method=_RETURNVERBOSE)
```

```
[Out] -1/16*((7/3*d-7/3*f+4/3*h-4/3*e-1/3*g)*x^3+(-6*d+4*f-2*h+2*g)*x^2+(20/3*d-1
3/3*f+5/3*h+1/3*e-8/3*g)*x-4*d+4/3*f-2*e+2*g)/(x^2-x+1)^2-1/96*(27*d-12*f+9
*h)*ln(x^2-x+1)-1/72*(-13/2*d-16*e-f+8*g-1/2*h)*3^(1/2)*arctan(1/3*(2*x-1)*
3^(1/2))+1/16*((-7/3*d+7/3*f-4/3*h-4/3*e-1/3*g)*x^3+(-6*d+4*f-2*h-2*g)*x^2+
(-20/3*d+13/3*f-5/3*h+1/3*e-8/3*g)*x-4*d+4/3*f+2*e-2*g)/(x^2+x+1)^2+1/96*(2
7*d-12*f+9*h)*ln(x^2+x+1)+1/72*(13/2*d-16*e+f+8*g+1/2*h)*arctan(1/3*(1+2*x)
*3^(1/2))*3^(1/2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 485 vs. $2(236) = 472$.

Time = 1.22 (sec) , antiderivative size = 485, normalized size of antiderivative = 1.84

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx = -\frac{12\,(7\,d-7\,f+4\,h)x^7-48\,(2\,e-g)x^6+60\,(d-2\,f+h)x^5-72\,(2\,e-g)x^4+84\,(d-2\,f+h)x^3-96}{}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="fricas")
```

```
[Out] -1/288*(12*(7*d - 7*f + 4*h)*x^7 - 48*(2*e - g)*x^6 + 60*(d - 2*f + h)*x^5
- 72*(2*e - g)*x^4 + 84*(d - 2*f + h)*x^3 - 96*(2*e - g)*x^2 - 2*sqrt(3)*((
13*d - 32*e + 2*f + 16*g + h)*x^8 + 2*(13*d - 32*e + 2*f + 16*g + h)*x^6 +
3*(13*d - 32*e + 2*f + 16*g + h)*x^4 + 2*(13*d - 32*e + 2*f + 16*g + h)*x^2
 + 13*d - 32*e + 2*f + 16*g + h)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*
((13*d + 32*e + 2*f - 16*g + h)*x^8 + 2*(13*d + 32*e + 2*f - 16*g + h)*x^6
+ 3*(13*d + 32*e + 2*f - 16*g + h)*x^4 + 2*(13*d + 32*e + 2*f - 16*g + h)*x
^2 + 13*d + 32*e + 2*f - 16*g + h)*arctan(1/3*sqrt(3)*(2*x - 1)) - 12*(4*d
+ 5*f - 5*h)*x - 9*((9*d - 4*f + 3*h)*x^8 + 2*(9*d - 4*f + 3*h)*x^6 + 3*(9*
d - 4*f + 3*h)*x^4 + 2*(9*d - 4*f + 3*h)*x^2 + 9*d - 4*f + 3*h)*log(x^2 + x
 + 1) + 9*((9*d - 4*f + 3*h)*x^8 + 2*(9*d - 4*f + 3*h)*x^6 + 3*(9*d - 4*f +
 3*h)*x^4 + 2*(9*d - 4*f + 3*h)*x^2 + 9*d - 4*f + 3*h)*log(x^2 - x + 1) - 7
2*e + 72*g)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 217, normalized size of antiderivative = 0.83

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx$$
$$= \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)$$
$$+\frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)$$
$$+\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2+x+1\right)-\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2-x+1\right)$$
$$-\frac{(7\,d-7\,f+4\,h)x^7-4\,(2\,e-g)x^6+5\,(d-2\,f+h)x^5-6\,(2\,e-g)x^4+7\,(d-2\,f+h)x^3-8\,(2\,e-g)x}{24\,(x^8+2\,x^6+3\,x^4+2\,x^2+1)}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="maxima")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g + h)*arctan(1/3*sqrt(3)*(2*x + 1))
+ 1/144*sqrt(3)*(13*d + 32*e + 2*f - 16*g + h)*arctan(1/3*sqrt(3)*(2*x - 1)
) + 1/32*(9*d - 4*f + 3*h)*log(x^2 + x + 1) - 1/32*(9*d - 4*f + 3*h)*log(x^
2 - x + 1) - 1/24*((7*d - 7*f + 4*h)*x^7 - 4*(2*e - g)*x^6 + 5*(d - 2*f + h
)*x^5 - 6*(2*e - g)*x^4 + 7*(d - 2*f + h)*x^3 - 8*(2*e - g)*x^2 - (4*d + 5*
f - 5*h)*x - 6*e + 6*g)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 222, normalized size of antiderivative = 0.84

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx$$
$$= \frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)$$
$$+\frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g+h)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)$$
$$+\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2+x+1\right)-\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2-x+1\right)$$
$$-\frac{7\,dx^7-7\,fx^7+4\,hx^7-8\,ex^6+4\,gx^6+5\,dx^5-10\,fx^5+5\,hx^5-12\,ex^4+6\,gx^4+7\,dx^3-14\,fx^3+7\,hx}{24\,(x^4+x^2+1)^2}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="giac")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g + h)*arctan(1/3*sqrt(3)*(2*x + 1))
+ 1/144*sqrt(3)*(13*d + 32*e + 2*f - 16*g + h)*arctan(1/3*sqrt(3)*(2*x - 1)
) + 1/32*(9*d - 4*f + 3*h)*log(x^2 + x + 1) - 1/32*(9*d - 4*f + 3*h)*log(x^
2 - x + 1) - 1/24*(7*d*x^7 - 7*f*x^7 + 4*h*x^7 - 8*e*x^6 + 4*g*x^6 + 5*d*x^
5 - 10*f*x^5 + 5*h*x^5 - 12*e*x^4 + 6*g*x^4 + 7*d*x^3 - 14*f*x^3 + 7*h*x^3
- 16*e*x^2 + 8*g*x^2 - 4*d*x - 5*f*x + 5*h*x - 6*e + 6*g)/(x^4 + x^2 + 1)^2
```

Mupad [B] (verification not implemented)

Time = 3.44 (sec) , antiderivative size = 1611, normalized size of antiderivative = 6.13

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(1+x^2+x^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(x^2 + x^4 + 1)^3,x)
```

```
[Out] (e/4 - g/4 + x^2*((2*e)/3 - g/3) + x^4*(e/2 - g/4) + x^6*(e/3 - g/6) + x*(d
/6 + (5*f)/24 - (5*h)/24) - x^7*((7*d)/24 - (7*f)/24 + h/6) - x^5*((5*d)/24
 - (5*f)/12 + (5*h)/24) - x^3*((7*d)/24 - (7*f)/12 + (7*h)/24))/(2*x^2 + 3*
x^4 + 2*x^6 + x^8 + 1) - log(960*d*g - 2763*d*f - 1920*d*e + 480*e*f + 1971
*d*h - 480*e*h - 240*f*g - 981*f*h + 240*g*h + 3^(1/2)*d^2*1620i + 3^(1/2)*
f^2*180i + 3^(1/2)*h^2*135i - 3807*d^2*x - 612*f^2*x - 378*h^2*x + 2754*d^2
 + 684*f^2 + 351*h^2 + 3^(1/2)*d*e*1088i - 3^(1/2)*d*f*1125i - 3^(1/2)*d*g*
544i - 3^(1/2)*e*f*608i + 3^(1/2)*d*h*945i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g
*304i - 3^(1/2)*f*h*315i - 3^(1/2)*g*h*208i - 672*d*e*x + 3069*d*f*x + 336*
d*g*x + 672*e*f*x - 2403*d*h*x - 384*e*h*x - 336*f*g*x + 963*f*h*x + 192*g*
h*x + 3^(1/2)*d^2*x*567i + 3^(1/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2
)*d*f*x*819i + 3^(1/2)*d*g*x*752i + 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i
 - 3^(1/2)*e*h*x*448i - 3^(1/2)*f*g*x*272i - 3^(1/2)*f*h*x*333i + 3^(1/2)*g
*h*x*224i - 3^(1/2)*d*e*x*1504i)*((9*d)/32 - f/8 + (3*h)/32 + (3^(1/2)*d*13
i)/288 + (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18 + (3^(1/
2)*h*1i)/288) - log(1920*d*e - 2763*d*f - 960*d*g - 480*e*f + 1971*d*h + 48
0*e*h + 240*f*g - 981*f*h - 240*g*h - 3^(1/2)*d^2*1620i - 3^(1/2)*f^2*180i
- 3^(1/2)*h^2*135i + 3807*d^2*x + 612*f^2*x + 378*h^2*x + 2754*d^2 + 684*f^
2 + 351*h^2 + 3^(1/2)*d*e*1088i + 3^(1/2)*d*f*1125i - 3^(1/2)*d*g*544i - 3^
(1/2)*e*f*608i - 3^(1/2)*d*h*945i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g*304i + 3
^(1/2)*f*h*315i - 3^(1/2)*g*h*208i - 672*d*e*x - 3069*d*f*x + 336*d*g*x + 6
72*e*f*x + 2403*d*h*x - 384*e*h*x - 336*f*g*x - 963*f*h*x + 192*g*h*x + 3^(
1/2)*d^2*x*567i + 3^(1/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2)*d*f*x*8
19i - 3^(1/2)*d*g*x*752i - 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i + 3^(1/2
)*e*h*x*448i + 3^(1/2)*f*g*x*272i - 3^(1/2)*f*h*x*333i - 3^(1/2)*g*h*x*224i
 + 3^(1/2)*d*e*x*1504i)*(f/8 - (9*d)/32 - (3*h)/32 + (3^(1/2)*d*13i)/288 -
(3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144 + (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/
288) + log(1920*d*e - 2763*d*f - 960*d*g - 480*e*f + 1971*d*h + 480*e*h + 2
40*f*g - 981*f*h - 240*g*h + 3^(1/2)*d^2*1620i + 3^(1/2)*f^2*180i + 3^(1/2)
```

```
*h^2*135i + 3807*d^2*x + 612*f^2*x + 378*h^2*x + 2754*d^2 + 684*f^2 + 351*h
^2 - 3^(1/2)*d*e*1088i - 3^(1/2)*d*f*1125i + 3^(1/2)*d*g*544i + 3^(1/2)*e*f
*608i + 3^(1/2)*d*h*945i - 3^(1/2)*e*h*416i - 3^(1/2)*f*g*304i - 3^(1/2)*f*
h*315i + 3^(1/2)*g*h*208i - 672*d*e*x - 3069*d*f*x + 336*d*g*x + 672*e*f*x
+ 2403*d*h*x - 384*e*h*x - 336*f*g*x - 963*f*h*x + 192*g*h*x - 3^(1/2)*d^2*
x*567i - 3^(1/2)*f^2*x*252i - 3^(1/2)*h^2*x*108i + 3^(1/2)*d*f*x*819i + 3^(
1/2)*d*g*x*752i + 3^(1/2)*e*f*x*544i - 3^(1/2)*d*h*x*513i - 3^(1/2)*e*h*x*4
48i - 3^(1/2)*f*g*x*272i + 3^(1/2)*f*h*x*333i + 3^(1/2)*g*h*x*224i - 3^(1/2
)*d*e*x*1504i)*((9*d)/32 - f/8 + (3*h)/32 + (3^(1/2)*d*13i)/288 - (3^(1/2)*
e*1i)/9 + (3^(1/2)*f*1i)/144 + (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/288) + lo
g(1920*d*e + 2763*d*f - 960*d*g - 480*e*f - 1971*d*h + 480*e*h + 240*f*g +
981*f*h - 240*g*h + 3^(1/2)*d^2*1620i + 3^(1/2)*f^2*180i + 3^(1/2)*h^2*135i
 + 3807*d^2*x + 612*f^2*x + 378*h^2*x - 2754*d^2 - 684*f^2 - 351*h^2 + 3^(1
/2)*d*e*1088i - 3^(1/2)*d*f*1125i - 3^(1/2)*d*g*544i - 3^(1/2)*e*f*608i + 3
^(1/2)*d*h*945i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g*304i - 3^(1/2)*f*h*315i -
3^(1/2)*g*h*208i + 672*d*e*x - 3069*d*f*x - 336*d*g*x - 672*e*f*x + 2403*d*
h*x + 384*e*h*x + 336*f*g*x - 963*f*h*x - 192*g*h*x + 3^(1/2)*d^2*x*567i +
3^(1/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2)*d*f*x*819i + 3^(1/2)*d*g*
x*752i + 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i - 3^(1/2)*e*h*x*448i - 3^(
1/2)*f*g*x*272i - 3^(1/2)*f*h*x*333i + 3^(1/2)*g*h*x*224i - 3^(1/2)*d*e*x*1
504i)*(f/8 - (9*d)/32 - (3*h)/32 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/9 +
 (3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/288)
```

3.51 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3} dx$

Optimal result

Integrand size = 36, antiderivative size = 269

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3} dx = \frac{x(d+f-2h-(d-2f+h)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12(1+x^2+x^4)^2} + \frac{(2e-g+i)(1+2x^2)}{12(1+x^2+x^4)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24(1+x^2+x^4)} - \frac{(13d+2f+h)\arctan\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(13d+2f+h)\arctan\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(2e-g+i)\arctan\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f+3h)\log\left(1-x+x^2\right) + \frac{1}{32}(9d-4f+3h)\log\left(1+x+x^2\right)$$

```
[Out] 1/12*x*(d+f-2*h-(d-2*f+h)*x^2)/(x^4+x^2+1)^2+1/12*(e-2*g+i+(2*e-g-i)*x^2)/(
x^4+x^2+1)^2+1/12*(2*e-g+i)*(2*x^2+1)/(x^4+x^2+1)+1/24*x*(2*d+3*f-h-(7*d-7*
f+4*h)*x^2)/(x^4+x^2+1)-1/32*(9*d-4*f+3*h)*ln(x^2-x+1)+1/32*(9*d-4*f+3*h)*l
n(x^2+x+1)-1/144*(13*d+2*f+h)*arctan(1/3*(1-2*x)*3^(1/2))*3^(1/2)+1/144*(13
```

```
*d+2*f+h)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)+1/9*(2*e-g+i)*arctan(1/3*(2*x
^2+1)*3^(1/2))*3^(1/2)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 269, normalized size of antiderivative = 1.00, number of steps used = 18, number of rules used = 12, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.333, Rules used = {1687, 1692, 1192, 1183, 648, 632, 210, 642, 1677, 1674, 12, 628}

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx = &-\frac{\arctan\left(\frac{1-2x}{\sqrt{3}}\right)(13d+2f+h)}{48\sqrt{3}}\\ &+\frac{\arctan\left(\frac{2x+1}{\sqrt{3}}\right)(13d+2f+h)}{48\sqrt{3}}\\ &+\frac{\arctan\left(\frac{2x^2+1}{\sqrt{3}}\right)(2e-g+i)}{3\sqrt{3}}\\ &-\frac{1}{32}\log\left(x^2-x+1\right)(9d-4f+3h)\\ &+\frac{1}{32}\log\left(x^2+x+1\right)(9d-4f+3h)\\ &+\frac{x(-(x^2(7d-7f+4h))+2d+3f-h)}{24\left(x^4+x^2+1\right)}\\ &+\frac{x(-(x^2(d-2f+h))+d+f-2h)}{12\left(x^4+x^2+1\right)^2}\\ &+\frac{(2x^2+1)(2e-g+i)}{12\left(x^4+x^2+1\right)}\\ &+\frac{x^2(2e-g-i)+e-2g+i}{12\left(x^4+x^2+1\right)^2}\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4)^3,x]
```

```
[Out] (x*(d + f - 2*h - (d - 2*f + h)*x^2))/(12*(1 + x^2 + x^4)^2) + (e - 2*g + i
 + (2*e - g - i)*x^2)/(12*(1 + x^2 + x^4)^2) + ((2*e - g + i)*(1 + 2*x^2))/
(12*(1 + x^2 + x^4)) + (x*(2*d + 3*f - h - (7*d - 7*f + 4*h)*x^2))/(24*(1 +
 x^2 + x^4)) - ((13*d + 2*f + h)*ArcTan[(1 - 2*x)/Sqrt[3]])/(48*Sqrt[3]) +
((13*d + 2*f + h)*ArcTan[(1 + 2*x)/Sqrt[3]])/(48*Sqrt[3]) + ((2*e - g + i)*
ArcTan[(1 + 2*x^2)/Sqrt[3]])/(3*Sqrt[3]) - ((9*d - 4*f + 3*h)*Log[1 - x + x
^2])/32 + ((9*d - 4*f + 3*h)*Log[1 + x + x^2])/32
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 210

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(-(Rt[-a, 2]*Rt[-b, 2])^(
-1))*ArcTan[Rt[-b, 2]*(x/Rt[-a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] &
& (LtQ[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1183

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[a/c, 2]}, With[{r = Rt[2*q - b/c, 2]}, Dist[1/(2*c*q*r), Int
[(d*r - (d - e*q)*x)/(q - r*x + x^2), x], x] + Dist[1/(2*c*q*r), Int[(d*r +
 (d - e*q)*x)/(q + r*x + x^2), x], x]]] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && NegQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
```

```
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{d + fx^2 + hx^4}{(1 + x^2 + x^4)^3}\,dx + \int \frac{x(e + gx^2 + ix^4)}{(1 + x^2 + x^4)^3}\,dx$$

$$
\begin{aligned}
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{1}{12}\int \frac{11d-f+2h-5(d-2f+h)x^2}{\left(1+x^2+x^4\right)^2}\,dx \\
&\quad + \frac{1}{2}\mathrm{Subst}\left(\int \frac{e+gx+ix^2}{\left(1+x+x^2\right)^3}\,dx, x, x^2\right) \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12\left(1+x^2+x^4\right)^2} \\
&\quad + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} \\
&\quad + \frac{1}{72}\int \frac{15(4d-f+h)-3(7d-7f+4h)x^2}{1+x^2+x^4}\,dx \\
&\quad + \frac{1}{12}\mathrm{Subst}\left(\int \frac{3(2e-g+i)}{\left(1+x+x^2\right)^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12\left(1+x^2+x^4\right)^2} \\
&\quad + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} \\
&\quad + \frac{1}{144}\int \frac{15(4d-f+h)-(15(4d-f+h)+3(7d-7f+4h))x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{144}\int \frac{15(4d-f+h)+(15(4d-f+h)+3(7d-7f+4h))x}{1+x+x^2}\,dx \\
&\quad + \frac{1}{4}(2e-g+i)\mathrm{Subst}\left(\int \frac{1}{\left(1+x+x^2\right)^2}\,dx, x, x^2\right) \\
&= \frac{x(d+f-2h-(d-2f+h)x^2)}{12\left(1+x^2+x^4\right)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12\left(1+x^2+x^4\right)^2} \\
&\quad + \frac{(2e-g+i)\left(1+2x^2\right)}{12\left(1+x^2+x^4\right)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24\left(1+x^2+x^4\right)} + \frac{1}{32}(-9d+4f \\
&\qquad - 3h)\int \frac{-1+2x}{1-x+x^2}\,dx \\
&\quad + \frac{1}{96}(13d+2f+h)\int \frac{1}{1-x+x^2}\,dx + \frac{1}{96}(13d+2f+h)\int \frac{1}{1+x+x^2}\,dx \\
&\quad + \frac{1}{32}(9d-4f+3h)\int \frac{1+2x}{1+x+x^2}\,dx + \frac{1}{6}(2e-g+i)\mathrm{Subst}\left(\int \frac{1}{1+x+x^2}\,dx, x, x^2\right)
\end{aligned}
$$

$$= \frac{x(d+f-2h-(d-2f+h)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12(1+x^2+x^4)^2} + \frac{(2e-g+i)(1+2x^2)}{12(1+x^2+x^4)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24(1+x^2+x^4)} - \frac{1}{32}(9d-4f+3h)\log(1-x+x^2) + \frac{1}{32}(9d-4f+3h)\log(1+x+x^2) + \frac{1}{48}(-13d-2f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, -1+2x\right) + \frac{1}{48}(-13d-2f-h)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x\right) + \frac{1}{3}(-2e+g-i)\text{Subst}\left(\int \frac{1}{-3-x^2}\,dx, x, 1+2x^2\right)$$

$$= \frac{x(d+f-2h-(d-2f+h)x^2)}{12(1+x^2+x^4)^2} + \frac{e-2g+i+(2e-g-i)x^2}{12(1+x^2+x^4)^2} + \frac{(2e-g+i)(1+2x^2)}{12(1+x^2+x^4)} + \frac{x(2d+3f-h-(7d-7f+4h)x^2)}{24(1+x^2+x^4)} - \frac{(13d+2f+h)\tan^{-1}\left(\frac{1-2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(13d+2f+h)\tan^{-1}\left(\frac{1+2x}{\sqrt{3}}\right)}{48\sqrt{3}} + \frac{(2e-g+i)\tan^{-1}\left(\frac{1+2x^2}{\sqrt{3}}\right)}{3\sqrt{3}} - \frac{1}{32}(9d-4f+3h)\log(1-x+x^2) + \frac{1}{32}(9d-4f+3h)\log(1+x+x^2)$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.58 (sec) , antiderivative size = 325, normalized size of antiderivative = 1.21

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx$$
$$=\frac{1}{144}\left(\frac{12(e+i+dx+fx-2hx+2ex^2-ix^2-dx^3+2fx^3-hx^3-g(2+x^2))}{(1+x^2+x^4)^2}\right.$$
$$+\frac{6(2i+2dx+3fx-hx+4ix^2-7dx^3+7fx^3-4hx^3-2g(1+2x^2)+e(4+8x^2))}{1+x^2+x^4}$$
$$-\frac{\left(\left(-47i+7\sqrt{3}\right)d+\left(17i-7\sqrt{3}\right)f+2\left(-7i+2\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(-i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1+i\sqrt{3}\right)}}$$
$$-\frac{\left(\left(47i+7\sqrt{3}\right)d-\left(17i+7\sqrt{3}\right)f+2\left(7i+2\sqrt{3}\right)h\right)\arctan\left(\frac{1}{2}\left(i+\sqrt{3}\right)x\right)}{\sqrt{\frac{1}{6}\left(1-i\sqrt{3}\right)}}$$
$$\left.-16\sqrt{3}(2e-g+i)\arctan\left(\frac{\sqrt{3}}{1+2x^2}\right)\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(1 + x^2 + x^4)^3,x]
```

```
[Out] ((12*(e + i + d*x + f*x - 2*h*x + 2*e*x^2 - i*x^2 - d*x^3 + 2*f*x^3 - h*x^3
 - g*(2 + x^2)))/(1 + x^2 + x^4)^2 + (6*(2*i + 2*d*x + 3*f*x - h*x + 4*i*x^
2 - 7*d*x^3 + 7*f*x^3 - 4*h*x^3 - 2*g*(1 + 2*x^2) + e*(4 + 8*x^2)))/(1 + x^
2 + x^4) - (((-47*I + 7*Sqrt[3])*d + (17*I - 7*Sqrt[3])*f + 2*(-7*I + 2*Sqr
t[3])*h)*ArcTan[((-I + Sqrt[3])*x)/2])/Sqrt[(1 + I*Sqrt[3])/6] - (((47*I +
7*Sqrt[3])*d - (17*I + 7*Sqrt[3])*f + 2*(7*I + 2*Sqrt[3])*h)*ArcTan[((I + S
qrt[3])*x)/2])/Sqrt[(1 - I*Sqrt[3])/6] - 16*Sqrt[3]*(2*e - g + i)*ArcTan[Sq
rt[3]/(1 + 2*x^2)])/144
```

Maple [A] (verified)

Time = 0.50 (sec) , antiderivative size = 292, normalized size of antiderivative = 1.09

method	result
default	$-\frac{\left(\frac{7d}{3}-\frac{7f}{3}+\frac{4h}{3}-\frac{4e}{3}-\frac{g}{3}+\frac{i}{3}\right)x^3+(-6d+4f-2h+2g-2i)x^2+\left(\frac{20d}{3}-\frac{13f}{3}+\frac{5h}{3}+\frac{e}{3}-\frac{8g}{3}+\frac{7i}{3}\right)x-4d+\frac{4f}{3}-2e+2g-\frac{4i}{3}}{16(x^2-x+1)^2}-\frac{(27d-12f+9h}{9}$
risch	Expression too large to display

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x,method=_RETURNVERBOSE)
```

```
[Out] -1/16*((7/3*d-7/3*f+4/3*h-4/3*e-1/3*g+1/3*i)*x^3+(-6*d+4*f-2*h+2*g-2*i)*x^2
+(20/3*d-13/3*f+5/3*h+1/3*e-8/3*g+7/3*i)*x-4*d+4/3*f-2*e+2*g-4/3*i)/(x^2-x+
1)^2-1/96*(27*d-12*f+9*h)*ln(x^2-x+1)-1/72*(-13/2*d-16*e-f+8*g-1/2*h-8*i)*3
```

```
^(1/2)*arctan(1/3*(2*x-1)*3^(1/2))+1/16*((-7/3*d+7/3*f-4/3*h-4/3*e-1/3*g+1/
3*i)*x^3+(-6*d+4*f-2*h-2*g+2*i)*x^2+(-20/3*d+13/3*f-5/3*h+1/3*e-8/3*g+7/3*i
)*x-4*d+4/3*f+2*e-2*g+4/3*i)/(x^2+x+1)^2+1/96*(27*d-12*f+9*h)*ln(x^2+x+1)+1
/72*(13/2*d-16*e+f+8*g+1/2*h-8*i)*arctan(1/3*(1+2*x)*3^(1/2))*3^(1/2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 521 vs. $2(242) = 484$.

Time = 5.49 (sec) , antiderivative size = 521, normalized size of antiderivative = 1.94

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx = -\frac{12\,(7\,d-7\,f+4\,h)x^7-48\,(2\,e-g+i)x^6+60\,(d-2\,f+h)x^5-72\,(2\,e-g+i)x^4+84\,(d-2\,f+h)x^3}{}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="frica
s")
```

```
[Out] -1/288*(12*(7*d - 7*f + 4*h)*x^7 - 48*(2*e - g + i)*x^6 + 60*(d - 2*f + h)*
x^5 - 72*(2*e - g + i)*x^4 + 84*(d - 2*f + h)*x^3 - 48*(4*e - 2*g + i)*x^2
- 2*sqrt(3)*((13*d - 32*e + 2*f + 16*g + h - 16*i)*x^8 + 2*(13*d - 32*e + 2
*f + 16*g + h - 16*i)*x^6 + 3*(13*d - 32*e + 2*f + 16*g + h - 16*i)*x^4 + 2
*(13*d - 32*e + 2*f + 16*g + h - 16*i)*x^2 + 13*d - 32*e + 2*f + 16*g + h -
 16*i)*arctan(1/3*sqrt(3)*(2*x + 1)) - 2*sqrt(3)*((13*d + 32*e + 2*f - 16*g
 + h + 16*i)*x^8 + 2*(13*d + 32*e + 2*f - 16*g + h + 16*i)*x^6 + 3*(13*d +
32*e + 2*f - 16*g + h + 16*i)*x^4 + 2*(13*d + 32*e + 2*f - 16*g + h + 16*i)
*x^2 + 13*d + 32*e + 2*f - 16*g + h + 16*i)*arctan(1/3*sqrt(3)*(2*x - 1)) -
 12*(4*d + 5*f - 5*h)*x - 9*((9*d - 4*f + 3*h)*x^8 + 2*(9*d - 4*f + 3*h)*x^
6 + 3*(9*d - 4*f + 3*h)*x^4 + 2*(9*d - 4*f + 3*h)*x^2 + 9*d - 4*f + 3*h)*lo
g(x^2 + x + 1) + 9*((9*d - 4*f + 3*h)*x^8 + 2*(9*d - 4*f + 3*h)*x^6 + 3*(9*
d - 4*f + 3*h)*x^4 + 2*(9*d - 4*f + 3*h)*x^2 + 9*d - 4*f + 3*h)*log(x^2 - x
 + 1) - 72*e + 72*g - 48*i)/(x^8 + 2*x^6 + 3*x^4 + 2*x^2 + 1)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4+x**2+1)**3,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 229, normalized size of antiderivative = 0.85

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx$$
$$=\frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g+h-16\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)$$
$$+\frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g+h+16\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)$$
$$+\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2+x+1\right)-\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2-x+1\right)$$
$$-\frac{(7\,d-7\,f+4\,h)x^7-4\,(2\,e-g+i)x^6+5\,(d-2\,f+h)x^5-6\,(2\,e-g+i)x^4+7\,(d-2\,f+h)x^3-4\,(}{24\,(x^8+2\,x^6+3\,x^4+2\,x^2+1)}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="maxim
a")
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g + h - 16*i)*arctan(1/3*sqrt(3)*(2*x
 + 1)) + 1/144*sqrt(3)*(13*d + 32*e + 2*f - 16*g + h + 16*i)*arctan(1/3*sqr
t(3)*(2*x - 1)) + 1/32*(9*d - 4*f + 3*h)*log(x^2 + x + 1) - 1/32*(9*d - 4*f
 + 3*h)*log(x^2 - x + 1) - 1/24*((7*d - 7*f + 4*h)*x^7 - 4*(2*e - g + i)*x^
6 + 5*(d - 2*f + h)*x^5 - 6*(2*e - g + i)*x^4 + 7*(d - 2*f + h)*x^3 - 4*(4*
e - 2*g + i)*x^2 - (4*d + 5*f - 5*h)*x - 6*e + 6*g - 4*i)/(x^8 + 2*x^6 + 3*
x^4 + 2*x^2 + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 249, normalized size of antiderivative = 0.93

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx$$
$$=\frac{1}{144}\sqrt{3}(13\,d-32\,e+2\,f+16\,g+h-16\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x+1)\right)$$
$$+\frac{1}{144}\sqrt{3}(13\,d+32\,e+2\,f-16\,g+h+16\,i)\arctan\left(\frac{1}{3}\sqrt{3}(2\,x-1)\right)$$
$$+\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2+x+1\right)-\frac{1}{32}(9\,d-4\,f+3\,h)\log\left(x^2-x+1\right)$$
$$-\frac{7\,dx^7-7\,fx^7+4\,hx^7-8\,ex^6+4\,gx^6-4\,ix^6+5\,dx^5-10\,fx^5+5\,hx^5-12\,ex^4+6\,gx^4-6\,ix^4+7\,dx^}{24\,(x^4+x^2+1)^2}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4+x^2+1)^3,x, algorithm="giac"
)
```

```
[Out] 1/144*sqrt(3)*(13*d - 32*e + 2*f + 16*g + h - 16*i)*arctan(1/3*sqrt(3)*(2*x
 + 1)) + 1/144*sqrt(3)*(13*d + 32*e + 2*f - 16*g + h + 16*i)*arctan(1/3*sqr
t(3)*(2*x - 1)) + 1/32*(9*d - 4*f + 3*h)*log(x^2 + x + 1) - 1/32*(9*d - 4*f
 + 3*h)*log(x^2 - x + 1) - 1/24*(7*d*x^7 - 7*f*x^7 + 4*h*x^7 - 8*e*x^6 + 4*
g*x^6 - 4*i*x^6 + 5*d*x^5 - 10*f*x^5 + 5*h*x^5 - 12*e*x^4 + 6*g*x^4 - 6*i*x
^4 + 7*d*x^3 - 14*f*x^3 + 7*h*x^3 - 16*e*x^2 + 8*g*x^2 - 4*i*x^2 - 4*d*x -
5*f*x + 5*h*x - 6*e + 6*g - 4*i)/(x^4 + x^2 + 1)^2
```

Mupad [B] (verification not implemented)

Time = 12.84 (sec) , antiderivative size = 1963, normalized size of antiderivative = 7.30

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(1+x^2+x^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(x^2 + x^4 + 1)^3,x)
```

```
[Out] (e/4 - g/4 + i/6 + x*(d/6 + (5*f)/24 - (5*h)/24) - x^7*((7*d)/24 - (7*f)/24
 + h/6) - x^5*((5*d)/24 - (5*f)/12 + (5*h)/24) - x^3*((7*d)/24 - (7*f)/12 +
 (7*h)/24) + x^4*(e/2 - g/4 + i/4) + x^2*((2*e)/3 - g/3 + i/6) + x^6*(e/3 -
 g/6 + i/6))/(2*x^2 + 3*x^4 + 2*x^6 + x^8 + 1) - log(960*d*g - 2763*d*f - 1
920*d*e + 480*e*f + 1971*d*h - 960*d*i - 480*e*h - 240*f*g - 981*f*h + 240*
f*i + 240*g*h - 240*h*i + 3^(1/2)*d^2*1620i + 3^(1/2)*f^2*180i + 3^(1/2)*h^
2*135i - 3807*d^2*x - 612*f^2*x - 378*h^2*x + 2754*d^2 + 684*f^2 + 351*h^2
+ 3^(1/2)*d*e*1088i - 3^(1/2)*d*f*1125i - 3^(1/2)*d*g*544i - 3^(1/2)*e*f*60
8i + 3^(1/2)*d*h*945i + 3^(1/2)*d*i*544i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g*3
04i - 3^(1/2)*f*h*315i - 3^(1/2)*f*i*304i - 3^(1/2)*g*h*208i + 3^(1/2)*h*i*
208i - 672*d*e*x + 3069*d*f*x + 336*d*g*x + 672*e*f*x - 2403*d*h*x - 336*d*
i*x - 384*e*h*x - 336*f*g*x + 963*f*h*x + 336*f*i*x + 192*g*h*x - 192*h*i*x
 + 3^(1/2)*d^2*x*567i + 3^(1/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2)*d
*f*x*819i + 3^(1/2)*d*g*x*752i + 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i -
3^(1/2)*d*i*x*752i - 3^(1/2)*e*h*x*448i - 3^(1/2)*f*g*x*272i - 3^(1/2)*f*h*
x*333i + 3^(1/2)*f*i*x*272i + 3^(1/2)*g*h*x*224i - 3^(1/2)*h*i*x*224i - 3^(
1/2)*d*e*x*1504i)*((9*d)/32 - f/8 + (3*h)/32 + (3^(1/2)*d*13i)/288 + (3^(1/
2)*e*1i)/9 + (3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/288 +
(3^(1/2)*i*1i)/18) - log(1920*d*e - 2763*d*f - 960*d*g - 480*e*f + 1971*d*h
 + 960*d*i + 480*e*h + 240*f*g - 981*f*h - 240*f*i - 240*g*h + 240*h*i - 3^
(1/2)*d^2*1620i - 3^(1/2)*f^2*180i - 3^(1/2)*h^2*135i + 3807*d^2*x + 612*f^
2*x + 378*h^2*x + 2754*d^2 + 684*f^2 + 351*h^2 + 3^(1/2)*d*e*1088i + 3^(1/2
)*d*f*1125i - 3^(1/2)*d*g*544i - 3^(1/2)*e*f*608i - 3^(1/2)*d*h*945i + 3^(1
/2)*d*i*544i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g*304i + 3^(1/2)*f*h*315i - 3^(
1/2)*f*i*304i - 3^(1/2)*g*h*208i + 3^(1/2)*h*i*208i - 672*d*e*x - 3069*d*f*
```

x + 336*d*g*x + 672*e*f*x + 2403*d*h*x - 336*d*i*x - 384*e*h*x - 336*f*g*x
- 963*f*h*x + 336*f*i*x + 192*g*h*x - 192*h*i*x + 3^(1/2)*d^2*x*567i + 3^(1
/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2)*d*f*x*819i - 3^(1/2)*d*g*x*75
2i - 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i + 3^(1/2)*d*i*x*752i + 3^(1/2)
*e*h*x*448i + 3^(1/2)*f*g*x*272i - 3^(1/2)*f*h*x*333i - 3^(1/2)*f*i*x*272i
- 3^(1/2)*g*h*x*224i + 3^(1/2)*h*i*x*224i + 3^(1/2)*d*e*x*1504i)*(f/8 - (9*
d)/32 - (3*h)/32 + (3^(1/2)*d*13i)/288 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/
144 + (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/288 - (3^(1/2)*i*1i)/18) + log(192
0*d*e - 2763*d*f - 960*d*g - 480*e*f + 1971*d*h + 960*d*i + 480*e*h + 240*f
*g - 981*f*h - 240*f*i - 240*g*h + 240*h*i + 3^(1/2)*d^2*1620i + 3^(1/2)*f^
2*180i + 3^(1/2)*h^2*135i + 3807*d^2*x + 612*f^2*x + 378*h^2*x + 2754*d^2 +
684*f^2 + 351*h^2 - 3^(1/2)*d*e*1088i - 3^(1/2)*d*f*1125i + 3^(1/2)*d*g*54
4i + 3^(1/2)*e*f*608i + 3^(1/2)*d*h*945i - 3^(1/2)*d*i*544i - 3^(1/2)*e*h*4
16i - 3^(1/2)*f*g*304i - 3^(1/2)*f*h*315i + 3^(1/2)*f*i*304i + 3^(1/2)*g*h*
208i - 3^(1/2)*h*i*208i - 672*d*e*x - 3069*d*f*x + 336*d*g*x + 672*e*f*x +
2403*d*h*x - 336*d*i*x - 384*e*h*x - 336*f*g*x - 963*f*h*x + 336*f*i*x + 19
2*g*h*x - 192*h*i*x - 3^(1/2)*d^2*x*567i - 3^(1/2)*f^2*x*252i - 3^(1/2)*h^2
*x*108i + 3^(1/2)*d*f*x*819i + 3^(1/2)*d*g*x*752i + 3^(1/2)*e*f*x*544i - 3^
(1/2)*d*h*x*513i - 3^(1/2)*d*i*x*752i - 3^(1/2)*e*h*x*448i - 3^(1/2)*f*g*x*
272i + 3^(1/2)*f*h*x*333i + 3^(1/2)*f*i*x*272i + 3^(1/2)*g*h*x*224i - 3^(1/
2)*h*i*x*224i - 3^(1/2)*d*e*x*1504i)*((9*d)/32 - f/8 + (3*h)/32 + (3^(1/2)*
d*13i)/288 - (3^(1/2)*e*1i)/9 + (3^(1/2)*f*1i)/144 + (3^(1/2)*g*1i)/18 + (3
^(1/2)*h*1i)/288 - (3^(1/2)*i*1i)/18) + log(1920*d*e + 2763*d*f - 960*d*g -
480*e*f - 1971*d*h + 960*d*i + 480*e*h + 240*f*g + 981*f*h - 240*f*i - 240
*g*h + 240*h*i + 3^(1/2)*d^2*1620i + 3^(1/2)*f^2*180i + 3^(1/2)*h^2*135i +
3807*d^2*x + 612*f^2*x + 378*h^2*x - 2754*d^2 - 684*f^2 - 351*h^2 + 3^(1/2)
*d*e*1088i - 3^(1/2)*d*f*1125i - 3^(1/2)*d*g*544i - 3^(1/2)*e*f*608i + 3^(1
/2)*d*h*945i + 3^(1/2)*d*i*544i + 3^(1/2)*e*h*416i + 3^(1/2)*f*g*304i - 3^(
1/2)*f*h*315i - 3^(1/2)*f*i*304i - 3^(1/2)*g*h*208i + 3^(1/2)*h*i*208i + 67
2*d*e*x - 3069*d*f*x - 336*d*g*x - 672*e*f*x + 2403*d*h*x + 336*d*i*x + 384
*e*h*x + 336*f*g*x - 963*f*h*x - 336*f*i*x - 192*g*h*x + 192*h*i*x + 3^(1/2
)*d^2*x*567i + 3^(1/2)*f^2*x*252i + 3^(1/2)*h^2*x*108i - 3^(1/2)*d*f*x*819i
+ 3^(1/2)*d*g*x*752i + 3^(1/2)*e*f*x*544i + 3^(1/2)*d*h*x*513i - 3^(1/2)*d
*i*x*752i - 3^(1/2)*e*h*x*448i - 3^(1/2)*f*g*x*272i - 3^(1/2)*f*h*x*333i +
3^(1/2)*f*i*x*272i + 3^(1/2)*g*h*x*224i - 3^(1/2)*h*i*x*224i - 3^(1/2)*d*e*
x*1504i)*(f/8 - (9*d)/32 - (3*h)/32 + (3^(1/2)*d*13i)/288 + (3^(1/2)*e*1i)/
9 + (3^(1/2)*f*1i)/144 - (3^(1/2)*g*1i)/18 + (3^(1/2)*h*1i)/288 + (3^(1/2)*
i*1i)/18)

3.52 $\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx$

Optimal result

Integrand size = 20, antiderivative size = 474

$$\begin{aligned}\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx &= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2}+\frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}\\ &+\frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)}+\frac{dx((b^2-7ac)(3b^2-4ac)+3bc(b^2-8ac)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}\\ &+\frac{3\sqrt{c}\left(b^4-10ab^2c+56a^2c^2+b(b^2-8ac)\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}\\ &+\frac{3\sqrt{c}\left(b^3-8abc-\frac{b^4-10ab^2c+56a^2c^2}{\sqrt{b^2-4ac}}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}-\frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}\end{aligned}$$

```
[Out] -1/4*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1/4*d*x*(b*c*x^2-2*a*c+b^
2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+3/2*c*e*(2*c*x^2+b)/(-4*a*c+b^2)^2/(c*x
^4+b*x^2+a)+1/8*d*x*((-7*a*c+b^2)*(-4*a*c+3*b^2)+3*b*c*(-8*a*c+b^2)*x^2)/a^
2/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)-6*c^2*e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(
1/2))/(-4*a*c+b^2)^(5/2)+3/16*d*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1
/2))^(1/2))*c^(1/2)*(b^4-10*a*b^2*c+56*a^2*c^2+b*(-8*a*c+b^2)*(-4*a*c+b^2)^
(1/2))/a^2/(-4*a*c+b^2)^(5/2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+3/16*d*a
rctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b^3-8*a*b*c+
(-56*a^2*c^2+10*a*b^2*c-b^4)/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2^(1/2)
/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 1.44 (sec) , antiderivative size = 474, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.500, Rules used = {1687, 12, 1106, 1192, 1180, 211, 1121, 628, 632, 212}

$$\begin{aligned}
&\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx \\
&= \frac{3\sqrt{c}d\left(56a^2c^2-10ab^2c+b(b^2-8ac)\sqrt{b^2-4ac}+b^4\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} \\
&+ \frac{3\sqrt{c}d\left(-\frac{56a^2c^2-10ab^2c+b^4}{\sqrt{b^2-4ac}}-8abc+b^3\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}} \\
&+ \frac{dx(3bcx^2(b^2-8ac)+(b^2-7ac)(3b^2-4ac))}{8a^2\left(b^2-4ac\right)^2(a+bx^2+cx^4)} \\
&- \frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}} + \frac{dx(-2ac+b^2+bcx^2)}{4a\left(b^2-4ac\right)(a+bx^2+cx^4)^2} \\
&+ \frac{3ce(b+2cx^2)}{2\left(b^2-4ac\right)^2(a+bx^2+cx^4)} - \frac{e(b+2cx^2)}{4\left(b^2-4ac\right)(a+bx^2+cx^4)^2}
\end{aligned}$$

```
[In] Int[(d + e*x)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] -1/4*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (d*x*(b^2 -
2*a*c + b*c*x^2))/(4*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (3*c*e*(b + 2
*c*x^2))/(2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (d*x*((b^2 - 7*a*c)*(3*b
^2 - 4*a*c) + 3*b*c*(b^2 - 8*a*c)*x^2))/(8*a^2*(b^2 - 4*a*c)^2*(a + b*x^2 +
 c*x^4)) + (3*Sqrt[c]*(b^4 - 10*a*b^2*c + 56*a^2*c^2 + b*(b^2 - 8*a*c)*Sqrt
[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(
8*Sqrt[2]*a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (3*Sqrt[c]
*(b^3 - 8*a*b*c - (b^4 - 10*a*b^2*c + 56*a^2*c^2)/Sqrt[b^2 - 4*a*c])*d*ArcT
an[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 -
4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (6*c^2*e*ArcTanh[(b + 2*c*x^2)/Sqrt
[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1106

```
Int[((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Simp[(-x)*(b^2
- 2*a*c + b*c*x^2)*((a + b*x^2 + c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))
), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c + 2*(p + 1)*(b
^2 - 4*a*c) + b*c*(4*p + 7)*x^2)*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; Fre
eQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
```

```
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 && !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d}{(a+bx^2+cx^4)^3}\,dx + \int \frac{ex}{(a+bx^2+cx^4)^3}\,dx \\
&= d\int \frac{1}{(a+bx^2+cx^4)^3}\,dx + e\int \frac{x}{(a+bx^2+cx^4)^3}\,dx \\
&= \frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} - \frac{d\int \frac{b^2-2ac-4(b^2-4ac)-5bcx^2}{(a+bx^2+cx^4)^2}\,dx}{4a(b^2-4ac)} \\
&\quad + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^3}\,dx, x, x^2\right) \\
&= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} \\
&\quad + \frac{dx((b^2-7ac)(3b^2-4ac)+3bc(b^2-8ac)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} \\
&\quad + \frac{d\int \frac{3(b^4-9ab^2c+28a^2c^2)+3bc(b^2-8ac)x^2}{a+bx^2+cx^4}\,dx}{8a^2(b^2-4ac)^2} - \frac{(3ce)\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{2(b^2-4ac)} \\
&= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} \\
&\quad + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{dx((b^2-7ac)(3b^2-4ac)+3bc(b^2-8ac)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} \\
&\quad - \frac{\left(3c\left(b^4-10ab^2c+56a^2c^2-b(b^2-8ac)\sqrt{b^2-4ac}\right)d\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^{5/2}} \\
&\quad + \frac{\left(3c\left(b^4-10ab^2c+56a^2c^2+b(b^2-8ac)\sqrt{b^2-4ac}\right)d\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^{5/2}} \\
&\quad + \frac{(3c^2e)\,\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{(b^2-4ac)^2}
\end{aligned}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{dx((b^2-7ac)(3b^2-4ac)+3bc(b^2-8ac)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{3\sqrt{c}\left(b^4-10ab^2c+56a^2c^2+b(b^2-8ac)\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{3\sqrt{c}\left(b^4-10ab^2c+56a^2c^2-b(b^2-8ac)\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(6c^2e)\,\mathrm{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{dx(b^2-2ac+bcx^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{dx((b^2-7ac)(3b^2-4ac)+3bc(b^2-8ac)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{3\sqrt{c}\left(b^4-10ab^2c+56a^2c^2+b(b^2-8ac)\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} - \frac{3\sqrt{c}\left(b^4-10ab^2c+56a^2c^2-b(b^2-8ac)\sqrt{b^2-4ac}\right)d\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{6c^2e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

Mathematica [A] (verified)

Time = 1.14 (sec) , antiderivative size = 488, normalized size of antiderivative = 1.03

$$\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx$$
$$=\frac{1}{16}\left(\frac{4abe+8acx(d+ex)-4bdx(b+cx^2)}{a(-b^2+4ac)(a+bx^2+cx^4)^2}+\frac{6b^3dx(b+cx^2)-2abcdx(25b+24cx^2)+8a^2c(3be+cx(7d+6ex))}{a^2(b^2-4ac)^2(a+bx^2+cx^4)}+\frac{3\sqrt{2}\sqrt{c}\left(b^4-10ab^2c+56a^2c^2+b^3\sqrt{b^2-4ac}-8abc\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}-\frac{3\sqrt{2}\sqrt{c}\left(b^4-10ab^2c+56a^2c^2-b^3\sqrt{b^2-4ac}+8abc\sqrt{b^2-4ac}\right)d\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{a^2(b^2-4ac)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}}+\frac{48c^2e\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{5/2}}-\frac{48c^2e\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{5/2}}\right)$$

```
[In] Integrate[(d + e*x)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] ((4*a*b*e + 8*a*c*x*(d + e*x) - 4*b*d*x*(b + c*x^2))/(a*(-b^2 + 4*a*c)*(a +
 b*x^2 + c*x^4)^2) + (6*b^3*d*x*(b + c*x^2) - 2*a*b*c*d*x*(25*b + 24*c*x^2)
 + 8*a^2*c*(3*b*e + c*x*(7*d + 6*e*x)))/(a^2*(b^2 - 4*a*c)^2*(a + b*x^2 + c
*x^4)) + (3*Sqrt[2]*Sqrt[c]*(b^4 - 10*a*b^2*c + 56*a^2*c^2 + b^3*Sqrt[b^2 -
 4*a*c] - 8*a*b*c*Sqrt[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b -
Sqrt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]])
- (3*Sqrt[2]*Sqrt[c]*(b^4 - 10*a*b^2*c + 56*a^2*c^2 - b^3*Sqrt[b^2 - 4*a*c]
 + 8*a*b*c*Sqrt[b^2 - 4*a*c])*d*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^
2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + (48*c
^2*e*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) - (48*c^2*e
*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.53 (sec) , antiderivative size = 499, normalized size of antiderivative = 1.05

method	result
risch	$\frac{-\frac{3b\,c^{2}d\left(8ac-b^{2}\right)x^{7}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{3c^{3}e\,x^{6}}{16a^{2}c^{2}-8a\,b^{2}c+b^{4}}+\frac{cd\left(28a^{2}c^{2}-49a\,b^{2}c+6b^{4}\right)x^{5}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{9b\,c^{2}e\,x^{4}}{2\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}-\frac{bd\left(4a^{2}c^{2}+20a\,b^{2}c-3b^{4}\right)x^{3}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{\left(5ac+b\right.}{16a^{2}c^{2}-8}}{\left(c\,x^{4}+b\,x^{2}+a\right)^{2}}$
default	$64c^{3}\left(\frac{-\frac{3\left(24a^{2}c^{2}\sqrt{-4ac+b^{2}}-10a\,b^{2}c\sqrt{-4ac+b^{2}}+b^{4}\sqrt{-4ac+b^{2}}+32a^{2}b\,c^{2}-12a\,b^{3}c+b^{5}\right)d\,x^{3}}{64a^{2}c^{3}}+\frac{3e\left(4ac-b^{2}\right)x^{2}}{8c^{2}}-\frac{d\left(-20\sqrt{-4ac+b^{2}}\,abc+5\sqrt{-4ac\right.}{64}}{\left(x^{2}+\frac{b}{2c}-\frac{\sqrt{-4ac+b^{2}}}{2c}\right)^{2}}\right.$

[In] int((e*x+d)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBOSE)

[Out] (-3/8*b*c^2*d*(8*a*c-b^2)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^7+3*c^3*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^6+1/8/a^2*c*d*(28*a^2*c^2-49*a*b^2*c+6*b^4)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^5+9/2*b*c^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^4-1/8*b*d*(4*a^2*c^2+20*a*b^2*c-3*b^4)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^3+(5*a*c+b^2)*c*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2+1/8*d*(44*a^2*c^2-37*a*b^2*c+5*b^4)/(16*a^2*c^2-8*a*b^2*c+b^4)/a*x+1/4*b*(10*a*c-b^2)*e/(16*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+3/16*sum((-b*c*d*(8*a*c-b^2)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2+16*c^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*_R+d*(28*a^2*c^2-9*a*b^2*c+b^4)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex}{\left(a+bx^2+cx^4\right)^3}\,dx=\text{Timed out}$$

[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="fricas")

[Out] Timed out

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx=\text{Timed out}$$

```
[In] integrate((e*x+d)/(c*x**4+b*x**2+a)**3,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx=\int \frac{ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="maxima")
[Out] 1/8*(24*a^2*c^3*e*x^6 + 36*a^2*b*c^2*e*x^4 + 3*(b^3*c^2 - 8*a*b*c^3)*d*x^7
+ (6*b^4*c - 49*a*b^2*c^2 + 28*a^2*c^3)*d*x^5 + (3*b^5 - 20*a*b^3*c - 4*a^2
*b*c^2)*d*x^3 + 8*(a^2*b^2*c + 5*a^3*c^2)*e*x^2 + (5*a*b^4 - 37*a^2*b^2*c +
 44*a^3*c^2)*d*x - 2*(a^2*b^3 - 10*a^3*b*c)*e)/((a^2*b^4*c^2 - 8*a^3*b^2*c^
3 + 16*a^4*c^4)*x^8 + a^4*b^4 - 8*a^5*b^2*c + 16*a^6*c^2 + 2*(a^2*b^5*c - 8
*a^3*b^3*c^2 + 16*a^4*b*c^3)*x^6 + (a^2*b^6 - 6*a^3*b^4*c + 32*a^5*c^3)*x^4
 + 2*(a^3*b^5 - 8*a^4*b^3*c + 16*a^5*b*c^2)*x^2) - 3/8*integrate(-(16*a^2*c
^2*e*x + (b^3*c - 8*a*b*c^2)*d*x^2 + (b^4 - 9*a*b^2*c + 28*a^2*c^2)*d)/(c*x
^4 + b*x^2 + a), x)/(a^2*b^4 - 8*a^3*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3389 vs. $2(420) = 840$.

Time = 3.11 (sec) , antiderivative size = 3389, normalized size of antiderivative = 7.15

$$\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx=\text{Too large to display}$$

```
[In] integrate((e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="giac")
[Out] 3/32*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7
*c - 2*b^8*c + 116*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*b^6*c^2 + 34*a*b^6*c^2 + 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 -
```

232*a^2*b^4*c^3 - 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 + 176*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^4*c^5 - 352*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7 + 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c - 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a*c)*b^6*c - 26*(b^2 - 4*a*c)*a*b^4*c^2 - 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2 - 4*a*c)*a^2*b^2*c^3 + 22*(b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*c^4 - 88*(b^2 - 4*a*c)*a^2*b*c^4)*d*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 3/32*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7*c + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 232*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 - 176*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896*a^4*c^5 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(b^2 - 4*a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3

```
*c^3 + 224*(b^2 - 4*a*c)*a^3*c^4 + 88*(b^2 - 4*a*c)*a^2*b*c^4)*d*arctan(2*s
qrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a
^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^
4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3
)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^
2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a
^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) - 3*(b^2*c^4
- 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*(a^2*b^5 - 8*a
^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4
*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c
^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*b^6*c - 2*b^7*
c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3 - 96*a^2*b^3*
c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2*c^4 - 64*a^3*
c^5)*c^2) + 3*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e*log(x
^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*
c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8
*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^
8 - 16*a*b^6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^
3*b^2*c^3 - 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 4
8*a^2*b^2*c^4 - 64*a^3*c^5)*c^2) + 1/8*(3*b^3*c^2*d*x^7 - 24*a*b*c^3*d*x^7
+ 24*a^2*c^3*e*x^6 + 6*b^4*c*d*x^5 - 49*a*b^2*c^2*d*x^5 + 28*a^2*c^3*d*x^5
+ 36*a^2*b*c^2*e*x^4 + 3*b^5*d*x^3 - 20*a*b^3*c*d*x^3 - 4*a^2*b*c^2*d*x^3 +
 8*a^2*b^2*c*e*x^2 + 40*a^3*c^2*e*x^2 + 5*a*b^4*d*x - 37*a^2*b^2*c*d*x + 44
*a^3*c^2*d*x - 2*a^2*b^3*e + 20*a^3*b*c*e)/((a^2*b^4 - 8*a^3*b^2*c + 16*a^4
*c^2)*(c*x^4 + b*x^2 + a)^2)
```

Mupad [B] (verification not implemented)

Time = 8.77 (sec) , antiderivative size = 4225, normalized size of antiderivative = 8.91

$$\int \frac{d+ex}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] symsum(log(root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 +
 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a
^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5
*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a
^15*c^10*z^4 + 65536*a^5*b^20*z^4 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 246487
4496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^
2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 7
54974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 - 188743680*a^
7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 23592960*a^6*b^8*c^5*e
^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 177
```

1776*a^2*b^15*c^2*d^2*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 2304*b^19*d^2*z^2 - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 46725120*a^3*b^8*c^5*d^2*e*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 13824*b^14*c^2*d^2*e*z + 34836480*a^4*b*c^8*d^2*e^2 - 435456*a*b^7*c^5*d^2*e^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 20736*b^9*c^4*d^2*e^2 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*(root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 2304*b^19*d^2*z^2 - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 46725120*a^3*b^8*c^5*d^2*e*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 13824*b^14*c^2*d^2*e*z + 34836480*a^4*b*c^8*d^2*e^2 - 435456*a*b^7*c^5*d^2*e^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 20736*b^9*c^4*d^2*e^2 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((x*(786432*a^9*c^9*e - 768*a^4*b^10*c^4*e + 15360*a^5*b^8*c^5*e - 122880*a^6*b^6*c^6*e + 491520*a^7*b^4*c^7*e - 983040*a^8*b^2*c^8*e))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - (3*(7340032*a^9*c^9*d - 256*a^2*b^14*c^2*d + 7424*a^3*b^12*c^3*d - 94208*a^4*b^10*c^4*d + 675840*a^5*b^8*c^5*d - 2949120*a^6*b^6*c^6*d + 7798784*a^7*b^4*c^7*d - 11534336*a^8*b^2*c^8*d))/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 2304*b^19*d^2*z^2 - 428544*a*b^12*c^3*d^2*e*z + 102275

4816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 46725120*a^3*b^8*c^5*d^2*e*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 13824*b^14*c^2*d^2*e*z + 34836480*a^4*b*c^8*d^2*e^2 - 435456*a*b^7*c^5*d^2*e^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 20736*b^9*c^4*d^2*e^2 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*x*(4194304*a^11*b*c^9 - 256*a^4*b^15*c^2 + 7168*a^5*b^13*c^3 - 86016*a^6*b^11*c^4 + 573440*a^7*b^9*c^5 - 2293760*a^8*b^7*c^6 + 5505024*a^9*b^5*c^7 - 7340032*a^10*b^3*c^8))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (3*(1081344*a^6*b*c^8*d*e + 1536*a^2*b^9*c^4*d*e - 29184*a^3*b^7*c^5*d*e + 227328*a^4*b^5*c^6*d*e - 811008*a^5*b^3*c^7*d*e))/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - (x*(225792*a^6*c^9*d^2 + 9*b^12*c^3*d^2 - 252*a*b^10*c^4*d^2 - 36864*a^6*b*c^8*e^2 + 3114*a^2*b^8*c^5*d^2 - 21312*a^3*b^6*c^6*d^2 + 88128*a^4*b^4*c^7*d^2 - 211968*a^5*b^2*c^8*d^2 - 2304*a^4*b^5*c^6*e^2 + 18432*a^5*b^3*c^7*e^2))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (3*(3456*a*b^5*c^6*d^3 - 189*b^7*c^5*d^3 + 56448*a^3*b*c^8*d^3 + 64512*a^4*c^8*d*e^2 - 22608*a^2*b^3*c^7*d^3 + 2304*a^2*b^4*c^6*d*e^2 - 20736*a^3*b^2*c^7*d*e^2))/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(6912*a^4*c^8*e^3 - 27*b^7*c^5*d^2*e + 486*a*b^5*c^6*d^2*e + 12096*a^3*b*c^8*d^2*e - 3672*a^2*b^3*c^7*d^2*e))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)))*root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 2304*b^19*d^2*z^2 - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 46725120*a^3*b^8*c^5*d^2*e*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 13824*b^14*c^2*d^2*e*z + 34836480*a^4*b*c^8*d^2*e^2 - 435456*a*b^7*c^5*d^2*e^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 20736*b^9*c^4*d^2*e^2 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k), k, 1, 4) + ((x^2*(5*a*c^2*e + b^2*c*e))/(b^4 + 16*a^2*c^2 - 8*a*b^2*c) - (b^3*e - 10*a*b*c*e)/(4*(b^4 + 16*a^

```
2*c^2 - 8*a*b^2*c)) + (3*c^3*e*x^6)/(b^4 + 16*a^2*c^2 - 8*a*b^2*c) + (9*b*c
^2*e*x^4)/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (d*x^3*(4*a^2*b*c^2 - 3*b^5
+ 20*a*b^3*c))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (d*x*(5*b^4 + 44*a^
2*c^2 - 37*a*b^2*c))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (d*x^5*(6*b^4*c
 + 28*a^2*c^3 - 49*a*b^2*c^2))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (3*
c*d*x^7*(b^3*c - 8*a*b*c^2))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)))/(x^4*(
2*a*c + b^2) + a^2 + c^2*x^8 + 2*a*b*x^2 + 2*b*c*x^6)
```

3.53 $\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3} \, dx$

Optimal result

Integrand size = 25, antiderivative size = 621

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3} \, dx = -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)-ab^2\left(30cd-\sqrt{b^2-4ac}f\right)+4a^2c(42}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}} - \frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

```
[Out] -1/4*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1/4*x*(b^2*d-2*a*c*d-a*b*
f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+3/2*c*e*(2*c*x^2+b)/
(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/8*x*(3*b^4*d-25*a*b^2*c*d+28*a^2*c^2*d+a*b
^3*f+8*a^2*b*c*f+c*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)*x^2)/a^2/(-4*a*c
+b^2)^2/(c*x^4+b*x^2+a)-6*c^2*e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4
*a*c+b^2)^(5/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))
*c^(1/2)*(3*b^4*d+b^3*(a*f+3*d*(-4*a*c+b^2)^(1/2))-4*a*b*c*(13*a*f+6*d*(-4*
a*c+b^2)^(1/2))-a*b^2*(30*c*d-f*(-4*a*c+b^2)^(1/2))+4*a^2*c*(42*c*d+5*f*(-4
*a*c+b^2)^(1/2)))/a^2/(-4*a*c+b^2)^(5/2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/
2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b
```

```
^3*d-24*a*b*c*d+a*b^2*f+20*a^2*c*f+(52*a^2*b*c*f-168*a^2*c^2*d-a*b^3*f+30*a
*b^2*c*d-3*b^4*d)/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c
+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 2.93 (sec) , antiderivative size = 621, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.360$, Rules used = {1687, 1192, 1180, 211, 12, 1121, 628, 632, 212}

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-52a^2bcf+168a^2c^2d+ab^3f-30ab^2cd+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-24abcd+3b^3d\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(4a^2c\left(5f\sqrt{b^2-4ac}+42cd\right)-ab^2\left(30cd-f\sqrt{b^2-4ac}\right)-4abc\left(6d\sqrt{b^2-4ac}+\ldots\right.\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{x\left(cx^2\left(20a^2cf+ab^2f-24abcd+3b^3d\right)+8a^2bcf+28a^2c^2d+ab^3f-25ab^2cd+3b^4d\right)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$-\frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}+\frac{x\left(cx^2(bd-2af)-abf-2acd+b^2d\right)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{3ce\left(b+2cx^2\right)}{2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}-\frac{e\left(b+2cx^2\right)}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

```
[In] Int[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] -1/4*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (x*(b^2*d -
2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(4*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x
^4)^2) + (3*c*e*(b + 2*c*x^2))/(2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (x
*(3*b^4*d - 25*a*b^2*c*d + 28*a^2*c^2*d + a*b^3*f + 8*a^2*b*c*f + c*(3*b^3*
d - 24*a*b*c*d + a*b^2*f + 20*a^2*c*f)*x^2))/(8*a^2*(b^2 - 4*a*c)^2*(a + b*
x^2 + c*x^4)) + (Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d + a*f) - 4*a
*b*c*(6*Sqrt[b^2 - 4*a*c]*d + 13*a*f) - a*b^2*(30*c*d - Sqrt[b^2 - 4*a*c]*f
) + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sq
rt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqr
t[b^2 - 4*a*c]]) + (Sqrt[c]*(3*b^3*d - 24*a*b*c*d + a*b^2*f + 20*a^2*c*f -
(3*b^4*d - 30*a*b^2*c*d + 168*a^2*c^2*d + a*b^3*f - 52*a^2*b*c*f)/Sqrt[b^2
- 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[
2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (6*c^2*e*ArcTanh[(b +
 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
```

```
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{ex}{(a+bx^2+cx^4)^3}\,dx + \int \frac{d+fx^2}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\,(b^2-4ac)\,(a+bx^2+cx^4)^2} - \frac{\int \frac{-3b^2d+14acd-abf-5c(bd-2af)x^2}{(a+bx^2+cx^4)^2}\,dx}{4a\,(b^2-4ac)} + e\int \frac{x}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\,(b^2-4ac)\,(a+bx^2+cx^4)^2} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\,(b^2-4ac)^2\,(a+bx^2+cx^4)} + \frac{\int \frac{3b^4d-27ab^2cd+84a^2c^2d+ab^3f-16a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2}{a+bx^2+cx^4}\,dx}{8a^2\,(b^2-4ac)^2} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^3}\,dx, x, x^2\right)$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} - \frac{(3ce)\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx,x,x^2\right)}{2(b^2-4ac)} + \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^2} + \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(3c^2e)\,\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(6c^2e)\,\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{3ce(b+2cx^2)}{2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{6c^2e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}$$

Mathematica [A] (verified)

Time = 1.93 (sec) , antiderivative size = 625, normalized size of antiderivative = 1.01

$$\int \frac{d+ex+fx^2}{\left(a+bx^2+cx^4\right)^3}\,dx = \frac{1}{16}\left(\frac{4ab(e+fx)-4bdx(b+cx^2)+8acx(d+x(e+fx))}{a\left(-b^2+4ac\right)\left(a+bx^2+cx^4\right)^2}\right.$$
$$+ \frac{6b^3dx(b+cx^2)+2abx(-25bcd+b^2f-24c^2dx^2+bcfx^2)+8a^2c(b(3e+2fx)+cx(7d+6ex+5fx^2))}{a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(-30cd+\sqrt{b^2-4ac}f\right)+4a^2c\ldots\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(-3b^4d+b^3\left(3\sqrt{b^2-4ac}d-af\right)+4abc\left(-6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(30cd+\sqrt{b^2-4ac}f\right)+4a\ldots\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$\left. + \frac{48c^2e\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{\left(b^2-4ac\right)^{5/2}} - \frac{48c^2e\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{\left(b^2-4ac\right)^{5/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] ((4*a*b*(e + f*x) - 4*b*d*x*(b + c*x^2) + 8*a*c*x*(d + x*(e + f*x)))/(a*(-b
^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (6*b^3*d*x*(b + c*x^2) + 2*a*b*x*(-25*
b*c*d + b^2*f - 24*c^2*d*x^2 + b*c*f*x^2) + 8*a^2*c*(b*(3*e + 2*f*x) + c*x*
(7*d + 6*e*x + 5*f*x^2)))/(a^2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (Sqrt
[2]*Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d + a*f) - 4*a*b*c*(6*Sqrt[
b^2 - 4*a*c]*d + 13*a*f) + a*b^2*(-30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2*c*
(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[
```

```
b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sq
rt[2]*Sqrt[c]*(-3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d - a*f) + 4*a*b*c*(-6*S
qrt[b^2 - 4*a*c]*d + 13*a*f) + a*b^2*(30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2
*c*(-42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + S
qrt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) +
 (48*c^2*e*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) - (48
*c^2*e*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.65 (sec) , antiderivative size = 607, normalized size of antiderivative = 0.98

method	result
risch	$\frac{\frac{c^{2}\left(20a^{2}cf+a\,b^{2}f-24abcd+3b^{3}d\right)x^{7}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{3c^{3}e\,x^{6}}{16a^{2}c^{2}-8a\,b^{2}c+b^{4}}+\frac{c\left(28a^{2}bcf+28a^{2}c^{2}d+2a\,b^{3}f-49a\,b^{2}cd+6d\,b^{4}\right)x^{5}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{9b\,c^{2}e\,x^{4}}{2\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{\left(36a^{3}c^{2}\right.}{}}{(c\,x^{4}+b\,x}$
default	Expression too large to display

```
[In] int((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBOSE)
```

```
[Out] (1/8*c^2*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+
b^4)*x^7+3*c^3*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^6+1/8/a^2*c*(28*a^2*b*c*f+28*
a^2*c^2*d+2*a*b^3*f-49*a*b^2*c*d+6*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^5+9/
2*b*c^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^4+1/8*(36*a^3*c^2*f+5*a^2*b^2*c*f-4*
a^2*b*c^2*d+a*b^4*f-20*a*b^3*c*d+3*b^5*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^
3+(5*a*c+b^2)*c*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2+1/8*(16*a^2*b*c*f+44*a^2*c
^2*d-a*b^3*f-37*a*b^2*c*d+5*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)/a*x+1/4*b*(10
*a*c-b^2)*e/(16*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((c*(20*a
^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2+48*c
^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*_R-(16*a^2*b*c*f-84*a^2*c^2*d-a*b^3*f+27*a*
b^2*c*d-3*b^4*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_
R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="fricas")
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(c*x**4+b*x**2+a)**3,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx = \int \frac{fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="maxima")
[Out] 1/8*(24*a^2*c^3*e*x^6 + 36*a^2*b*c^2*e*x^4 + (3*(b^3*c^2 - 8*a*b*c^3)*d + (
a*b^2*c^2 + 20*a^2*c^3)*f)*x^7 + ((6*b^4*c - 49*a*b^2*c^2 + 28*a^2*c^3)*d +
 2*(a*b^3*c + 14*a^2*b*c^2)*f)*x^5 + 8*(a^2*b^2*c + 5*a^3*c^2)*e*x^2 + ((3*
b^5 - 20*a*b^3*c - 4*a^2*b*c^2)*d + (a*b^4 + 5*a^2*b^2*c + 36*a^3*c^2)*f)*x
^3 - 2*(a^2*b^3 - 10*a^3*b*c)*e + ((5*a*b^4 - 37*a^2*b^2*c + 44*a^3*c^2)*d
- (a^2*b^3 - 16*a^3*b*c)*f)*x)/((a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)*
x^8 + a^4*b^4 - 8*a^5*b^2*c + 16*a^6*c^2 + 2*(a^2*b^5*c - 8*a^3*b^3*c^2 + 1
6*a^4*b*c^3)*x^6 + (a^2*b^6 - 6*a^3*b^4*c + 32*a^5*c^3)*x^4 + 2*(a^3*b^5 -
8*a^4*b^3*c + 16*a^5*b*c^2)*x^2) + 1/8*integrate((48*a^2*c^2*e*x + (3*(b^3*
c - 8*a*b*c^2)*d + (a*b^2*c + 20*a^2*c^2)*f)*x^2 + 3*(b^4 - 9*a*b^2*c + 28*
a^2*c^2)*d + (a*b^3 - 16*a^2*b*c)*f)/(c*x^4 + b*x^2 + a), x)/(a^2*b^4 - 8*a
^3*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5284 vs. $2(561) = 1122$.

Time = 2.44 (sec) , antiderivative size = 5284, normalized size of antiderivative = 8.51

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="giac")
```

```
[Out] -3*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*(a
^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*
b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^
2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*b^
6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3 -
 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2*c
^4 - 64*a^3*c^5)*c^2) + 3*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*
a*c)*e*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5
- 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a
^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^
4*c^3))/((b^8 - 16*a*b^6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*
c^2 - 256*a^3*b^2*c^3 - 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a
^3*b*c^4 + 48*a^2*b^2*c^4 - 64*a^3*c^5)*c^2) + 1/32*(3*(sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6
*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7*c - 2*b^8*c + 116*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c^2
 + 34*a*b^6*c^2 + 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 13*s
qrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 - 232*a^2*b^4*c^3 - 30*a*b
^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 224*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 + 176*a^2*b^3*c^4 - 112*sqrt(2)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^4*c^5 - 352*a^3*b*c^5 - sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7 + 15*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c - 88*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 22*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a*c)*b^6*c - 26*(b^2 - 4*a*c)
*a*b^4*c^2 - 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2 - 4*a*c)*a^2*b^2*c^3 + 22*(
```

b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*c^4 - 88*(b^2 - 4*a*c)*a^2*b
*c^4)*d + (sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a*b^6*c - 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3
*b^3*c^2 + 40*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 48*a^2*b^5*c^2 + 2*a*b^6*c^2 -
256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^2*b^3*c^3 - 288*a^3*b^3*c^3 - 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 512*a^4*b*c^4 + 64*a^3*b^2*c^4 + 320*a
^4*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 +
22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 32*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - 36*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 160*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 80*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 18*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 40*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 2*(b^2 - 4*a*c)*a*b^5
c - 40(b^2 - 4*a*c)*a^2*b^3*c^2 - 2*(b^2 - 4*a*c)*a*b^4*c^2 + 128*(b^2 -
4*a*c)*a^3*b*c^3 + 36*(b^2 - 4*a*c)*a^2*b^2*c^3 + 80*(b^2 - 4*a*c)*a^3*c^4)
*f)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt(
(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^
5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^
2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 +
24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b
^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c))
+ 1/32*(3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*b^7*c + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^
2 + 26*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 + 2*b^7*c^2 - 368*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4
*c^3 + 232*a^2*b^4*c^3 - 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 +
64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 +
176*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896
*a^4*c^5 - 352*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*b^7 + 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6
*c - 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c
^2 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 176*

```
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 88*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 11*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 44*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 -
4*a*c)*b^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 - 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(
b^2 - 4*a*c)*a^2*b^2*c^3 + 22*(b^2 - 4*a*c)*a*b^3*c^3 + 224*(b^2 - 4*a*c)*a
^3*c^4 - 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqr
t(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c + 2*a*b^7*c + 144*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 -
 48*a^2*b^5*c^2 + 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^4*b*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 288*a^3*b^3*c^3 - 44*a
^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 512*a^4
*b*c^4 + 64*a^3*b^2*c^4 + 320*a^4*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a*b^6 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a*b^5*c - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^3*b^2*c^2 - 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^2*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a*b^4*c^2 - 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^4*c^3 - 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^3*b*c^3 + 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^2*b^2*c^3 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^3*c^4 - 2*(b^2 - 4*a*c)*a*b^5*c + 40*(b^2 - 4*a*c)*a^2*b^3*c^2 - 2*(b^2 -
 4*a*c)*a*b^4*c^2 - 128*(b^2 - 4*a*c)*a^3*b*c^3 + 36*(b^2 - 4*a*c)*a^2*b^2*
c^3 + 80*(b^2 - 4*a*c)*a^3*c^4)*f)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a
^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4
*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c
^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c -
 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*
c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^
5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/8*(3*b^3*c^2*d*x^7 - 24*a*b*c^3*d*x^7 +
 a*b^2*c^2*f*x^7 + 20*a^2*c^3*f*x^7 + 24*a^2*c^3*e*x^6 + 6*b^4*c*d*x^5 - 49
*a*b^2*c^2*d*x^5 + 28*a^2*c^3*d*x^5 + 2*a*b^3*c*f*x^5 + 28*a^2*b*c^2*f*x^5
+ 36*a^2*b*c^2*e*x^4 + 3*b^5*d*x^3 - 20*a*b^3*c*d*x^3 - 4*a^2*b*c^2*d*x^3 +
 a*b^4*f*x^3 + 5*a^2*b^2*c*f*x^3 + 36*a^3*c^2*f*x^3 + 8*a^2*b^2*c*e*x^2 + 4
0*a^3*c^2*e*x^2 + 5*a*b^4*d*x - 37*a^2*b^2*c*d*x + 44*a^3*c^2*d*x - a^2*b^3
*f*x + 16*a^3*b*c*f*x - 2*a^2*b^3*e + 20*a^3*b*c*e)/((a^2*b^4 - 8*a^3*b^2*c
 + 16*a^4*c^2)*(c*x^4 + b*x^2 + a)^2)
```

Mupad [B] (verification not implemented)

Time = 9.38 (sec) , antiderivative size = 8689, normalized size of antiderivative = 13.99

$$\int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] ((x^2*(5*a*c^2*e + b^2*c*e))/(b^4 + 16*a^2*c^2 - 8*a*b^2*c) - (b^3*e - 10*a
*b*c*e)/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^5*(28*a^2*c^3*d + 6*b^4*c*d
 + 2*a*b^3*c*f - 49*a*b^2*c^2*d + 28*a^2*b*c^2*f))/(8*a^2*(b^4 + 16*a^2*c^2
 - 8*a*b^2*c)) + (x*(5*b^4*d + 44*a^2*c^2*d - a*b^3*f - 37*a*b^2*c*d + 16*a
^2*b*c*f))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (3*c^3*e*x^6)/(b^4 + 16*a
^2*c^2 - 8*a*b^2*c) + (x^3*(3*b^5*d + 36*a^3*c^2*f + a*b^4*f - 20*a*b^3*c*d
 - 4*a^2*b*c^2*d + 5*a^2*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) +
 (9*b*c^2*e*x^4)/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^7*(20*a^2*c^2*f
+ 3*b^3*c*d - 24*a*b*c^2*d + a*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2
*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2*a*b*x^2 + 2*b*c*x^6) + symsum(
log(root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 471859
20*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4
*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 +
3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^1
0*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*
c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2
 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680
*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8
*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330
240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b
*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*
z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^
5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f
^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 1
1206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^
8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2
- 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^
2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 25
6*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216
*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6
*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350
208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c
^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z
- 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^
3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2
*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b
```

^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13
824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*
f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3
*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8
068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*
d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*
c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400
*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376
*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f
^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*
b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^
2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^
2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4
 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^
4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160
000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*(root(5637
1445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c
^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 12
8849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^
9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536
*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 +
 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*
a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680*a^4*b^12*c^3
*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 17
61607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c
^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2
- 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a
*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2
*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 1887
43680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 11206656*a^7*b
^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2
 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^
3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^
2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^
2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d
*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6*d*e*f*z - 32
440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*
c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z -
642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 50724864*a^
7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^3*b^8*c^5*d^2
*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^
3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e
*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*
d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a
^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*

f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((768*a^2*b^14*c^2*d - 22020096*a^9*c^9*d - 22272*a^3*b^12*c^3*d + 282624*a^4*b^10*c^4*d - 2027520*a^5*b^8*c^5*d + 8847360*a^6*b^6*c^6*d - 23396352*a^7*b^4*c^7*d + 34603008*a^8*b^2*c^8*d + 256*a^3*b^13*c^2*f - 9216*a^4*b^11*c^3*f + 122880*a^5*b^9*c^4*f - 819200*a^6*b^7*c^5*f + 2949120*a^7*b^5*c^6*f - 5505024*a^8*b^3*c^7*f + 4194304*a^9*b*c^8*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(786432*a^9*c^9*e - 768*a^4*b^10*c^4*e + 15360*a^5*b^8*c^5*e - 122880*a^6*b^6*c^6*e + 491520*a^7*b^4*c^7*e - 983040*a^8*b^2*c^8*e))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z

+ 223395840*a^4*b^6*c^6*d^2*e*z - 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a
^6*b^4*c^6*e*f^2*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f
^2*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*
b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z
 + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^
2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^
2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2
+ 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*
c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a
^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15
482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f +
 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e
^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^
3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*
e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^
8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4
*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*
c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*
d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 357
21*b^8*c^5*d^4, z, k)*x*(4194304*a^11*b*c^9 - 256*a^4*b^15*c^2 + 7168*a^5*b
^13*c^3 - 86016*a^6*b^11*c^4 + 573440*a^7*b^9*c^5 - 2293760*a^8*b^7*c^6 + 5
505024*a^9*b^5*c^7 - 7340032*a^10*b^3*c^8))/(32*(a^4*b^12 + 4096*a^10*c^6 -
 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 61
44*a^9*b^2*c^5))) + (3244032*a^6*b*c^8*d*e - 983040*a^7*c^8*e*f + 4608*a^2*
b^9*c^4*d*e - 87552*a^3*b^7*c^5*d*e + 681984*a^4*b^5*c^6*d*e - 2433024*a^5*
b^3*c^7*d*e + 1536*a^3*b^8*c^4*e*f - 39936*a^4*b^6*c^5*e*f + 184320*a^5*b^4
*c^6*e*f + 49152*a^6*b^2*c^7*e*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b
^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^
2*c^5)) - (x*(225792*a^6*c^9*d^2 + 9*b^12*c^3*d^2 - 12800*a^7*c^8*f^2 - 252
*a*b^10*c^4*d^2 - 36864*a^6*b*c^8*e^2 + 3114*a^2*b^8*c^5*d^2 - 21312*a^3*b^
6*c^6*d^2 + 88128*a^4*b^4*c^7*d^2 - 211968*a^5*b^2*c^8*d^2 - 2304*a^4*b^5*c
^6*e^2 + 18432*a^5*b^3*c^7*e^2 + a^2*b^10*c^3*f^2 - 42*a^3*b^8*c^4*f^2 + 17
60*a^4*b^6*c^5*f^2 - 13120*a^5*b^4*c^6*f^2 + 29952*a^6*b^2*c^7*f^2 + 6*a*b^
11*c^3*d*f - 109056*a^6*b*c^8*d*f - 210*a^2*b^9*c^4*d*f + 2496*a^3*b^7*c^5*
d*f - 18240*a^4*b^5*c^6*d*f + 72192*a^5*b^3*c^7*d*f))/(32*(a^4*b^12 + 4096*
a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^
4*c^4 - 6144*a^9*b^2*c^5))) - (567*b^7*c^5*d^3 + 8000*a^5*c^7*f^3 - 10368*a
*b^5*c^6*d^3 - 169344*a^3*b*c^8*d^3 - 193536*a^4*c^8*d*e^2 + 141120*a^4*c^8
*d^2*f - 315*b^8*c^4*d^2*f + 67824*a^2*b^3*c^7*d^3 - 35*a^2*b^6*c^4*f^3 - 8
4*a^3*b^4*c^5*f^3 + 12720*a^4*b^2*c^6*f^3 + 6237*a*b^6*c^5*d^2*f - 210*a*b^
7*c^4*d*f^2 - 116160*a^4*b*c^7*d*f^2 + 36864*a^4*b*c^7*e^2*f - 6912*a^2*b^4
*c^6*d*e^2 + 62208*a^3*b^2*c^7*d*e^2 - 42372*a^2*b^4*c^6*d^2*f + 1764*a^2*b
^5*c^5*d*f^2 + 96048*a^3*b^2*c^7*d^2*f + 4608*a^3*b^3*c^6*d*f^2 - 2304*a^3*
b^3*c^6*e^2*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8
*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(6912*

a^4*c^8*e^3 - 27*b^7*c^5*d^2*e - 10080*a^4*c^8*d*e*f + 486*a*b^5*c^6*d^2*e
+ 12096*a^3*b*c^8*d^2*e + 3120*a^4*b*c^7*e*f^2 - 3672*a^2*b^3*c^7*d^2*e - 3
*a^2*b^5*c^5*e*f^2 + 96*a^3*b^3*c^6*e*f^2 - 18*a*b^6*c^5*d*e*f + 450*a^2*b^
4*c^6*d*e*f - 2448*a^3*b^2*c^7*d*e*f))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a
^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^
9*b^2*c^5)))*root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4
 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320
*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c
^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736
*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760
*a^9*b^2*c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^
5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 -
 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a
^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2
+ 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617
280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5
*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730
054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*
b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^
2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 235929
60*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2
*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1
771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2
*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f
*z + 9216*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^
5*b^5*c^6*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*
f*z - 350208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*
a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6
*d^2*e*z - 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46
725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*
c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930
496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f
^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c
^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 373
10976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^
2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3
*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 4354
56*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f
- 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 +
 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*
c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 287
0784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c
^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c
^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^
6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b

```
^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*
d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k),
k, 1, 4)
```

3.54 $\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3} \, dx$

Optimal result

Integrand size = 30, antiderivative size = 646

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3} \, dx = \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} - \frac{be-2ag+(2ce-bg)x^2}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3(2ce-bg)(b+2cx^2)}{4(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)-ab^2\left(30cd-\sqrt{b^2-4ac}f\right)+4a^2c(42c\cdots}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}} - \frac{3c(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

```
[Out] 1/4*x*(b^2*d-2*a*c*d-a*b*f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+
a)^2+1/4*(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+3/4*(
-b*g+2*c*e)*(2*c*x^2+b)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/8*x*(3*b^4*d-25*a*
b^2*c*d+28*a^2*c^2*d+a*b^3*f+8*a^2*b*c*f+c*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3
*b^3*d)*x^2)/a^2/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)-3*c*(-b*g+2*c*e)*arctanh((2
*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(5/2)+1/16*arctan(x*2^(1/2)*c^(1
/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b^4*d+b^3*(a*f+3*d*(-4*a*c+b^2
)^(1/2))-4*a*b*c*(13*a*f+6*d*(-4*a*c+b^2)^(1/2))-a*b^2*(30*c*d-f*(-4*a*c+b^
2)^(1/2))+4*a^2*c*(42*c*d+5*f*(-4*a*c+b^2)^(1/2)))/a^2/(-4*a*c+b^2)^(5/2)*2
^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*
```

```
c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b^3*d-24*a*b*c*d+a*b^2*f+20*a^2*c*f+(52*a^2
*b*c*f-168*a^2*c^2*d-a*b^3*f+30*a*b^2*c*d-3*b^4*d)/(-4*a*c+b^2)^(1/2))/a^2/
(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 2.09 (sec) , antiderivative size = 646, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.300, Rules used = {1687, 1192, 1180, 211, 1261, 652, 628, 632, 212}

$$\begin{aligned}
&\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx \\
&= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-52a^2bcf+168a^2c^2d+ab^3f-30ab^2cd+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-24abcd+3b^3d\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}} \\
&+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(4a^2c\left(5f\sqrt{b^2-4ac}+42cd\right)-ab^2\left(30cd-f\sqrt{b^2-4ac}\right)-4abc\left(6d\sqrt{b^2-4ac}\right.\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} \\
&+\frac{x\left(cx^2\left(20a^2cf+ab^2f-24abcd+3b^3d\right)+8a^2bcf+28a^2c^2d+ab^3f-25ab^2cd+3b^4d\right)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} \\
&-\frac{3c(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}+\frac{x\left(cx^2(bd-2af)-abf-2acd+b^2d\right)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} \\
&+\frac{3\left(b+2cx^2\right)(2ce-bg)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}-\frac{-2ag+x^2(2ce-bg)+be}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}
\end{aligned}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] (x*(b^2*d - 2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(4*a*(b^2 - 4*a*c)*(a +
 b*x^2 + c*x^4)^2) - (b*e - 2*a*g + (2*c*e - b*g)*x^2)/(4*(b^2 - 4*a*c)*(a
+ b*x^2 + c*x^4)^2) + (3*(2*c*e - b*g)*(b + 2*c*x^2))/(4*(b^2 - 4*a*c)^2*(a
 + b*x^2 + c*x^4)) + (x*(3*b^4*d - 25*a*b^2*c*d + 28*a^2*c^2*d + a*b^3*f +
8*a^2*b*c*f + c*(3*b^3*d - 24*a*b*c*d + a*b^2*f + 20*a^2*c*f)*x^2))/(8*a^2*
(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2
- 4*a*c]*d + a*f) - 4*a*b*c*(6*Sqrt[b^2 - 4*a*c]*d + 13*a*f) - a*b^2*(30*c*
d - Sqrt[b^2 - 4*a*c]*f) + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan
[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*
a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(3*b^3*d - 24*a*b*c*d +
a*b^2*f + 20*a^2*c*f - (3*b^4*d - 30*a*b^2*c*d + 168*a^2*c^2*d + a*b^3*f -
52*a^2*b*c*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b
^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]])
- (3*c*(2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c
)^(5/2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
```

```
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2}{(a+bx^2+cx^4)^3}\,dx + \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(a+bx+cx^2)^3}\,dx, x, x^2\right) - \frac{\int \frac{-3b^2d+14acd-abf-5c(bd-2af)x^2}{(a+bx^2+cx^4)^2}\,dx}{4a(b^2-4ac)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} - \frac{be-2ag+(2ce-bg)x^2}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\int \frac{3b^4d-27ab^2cd+84a^2c^2d+ab^3f-16a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2}{a+bx^2+cx^4}\,dx}{8a^2(b^2-4ac)^2} - \frac{(3(2ce-bg))\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{4(b^2-4ac)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} - \frac{be-2ag+(2ce-bg)x^2}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{3(2ce-bg)\left(b+2cx^2\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} + \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int\frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2\left(b^2-4ac\right)^2} + \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int\frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2\left(b^2-4ac\right)^2} + \frac{(3c(2ce-bg))\text{Subst}\left(\int\frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)^2}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} - \frac{be-2ag+(2ce-bg)x^2}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{3(2ce-bg)\left(b+2cx^2\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(3c(2ce-bg))\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{\left(b^2-4ac\right)^2}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} - \frac{be-2ag+(2ce-bg)x^2}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{3(2ce-bg)\left(b+2cx^2\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{3c(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}$$

Mathematica [A] (verified)

Time = 2.36 (sec) , antiderivative size = 661, normalized size of antiderivative = 1.02

$$\int \frac{d+ex+fx^2+gx^3}{\left(a+bx^2+cx^4\right)^3}\,dx$$
$$= \frac{1}{16}\left(\frac{-8a^2g-4bdx(b+cx^2)+8acx(d+x(e+fx))+4ab(e+x(f-gx))}{a\left(-b^2+4ac\right)\left(a+bx^2+cx^4\right)^2}\right.$$
$$+ \frac{2(3b^3dx(b+cx^2)+abx(-25bcd+b^2f-24c^2dx^2+bcfx^2)+a^2(-6b^2g+4c^2x(7d+6ex+5fx^2)+4bc}{a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(-30cd+\sqrt{b^2-4ac}f\right)+4a^2\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(-3b^4d+b^3\left(3\sqrt{b^2-4ac}d-af\right)+4abc\left(-6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(30cd+\sqrt{b^2-4ac}f\right)+4\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{24c(-2ce+bg)\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{\left(b^2-4ac\right)^{5/2}}$$
$$\left.+ \frac{24c(-2ce+bg)\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{\left(b^2-4ac\right)^{5/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] ((-8*a^2*g - 4*b*d*x*(b + c*x^2) + 8*a*c*x*(d + x*(e + f*x)) + 4*a*b*(e + x
*(f - g*x)))/(a*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (2*(3*b^3*d*x*(b +
```

```
c*x^2) + a*b*x*(-25*b*c*d + b^2*f - 24*c^2*d*x^2 + b*c*f*x^2) + a^2*(-6*b^2
*g + 4*c^2*x*(7*d + 6*e*x + 5*f*x^2) + 4*b*c*(3*e + 2*f*x - 3*g*x^2))))/(a^
2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(3*b^4*d + b^3*(3
*Sqrt[b^2 - 4*a*c]*d + a*f) - 4*a*b*c*(6*Sqrt[b^2 - 4*a*c]*d + 13*a*f) + a*
b^2*(-30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]
*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4
*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[2]*Sqrt[c]*(-3*b^4*d + b^3
*(3*Sqrt[b^2 - 4*a*c]*d - a*f) + 4*a*b*c*(-6*Sqrt[b^2 - 4*a*c]*d + 13*a*f)
+ a*b^2*(30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2*c*(-42*c*d + 5*Sqrt[b^2 - 4*
a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a^2*(b^2
 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (24*c*(-2*c*e + b*g)*Log[-b
+ Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) + (24*c*(-2*c*e + b*g)*
Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.58 (sec) , antiderivative size = 644, normalized size of antiderivative = 1.00

method	result
risch	$\frac{\frac{c^{2}\left(20 a^{2} c f+a\,b^{2} f-24 a b c d+3 b^{3} d\right) x^{7}}{8 a^{2}\left(16 a^{2} c^{2}-8 a\,b^{2} c+b^{4}\right)}-\frac{3 \left(b g-2 e c\right) c^{2} x^{6}}{2 \left(16 a^{2} c^{2}-8 a\,b^{2} c+b^{4}\right)}+\frac{c \left(28 a^{2} b c f+28 a^{2} c^{2} d+2 a\,b^{3} f-49 a\,b^{2} c d+6 d\,b^{4}\right) x^{5}}{8 a^{2}\left(16 a^{2} c^{2}-8 a\,b^{2} c+b^{4}\right)}-\frac{9 b \left(b g-2 e c\right) c\,x^{4}}{4 \left(16 a^{2} c^{2}-8 a\,b^{2} c+b^{4}\right)}+\frac{\left(36 a^{3}\right.}{}}{\left(c\,x^{4}\right.}$
default	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBOSE)
```

```
[Out] (1/8*c^2*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+
b^4)*x^7-3/2*(b*g-2*c*e)*c^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^6+1/8/a^2*c*(28*a
^2*b*c*f+28*a^2*c^2*d+2*a*b^3*f-49*a*b^2*c*d+6*b^4*d)/(16*a^2*c^2-8*a*b^2*c
+b^4)*x^5-9/4*b*(b*g-2*c*e)*c/(16*a^2*c^2-8*a*b^2*c+b^4)*x^4+1/8*(36*a^3*c^
2*f+5*a^2*b^2*c*f-4*a^2*b*c^2*d+a*b^4*f-20*a*b^3*c*d+3*b^5*d)/a^2/(16*a^2*c
^2-8*a*b^2*c+b^4)*x^3-1/2*(5*a*c+b^2)*(b*g-2*c*e)/(16*a^2*c^2-8*a*b^2*c+b^4
)*x^2+1/8*(16*a^2*b*c*f+44*a^2*c^2*d-a*b^3*f-37*a*b^2*c*d+5*b^4*d)/(16*a^2*
c^2-8*a*b^2*c+b^4)/a*x-1/4*(8*a^2*c*g+a*b^2*g-10*a*b*c*e+b^3*e)/(16*a^2*c^2
-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((c*(20*a^2*c*f+a*b^2*f-24*a*b*c
*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2-24*(b*g-2*c*e)*c/(16*a^2*c^
2-8*a*b^2*c+b^4)*_R-(16*a^2*b*c*f-84*a^2*c^2*d-a*b^3*f+27*a*b^2*c*d-3*b^4*d
)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c
+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**3,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="maxima")
```

```
[Out] 1/8*((3*(b^3*c^2 - 8*a*b*c^3)*d + (a*b^2*c^2 + 20*a^2*c^3)*f)*x^7 + 12*(2*a
^2*c^3*e - a^2*b*c^2*g)*x^6 + ((6*b^4*c - 49*a*b^2*c^2 + 28*a^2*c^3)*d + 2*
(a*b^3*c + 14*a^2*b*c^2)*f)*x^5 + 18*(2*a^2*b*c^2*e - a^2*b^2*c*g)*x^4 + ((
3*b^5 - 20*a*b^3*c - 4*a^2*b*c^2)*d + (a*b^4 + 5*a^2*b^2*c + 36*a^3*c^2)*f)
*x^3 + 4*(2*(a^2*b^2*c + 5*a^3*c^2)*e - (a^2*b^3 + 5*a^3*b*c)*g)*x^2 - 2*(a
^2*b^3 - 10*a^3*b*c)*e - 2*(a^3*b^2 + 8*a^4*c)*g + ((5*a*b^4 - 37*a^2*b^2*c
 + 44*a^3*c^2)*d - (a^2*b^3 - 16*a^3*b*c)*f)*x)/((a^2*b^4*c^2 - 8*a^3*b^2*c
^3 + 16*a^4*c^4)*x^8 + a^4*b^4 - 8*a^5*b^2*c + 16*a^6*c^2 + 2*(a^2*b^5*c -
8*a^3*b^3*c^2 + 16*a^4*b*c^3)*x^6 + (a^2*b^6 - 6*a^3*b^4*c + 32*a^5*c^3)*x^
4 + 2*(a^3*b^5 - 8*a^4*b^3*c + 16*a^5*b*c^2)*x^2) + 1/8*integrate(((3*(b^3*
c - 8*a*b*c^2)*d + (a*b^2*c + 20*a^2*c^2)*f)*x^2 + 3*(b^4 - 9*a*b^2*c + 28*
a^2*c^2)*d + (a*b^3 - 16*a^2*b*c)*f + 24*(2*a^2*c^2*e - a^2*b*c*g)*x)/(c*x^
4 + b*x^2 + a), x)/(a^2*b^4 - 8*a^3*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5619 vs. 2(586) = 1172.

Time = 2.55 (sec) , antiderivative size = 5619, normalized size of antiderivative = 8.70

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="giac")
```

```
[Out] 1/32*(3*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
b^7*c - 2*b^8*c + 116*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 +
 26*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^6*c^2 + 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^
3 - 232*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 - 176
*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^
4*c^5 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b^5*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c
+ 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2
+ 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 +
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a
*c)*b^6*c - 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2
 - 4*a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*
c^4 + 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*a*b^7*c + 144*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 48
*a^2*b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
4*b*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 288*a^3*b^3*c^3 + 44*a^2*
b^4*c^3 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 512*a^4*b*
c^4 - 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
```

4*a*c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 2*(b^2 - 4*a*c)*a*b^5*c - 40*(b^2 - 4*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*a*b^4*c^2 + 128*(b^2 - 4*a*c)*a^3*b*c^3 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80*(b^2 - 4*a*c)*a^3*c^4)*f)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/32*(3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7*c + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 232*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 - 176*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896*a^4*c^5 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(b^2 - 4*a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 + 224*(b^2 - 4*a*c)*a^3*c^4 + 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c + 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 - 48*a^2*b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 -

4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*
c^3 + 288*a^3*b^3*c^3 + 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^3*b*c^4 - 512*a^4*b*c^4 - 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 2*(b^2 - 4*a*c)*a*b^5*c + 40*(b^2 - 4
*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*a*b^4*c^2 - 128*(b^2 - 4*a*c)*a^3*b*c^3
 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80*(b^2 - 4*a*c)*a^3*c^4)*f)*arctan(2*sqr
t(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3
*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*
c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))
)/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2
+ a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7
*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 3/4*(2*(b^5*c
^3 - 8*a*b^3*c^4 - 2*b^4*c^4 + 16*a^2*b*c^5 + 8*a*b^2*c^5 + b^3*c^5 - 4*a*b
*c^6 + (b^4*c^3 - 6*a*b^2*c^4 - 2*b^3*c^4 + 8*a^2*c^5 + 4*a*b*c^5 + b^2*c^5
 - 2*a*c^6)*sqrt(b^2 - 4*a*c))*e - (b^6*c^2 - 8*a*b^4*c^3 - 2*b^5*c^3 + 16*
a^2*b^2*c^4 + 8*a*b^3*c^4 + b^4*c^4 - 4*a*b^2*c^5 - (b^5*c^2 - 6*a*b^3*c^3
- 2*b^4*c^3 + 8*a^2*b*c^4 + 4*a*b^2*c^4 + b^3*c^4 - 2*a*b*c^5)*sqrt(b^2 - 4
*a*c))*g)*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b
^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)
*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16
*a^4*c^3))/((a*b^8 - 16*a^2*b^6*c - 2*a*b^7*c + 96*a^3*b^4*c^2 + 24*a^2*b^5
*c^2 + a*b^6*c^2 - 256*a^4*b^2*c^3 - 96*a^3*b^3*c^3 - 12*a^2*b^4*c^3 + 256*
a^5*c^4 + 128*a^4*b*c^4 + 48*a^3*b^2*c^4 - 64*a^4*c^5)*c^2) + 3/2*(2*(b^2*c
^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e - (b^3*c^3 - 4*a*b*c^4 -
2*b^2*c^4 + b*c^5)*sqrt(b^2 - 4*a*c)*g)*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*
c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b
^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(
a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*b^6*c - 2*b^7*c + 96*
a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3 - 96*a^2*b^3*c^3 - 1
2*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2*c^4 - 64*a^3*c^5)*c^
2) + 1/8*(3*b^3*c^2*d*x^7 - 24*a*b*c^3*d*x^7 + a*b^2*c^2*f*x^7 + 20*a^2*c^3
*f*x^7 + 24*a^2*c^3*e*x^6 - 12*a^2*b*c^2*g*x^6 + 6*b^4*c*d*x^5 - 49*a*b^2*c
^2*d*x^5 + 28*a^2*c^3*d*x^5 + 2*a*b^3*c*f*x^5 + 28*a^2*b*c^2*f*x^5 + 36*a^2
*b*c^2*e*x^4 - 18*a^2*b^2*c*g*x^4 + 3*b^5*d*x^3 - 20*a*b^3*c*d*x^3 - 4*a^2*
b*c^2*d*x^3 + a*b^4*f*x^3 + 5*a^2*b^2*c*f*x^3 + 36*a^3*c^2*f*x^3 + 8*a^2*b^
2*c*e*x^2 + 40*a^3*c^2*e*x^2 - 4*a^2*b^3*g*x^2 - 20*a^3*b*c*g*x^2 + 5*a*b^4

```
*d*x - 37*a^2*b^2*c*d*x + 44*a^3*c^2*d*x - a^2*b^3*f*x + 16*a^3*b*c*f*x - 2
*a^2*b^3*e + 20*a^3*b*c*e - 2*a^3*b^2*g - 16*a^4*c*g)/((a^2*b^4 - 8*a^3*b^2
*c + 16*a^4*c^2)*(c*x^4 + b*x^2 + a)^2)
```

Mupad [B] (verification not implemented)

Time = 10.42 (sec) , antiderivative size = 13431, normalized size of antiderivative = 20.79

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] symsum(log((x*(13824*a^4*c^8*e^3 - 54*b^7*c^5*d^2*e + 27*b^8*c^4*d^2*g - 17
28*a^4*b^3*c^5*g^3 - 20160*a^4*c^8*d*e*f + 972*a*b^5*c^6*d^2*e + 24192*a^3*
b*c^8*d^2*e - 486*a*b^6*c^5*d^2*g + 6240*a^4*b*c^7*e*f^2 - 20736*a^4*b*c^7*
e^2*g - 7344*a^2*b^3*c^7*d^2*e + 3672*a^2*b^4*c^6*d^2*g - 6*a^2*b^5*c^5*e*f
^2 - 12096*a^3*b^2*c^7*d^2*g + 192*a^3*b^3*c^6*e*f^2 + 10368*a^4*b^2*c^6*e*
g^2 + 3*a^2*b^6*c^4*f^2*g - 96*a^3*b^4*c^5*f^2*g - 3120*a^4*b^2*c^6*f^2*g -
 36*a*b^6*c^5*d*e*f + 18*a*b^7*c^4*d*f*g + 10080*a^4*b*c^7*d*f*g + 900*a^2*
b^4*c^6*d*e*f - 4896*a^3*b^2*c^7*d*e*f - 450*a^2*b^5*c^5*d*f*g + 2448*a^3*b
^3*c^6*d*f*g))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*
c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - root(56371
445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^
2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128
849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9
*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*
a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 -
1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*
a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7
*d*f*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 -
23592960*a^6*b^9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 - 15175680*a^4*
b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g
*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a
^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6
*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 -
5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4
*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^
2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2 + 188743
680*a^8*b^6*c^5*g^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6
*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^
2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*z^2 + 11206656*a
^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2
*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 1986048
0*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^
```

2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^1
7*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c
^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z - 221773824*a^6*b^3*c^7*d*e*f*z + 110886
912*a^6*b^4*c^6*d*f*g*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*
c^6*d*e*f*z - 58982400*a^5*b^6*c^5*d*f*g*z + 16220160*a^4*b^8*c^4*d*f*g*z -
2396160*a^3*b^10*c^3*d*f*g*z + 175104*a^2*b^12*c^2*d*f*g*z - 32440320*a^4*
b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z
+ 346816512*a^7*b*c^8*d^2*g*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*
c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z + 1022754
816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3
*c^7*d^2*g*z + 321159168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*
z - 111697920*a^4*b^7*c^5*d^2*g*z + 25362432*a^7*b^3*c^6*f^2*g*z - 50724864
*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*c^5*f^2*g*z + 3563520*a^5*b^7*c^4*f
^2*g*z - 506880*a^4*b^9*c^3*f^2*g*z + 34560*a^3*b^11*c^2*f^2*g*z + 26542080
*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*
d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013
760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*
f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600
*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 15482880*
a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 7741440*a^4*b^3*c^6*d*e*f*g -
2903040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g + 3456*a*b^10*c^2
*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f - 3870720*a^5
*b^2*c^6*e*f^2*g - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f^2*
g + 80640*a^3*b^6*c^4*e*f^2*g - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*
c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 + 725760*a^3*b^6*c^4*d*f*g^2 + 17
418240*a^3*b^4*c^6*d^2*e*g - 96768*a^2*b^8*c^3*d*f*g^2 - 3919104*a^2*b^6*c^
5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387
072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f - 2654208*a^5*b^3*c^5*
e*g^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b
^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 17
37792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f
^3 - 435456*a*b^7*c^5*d^2*e^2 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8
*d^3*f - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 4262400*a^5*
b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 967680*a^5*b^
3*c^5*f^2*g^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 57
6*a^2*b^9*c^2*f^2*g^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*
d^2*f^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 97977
6*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*
f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2
*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e
^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f
^2 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^
4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 -
27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f -
734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416

```
*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((983040*a^7*c^8*e*f - 3244032*a^6*
b*c^8*d*e - 491520*a^7*b*c^7*f*g - 4608*a^2*b^9*c^4*d*e + 87552*a^3*b^7*c^5
*d*e - 681984*a^4*b^5*c^6*d*e + 2433024*a^5*b^3*c^7*d*e + 2304*a^2*b^10*c^3
*d*g - 43776*a^3*b^8*c^4*d*g - 1536*a^3*b^8*c^4*e*f + 340992*a^4*b^6*c^5*d*
g + 39936*a^4*b^6*c^5*e*f - 1216512*a^5*b^4*c^6*d*g - 184320*a^5*b^4*c^6*e*
f + 1622016*a^6*b^2*c^7*d*g - 49152*a^6*b^2*c^7*e*f + 768*a^3*b^9*c^3*f*g -
 19968*a^4*b^7*c^4*f*g + 92160*a^5*b^5*c^5*f*g + 24576*a^6*b^3*c^6*f*g)/(51
2*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^
6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - root(56371445760*a^11*b^8*c
^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691
840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^
6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2
621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73
728*a^2*b^16*c*d*f*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1321205760*a^9*b^
2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*a^7*b^6*c^6*d*f*z
^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 188743
680*a^7*b^7*c^5*e*g*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 23592960*a^6*b^9*
c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2
 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*
a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2
 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 396361
7280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^
5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 73
0054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9
*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g
^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 14
6165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 + 5898240*a^6*b
^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^
2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^
5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^
2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*
a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*
b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 4608
*a*b^14*c*d*f*g*z - 221773824*a^6*b^3*c^7*d*e*f*z + 110886912*a^6*b^4*c^6*d
*f*g*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 589
82400*a^5*b^6*c^5*d*f*g*z + 16220160*a^4*b^8*c^4*d*f*g*z - 2396160*a^3*b^10
*c^3*d*f*g*z + 175104*a^2*b^12*c^2*d*f*g*z - 32440320*a^4*b^7*c^5*d*e*f*z +
 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*
b*c^8*d^2*g*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 21427
2*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d
^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 32
1159168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4
*b^7*c^5*d^2*g*z + 25362432*a^7*b^3*c^6*f^2*g*z - 50724864*a^7*b^2*c^7*e*f^
2*z - 13271040*a^6*b^5*c^5*f^2*g*z + 3563520*a^5*b^7*c^4*f^2*g*z - 506880*a
^4*b^9*c^3*f^2*g*z + 34560*a^3*b^11*c^2*f^2*g*z + 26542080*a^6*b^4*c^6*e*f^
```

2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040
*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e
*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a
^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z
+ 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 15482880*a^5*b*c^7*d*e*f*g
 - 13824*a*b^9*c^3*d*e*f*g + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*
c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g + 3456*a*b^10*c^2*d*f*g^2 + 435456
*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f - 3870720*a^5*b^2*c^6*e*f^2*g
- 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f^2*g + 80640*a^3*b^6
*c^4*e*f^2*g - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 193
5360*a^4*b^4*c^5*d*f*g^2 + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^4*c^
6*d^2*e*g - 96768*a^2*b^8*c^3*d*f*g^2 - 3919104*a^2*b^6*c^5*d^2*e*g - 77414
40*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d
*e^2*f + 37310976*a^3*b^3*c^7*d^3*f - 2654208*a^5*b^3*c^5*e*g^3 + 3870720*a
^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 -
8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5
*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7
*c^5*d^2*e^2 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 15482880
*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 4262400*a^5*b*c^7*d*f^3 + 852
768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 967680*a^5*b^3*c^5*f^2*g^2 + 1
61280*a^4*b^5*c^4*f^2*g^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2
*g^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 + 8709120
*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2
*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^
7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2
- 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 5184*b^11*c^
2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^
4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^
4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2
*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6
*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35
721*b^8*c^5*d^4, z, k)*((768*a^2*b^14*c^2*d - 22020096*a^9*c^9*d - 22272*a^
3*b^12*c^3*d + 282624*a^4*b^10*c^4*d - 2027520*a^5*b^8*c^5*d + 8847360*a^6*
b^6*c^6*d - 23396352*a^7*b^4*c^7*d + 34603008*a^8*b^2*c^8*d + 256*a^3*b^13*
c^2*f - 9216*a^4*b^11*c^3*f + 122880*a^5*b^9*c^4*f - 819200*a^6*b^7*c^5*f +
 2949120*a^7*b^5*c^6*f - 5505024*a^8*b^3*c^7*f + 4194304*a^9*b*c^8*f)/(512*
(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*
c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(1572864*a^9*c^9*e - 1536*
a^4*b^10*c^4*e + 30720*a^5*b^8*c^5*e - 245760*a^6*b^6*c^6*e + 983040*a^7*b^
4*c^7*e - 1966080*a^8*b^2*c^8*e + 768*a^4*b^11*c^3*g - 15360*a^5*b^9*c^4*g
+ 122880*a^6*b^7*c^5*g - 491520*a^7*b^5*c^6*g + 983040*a^8*b^3*c^7*g - 7864
32*a^9*b*c^8*g))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^
8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (root(56
371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16
*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 -

128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*
a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 655
36*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2
 - 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 7321681
92*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*
c^7*d*f*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2
 - 23592960*a^6*b^9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 - 15175680*a
^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*
e*g*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 1408
0*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*
c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2
 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*
b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2
*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2 + 188
743680*a^8*b^6*c^5*g^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*
b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2
*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*z^2 + 1120665
6*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*
e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 1986
0480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15
*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*
b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^1
3*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z - 221773824*a^6*b^3*c^7*d*e*f*z + 110
886912*a^6*b^4*c^6*d*f*g*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b
^5*c^6*d*e*f*z - 58982400*a^5*b^6*c^5*d*f*g*z + 16220160*a^4*b^8*c^4*d*f*g*
z - 2396160*a^3*b^10*c^3*d*f*g*z + 175104*a^2*b^12*c^2*d*f*g*z - 32440320*a
^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*
f*z + 346816512*a^7*b*c^8*d^2*g*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^
13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z + 1022
754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*
b^3*c^7*d^2*g*z + 321159168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2
*e*z - 111697920*a^4*b^7*c^5*d^2*g*z + 25362432*a^7*b^3*c^6*f^2*g*z - 50724
864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*c^5*f^2*g*z + 3563520*a^5*b^7*c^
4*f^2*g*z - 506880*a^4*b^9*c^3*f^2*g*z + 34560*a^3*b^11*c^2*f^2*g*z + 26542
080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c
^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1
013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2
*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321
600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 154828
80*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 7741440*a^4*b^3*c^6*d*e*f*
g - 2903040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g + 3456*a*b^10*
c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f - 3870720*
a^5*b^2*c^6*e*f^2*g - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f
^2*g + 80640*a^3*b^6*c^4*e*f^2*g - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b
^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 + 725760*a^3*b^6*c^4*d*f*g^2 +

17418240*a^3*b^4*c^6*d^2*e*g - 96768*a^2*b^8*c^3*d*f*g^2 - 3919104*a^2*b^6
*c^5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f -
387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f - 2654208*a^5*b^3*c
^5*e*g^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*
a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 +
1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*
d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*
c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 4262400*a
^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 967680*a^5
*b^3*c^5*f^2*g^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 20160*a^3*b^7*c^3*f^2*g^2 +
576*a^2*b^9*c^2*f^2*g^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c
^7*d^2*f^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 97
9776*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e
^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*
a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^
2*e^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^
2*f^2 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6
*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^
4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*
f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308
416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*x*(8388608*a^11*b*c^9 - 512*a^4*
b^15*c^2 + 14336*a^5*b^13*c^3 - 172032*a^6*b^11*c^4 + 1146880*a^7*b^9*c^5 -
4587520*a^8*b^7*c^6 + 11010048*a^9*b^5*c^7 - 14680064*a^10*b^3*c^8))/(64*(
a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c
^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (x*(451584*a^6*c^9*d^2 + 18*b
^12*c^3*d^2 - 25600*a^7*c^8*f^2 - 504*a*b^10*c^4*d^2 - 73728*a^6*b*c^8*e^2
+ 6228*a^2*b^8*c^5*d^2 - 42624*a^3*b^6*c^6*d^2 + 176256*a^4*b^4*c^7*d^2 - 4
23936*a^5*b^2*c^8*d^2 - 4608*a^4*b^5*c^6*e^2 + 36864*a^5*b^3*c^7*e^2 + 2*a^
2*b^10*c^3*f^2 - 84*a^3*b^8*c^4*f^2 + 3520*a^4*b^6*c^5*f^2 - 26240*a^5*b^4*
c^6*f^2 + 59904*a^6*b^2*c^7*f^2 - 1152*a^4*b^7*c^4*g^2 + 9216*a^5*b^5*c^5*g
^2 - 18432*a^6*b^3*c^6*g^2 + 12*a*b^11*c^3*d*f - 218112*a^6*b*c^8*d*f - 420
*a^2*b^9*c^4*d*f + 4992*a^3*b^7*c^5*d*f - 36480*a^4*b^5*c^6*d*f + 144384*a^
5*b^3*c^7*d*f + 4608*a^4*b^6*c^5*e*g - 36864*a^5*b^4*c^6*e*g + 73728*a^6*b^
2*c^7*e*g))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2
- 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) - (567*b^7*c^5
*d^3 + 8000*a^5*c^7*f^3 - 10368*a*b^5*c^6*d^3 - 169344*a^3*b*c^8*d^3 - 1935
36*a^4*c^8*d*e^2 + 141120*a^4*c^8*d^2*f - 315*b^8*c^4*d^2*f + 67824*a^2*b^3
*c^7*d^3 - 35*a^2*b^6*c^4*f^3 - 84*a^3*b^4*c^5*f^3 + 12720*a^4*b^2*c^6*f^3
+ 6237*a*b^6*c^5*d^2*f - 210*a*b^7*c^4*d*f^2 - 116160*a^4*b*c^7*d*f^2 + 368
64*a^4*b*c^7*e^2*f - 6912*a^2*b^4*c^6*d*e^2 + 62208*a^3*b^2*c^7*d*e^2 - 423
72*a^2*b^4*c^6*d^2*f + 1764*a^2*b^5*c^5*d*f^2 + 96048*a^3*b^2*c^7*d^2*f + 4
608*a^3*b^3*c^6*d*f^2 - 1728*a^2*b^6*c^4*d*g^2 - 2304*a^3*b^3*c^6*e^2*f + 1
5552*a^3*b^4*c^5*d*g^2 - 48384*a^4*b^2*c^6*d*g^2 - 576*a^3*b^5*c^4*f*g^2 +
9216*a^4*b^3*c^5*f*g^2 + 193536*a^4*b*c^7*d*e*g + 6912*a^2*b^5*c^5*d*e*g -
62208*a^3*b^3*c^6*d*e*g + 2304*a^3*b^4*c^5*e*f*g - 36864*a^4*b^2*c^6*e*f*g)

/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)))*root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z - 221773824*a^6*b^3*c^7*d*e*f*z + 110886912*a^6*b^4*c^6*d*f*g*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 58982400*a^5*b^6*c^5*d*f*g*z + 16220160*a^4*b^8*c^4*d*f*g*z - 2396160*a^3*b^10*c^3*d*f*g*z + 175104*a^2*b^12*c^2*d*f*g*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4*b^7*c^5*d^2*g*z + 25362432*a^7*b^3*c^6*f^2*g*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*c^5*f^2*g*z + 3563520*a^5*b^7*c^4*f^2*g*z - 506880*a^4*b^9*c^3*f^2*g*z + 34560*a^3*b^11*c^2*f^2*g*z + 26542080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 15482880*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g + 3456*a*b^10*c^2*d*f*g^2 + 435

456*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f - 3870720*a^5*b^2*c^6*e*f^2*g - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f^2*g + 80640*a^3*b^6*c^4*e*f^2*g - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^4*c^6*d^2*e*g - 96768*a^2*b^8*c^3*d*f*g^2 - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f - 2654208*a^5*b^3*c^5*e*g^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 967680*a^5*b^3*c^5*f^2*g^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k), k, 1, 4) + ((9*x^4*(2*b*c^2*e - b^2*c*g))/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (x^2*(b^3*g - 10*a*c^2*e - 2*b^2*c*e + 5*a*b*c*g))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (b^3*e + a*b^2*g + 8*a^2*c*g - 10*a*b*c*e)/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^5*(28*a^2*c^3*d + 6*b^4*c*d + 2*a*b^3*c*f - 49*a*b^2*c^2*d + 28*a^2*b*c^2*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x*(5*b^4*d + 44*a^2*c^2*d - a*b^3*f - 37*a*b^2*c*d + 16*a^2*b*c*f))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^3*(3*b^5*d + 36*a^3*c^2*f + a*b^4*f - 20*a*b^3*c*d - 4*a^2*b*c^2*d + 5*a^2*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (3*c^2*x^6*(2*c*e - b*g))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^7*(20*a^2*c^2*f + 3*b^3*c*d - 24*a*b*c^2*d + a*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2*a*b*x^2 + 2*b*c*x^6)

3.55 $\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3} dx$

Optimal result

Integrand size = 35, antiderivative size = 679

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3} dx = -\frac{be-2ag+(2ce-bg)x^2}{4(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+\frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}+\frac{3(2ce-bg)(b+2cx^2)}{4(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+\frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab(2cd+}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)+\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{}}{\sqrt{b-}}\right.}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)-\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{}}{\sqrt{b+}}\right.}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$-\frac{3c(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

```
[Out] 1/4*(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1/4*x*(b^2
*d-a*b*f-2*a*(-a*h+c*d)+(a*b*h-2*a*c*f+b*c*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*
x^2+a)^2+3/4*(-b*g+2*c*e)*(2*c*x^2+b)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/8*x*
(3*b^4*d+a*b^3*f+8*a^2*b*c*f+4*a^2*c*(a*h+7*c*d)-a*b^2*(7*a*h+25*c*d)+c*(3*
b^3*d+a*b^2*f+20*a^2*c*f-12*a*b*(a*h+2*c*d))*x^2)/a^2/(-4*a*c+b^2)^2/(c*x^4
+b*x^2+a)-3*c*(-b*g+2*c*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+
b^2)^(5/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*c^(1
/2)*(3*b^3*d+a*b^2*f+20*a^2*c*f-12*a*b*(a*h+2*c*d)+(3*b^4*d+a*b^3*f-52*a^2*
b*c*f-6*a*b^2*(-3*a*h+5*c*d)+24*a^2*c*(a*h+7*c*d))/(-4*a*c+b^2)^(1/2))/a^2/
(-4*a*c+b^2)^2*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/16*arctan(x*2^(1/2)*c
```

```
^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b^3*d+a*b^2*f+20*a^2*c*f-12
*a*b*(a*h+2*c*d)+(-3*b^4*d-a*b^3*f+52*a^2*b*c*f+6*a*b^2*(-3*a*h+5*c*d)-24*a
^2*c*(a*h+7*c*d))/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c
+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 2.76 (sec) , antiderivative size = 679, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}} = 0.286$, Rules used = {1687, 1692, 1192, 1180, 211, 1261, 652, 628, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{-52a^2bcf+24a^2c(ah+7cd)+ab^3f-6ab^2(5cd-3ah)+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-12ab(ah+2cd)+\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-52a^2bcf+24a^2c(ah+7cd)+ab^3f-6ab^2(5cd-3ah)+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-12ab(ah+2c\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{x(cx^2(20a^2cf+ab^2f-12ab(ah+2cd)+3b^3d)+8a^2bcf+4a^2c(ah+7cd)+ab^3f-ab^2(7ah+25cd)+}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$-\frac{3c(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}+\frac{x(x^2(abh-2acf+bcd)-abf-2a(cd-ah)+b^2d)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{3\left(b+2cx^2\right)(2ce-bg)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}-\frac{-2ag+x^2(2ce-bg)+be}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] -1/4*(b*e - 2*a*g + (2*c*e - b*g)*x^2)/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2
) + (x*(b^2*d - a*b*f - 2*a*(c*d - a*h) + (b*c*d - 2*a*c*f + a*b*h)*x^2))/(
4*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (3*(2*c*e - b*g)*(b + 2*c*x^2))/
(4*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (x*(3*b^4*d + a*b^3*f + 8*a^2*b*c
*f + 4*a^2*c*(7*c*d + a*h) - a*b^2*(25*c*d + 7*a*h) + c*(3*b^3*d + a*b^2*f
+ 20*a^2*c*f - 12*a*b*(2*c*d + a*h))*x^2))/(8*a^2*(b^2 - 4*a*c)^2*(a + b*x^
2 + c*x^4)) + (Sqrt[c]*(3*b^3*d + a*b^2*f + 20*a^2*c*f - 12*a*b*(2*c*d + a*
h) + (3*b^4*d + a*b^3*f - 52*a^2*b*c*f - 6*a*b^2*(5*c*d - 3*a*h) + 24*a^2*c
*(7*c*d + a*h))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt
[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b - Sqrt[b^2 - 4*a*c]]
) + (Sqrt[c]*(3*b^3*d + a*b^2*f + 20*a^2*c*f - 12*a*b*(2*c*d + a*h) - (3*b^
4*d + a*b^3*f - 52*a^2*b*c*f - 6*a*b^2*(5*c*d - 3*a*h) + 24*a^2*c*(7*c*d +
a*h))/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a
*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (3*c*(
2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
```

```
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^3}\,dx + \int \frac{d+fx^2+hx^4}{(a+bx^2+cx^4)^3}\,dx \\
&= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a\,(b^2-4ac)\,(a+bx^2+cx^4)^2} \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(a+bx+cx^2)^3}\,dx, x, x^2\right) \\
&\quad - \frac{\int \frac{-3b^2d-abf+2a(7cd+ah)-5(bcd-2acf+abh)x^2}{(a+bx^2+cx^4)^2}\,dx}{4a\,(b^2-4ac)}
\end{aligned}$$

$$= -\frac{be - 2ag + (2ce - bg)x^2}{4(b^2 - 4ac)(a + bx^2 + cx^4)^2}$$
$$+ \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{4a(b^2 - 4ac)(a + bx^2 + cx^4)^2}$$
$$+ \frac{x(3b^4d + ab^3f + 8a^2bcf + 4a^2c(7cd + ah) - ab^2(25cd + 7ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab}{8a^2(b^2 - 4ac)^2(a + bx^2 + cx^4)}$$
$$+ \frac{\int \frac{3b^4d + ab^3f - 16a^2bcf - 3ab^2(9cd - ah) + 12a^2c(7cd + ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah))x^2}{a + bx^2 + cx^4} dx}{8a^2(b^2 - 4ac)^2}$$
$$- \frac{(3(2ce - bg))\text{Subst}\left(\int \frac{1}{(a + bx + cx^2)^2} dx, x, x^2\right)}{4(b^2 - 4ac)}$$

$$= -\frac{be - 2ag + (2ce - bg)x^2}{4(b^2 - 4ac)(a + bx^2 + cx^4)^2}$$
$$+ \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{4a(b^2 - 4ac)(a + bx^2 + cx^4)^2}$$
$$+ \frac{3(2ce - bg)(b + 2cx^2)}{4(b^2 - 4ac)^2(a + bx^2 + cx^4)}$$
$$+ \frac{x(3b^4d + ab^3f + 8a^2bcf + 4a^2c(7cd + ah) - ab^2(25cd + 7ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab}{8a^2(b^2 - 4ac)^2(a + bx^2 + cx^4)}$$
$$+ \frac{(3c(2ce - bg))\text{Subst}\left(\int \frac{1}{a + bx + cx^2} dx, x, x^2\right)}{2(b^2 - 4ac)^2}$$
$$+ \frac{\left(c\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) - \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\right) \int \frac{b}{2}}{16a^2(b^2 - 4ac)^2}$$
$$+ \frac{\left(c\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) + \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\right) \int \frac{b}{2}}{16a^2(b^2 - 4ac)^2}$$

$$= -\frac{be-2ag+(2ce-bg)x^2}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{3(2ce-bg)\left(b+2cx^2\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+\frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab(}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)+\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)-\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$-\frac{(3c(2ce-bg))\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{\left(b^2-4ac\right)^2}$$

$$= -\frac{be-2ag+(2ce-bg)x^2}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{3(2ce-bg)\left(b+2cx^2\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+\frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab(}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)+\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)-\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$-\frac{3c(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}$$

Mathematica [A] (verified)

Time = 3.90 (sec) , antiderivative size = 739, normalized size of antiderivative = 1.09

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{1}{16}\left(\frac{-8a^2(g+hx)-4bdx(b+cx^2)+8acx(d+x(e+fx))+4ab(e+x(f-x(g+hx)))}{a\left(-b^2+4ac\right)\left(a+bx^2+cx^4\right)^2}\right.$$

$$+\frac{2(4a^3chx+3b^3dx(b+cx^2)+abx(b^2f-24c^2dx^2+bc(-25d+fx^2))+a^2(-b^2(6g+7hx)+4c^2x(7d+}{a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{\sqrt{2}\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)+4a^2c\left(42cd+5\sqrt{b^2-4ac}f+6ah\right)+ab^2\left(-30cd+\sqrt{b^2-4ac}f\right.\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-}}}$$

$$-\frac{\sqrt{2}\sqrt{c}\left(3b^4d+b^3\left(-3\sqrt{b^2-4ac}d+af\right)+4a^2c\left(42cd-5\sqrt{b^2-4ac}f+6ah\right)+ab^2\left(-30cd-\sqrt{b^2-4ac}\right.\right.}{a^2\left(b^2-4ac\right)^{5/2}\sqrt{b+\sqrt{b^2}}}$$

$$-\frac{24c(-2ce+bg)\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{\left(b^2-4ac\right)^{5/2}}$$

$$\left.+\frac{24c(-2ce+bg)\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{\left(b^2-4ac\right)^{5/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] ((-8*a^2*(g + h*x) - 4*b*d*x*(b + c*x^2) + 8*a*c*x*(d + x*(e + f*x)) + 4*a*
b*(e + x*(f - x*(g + h*x))))/(a*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (2*
(4*a^3*c*h*x + 3*b^3*d*x*(b + c*x^2) + a*b*x*(b^2*f - 24*c^2*d*x^2 + b*c*(-
25*d + f*x^2)) + a^2*(-(b^2*(6*g + 7*h*x)) + 4*c^2*x*(7*d + x*(6*e + 5*f*x)
) + 4*b*c*(3*e + x*(2*f - 3*x*(g + h*x))))))/(a^2*(b^2 - 4*a*c)^2*(a + b*x^
2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d + a*f)
 + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f + 6*a*h) + a*b^2*(-30*c*d + Sqrt
[b^2 - 4*a*c]*f + 18*a*h) - 4*a*b*(6*c*Sqrt[b^2 - 4*a*c]*d + 13*a*c*f + 3*a
*Sqrt[b^2 - 4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c
]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) - (Sqrt[2]*Sqrt[
c]*(3*b^4*d + b^3*(-3*Sqrt[b^2 - 4*a*c]*d + a*f) + 4*a^2*c*(42*c*d - 5*Sqrt
[b^2 - 4*a*c]*f + 6*a*h) + a*b^2*(-30*c*d - Sqrt[b^2 - 4*a*c]*f + 18*a*h) +
 4*a*b*(6*c*Sqrt[b^2 - 4*a*c]*d - 13*a*c*f + 3*a*Sqrt[b^2 - 4*a*c]*h))*ArcT
an[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/
2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (24*c*(-2*c*e + b*g)*Log[-b + Sqrt[b^2 -
4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) + (24*c*(-2*c*e + b*g)*Log[b + Sqrt[
b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.24 (sec) , antiderivative size = 724, normalized size of antiderivative = 1.07

method	result
risch	$\frac{-\frac{c^{2}\left(12a^{2}bh-20a^{2}cf-a\,b^{2}f+24abcd-3b^{3}d\right)x^{7}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}-\frac{3(bg-2ec)c^{2}x^{6}}{2\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{c\left(4a^{3}ch-19a^{2}b^{2}h+28a^{2}bcf+28a^{2}c^{2}d+2a\,b^{3}f-49a\,b^{2}cd+6d\,b^{4}\right)x^{5}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}-\frac{}{4(}}{}$
default	Expression too large to display

```
[In] int((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBOSE)
```

```
[Out] (-1/8*c^2*(12*a^2*b*h-20*a^2*c*f-a*b^2*f+24*a*b*c*d-3*b^3*d)/a^2/(16*a^2*c^
2-8*a*b^2*c+b^4)*x^7-3/2*(b*g-2*c*e)*c^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^6+1/8
/a^2*c*(4*a^3*c*h-19*a^2*b^2*h+28*a^2*b*c*f+28*a^2*c^2*d+2*a*b^3*f-49*a*b^2
*c*d+6*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^5-9/4*b*(b*g-2*c*e)*c/(16*a^2*c^
2-8*a*b^2*c+b^4)*x^4-1/8*(16*a^3*b*c*h-36*a^3*c^2*f+5*a^2*b^3*h-5*a^2*b^2*c
*f+4*a^2*b*c^2*d-a*b^4*f+20*a*b^3*c*d-3*b^5*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^
4)*x^3-1/2*(5*a*c+b^2)*(b*g-2*c*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2-1/8*(12*a
^3*c*h+3*a^2*b^2*h-16*a^2*b*c*f-44*a^2*c^2*d+a*b^3*f+37*a*b^2*c*d-5*b^4*d)/
(16*a^2*c^2-8*a*b^2*c+b^4)/a*x-1/4*(8*a^2*c*g+a*b^2*g-10*a*b*c*e+b^3*e)/(16
*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((-c*(12*a^2*b*h-20*a^2*
c*f-a*b^2*f+24*a*b*c*d-3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2-24*(b*g
-2*c*e)*c/(16*a^2*c^2-8*a*b^2*c+b^4)*_R+(12*a^3*c*h+3*a^2*b^2*h-16*a^2*b*c*
f+84*a^2*c^2*d+a*b^3*f-27*a*b^2*c*d+3*b^4*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)
)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="fricas"
)
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**3,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx = \int \frac{hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="maxima"
)
[Out] -1/8*((12*a^2*b*c^2*h - 3*(b^3*c^2 - 8*a*b*c^3)*d - (a*b^2*c^2 + 20*a^2*c^3
)*f)*x^7 - 12*(2*a^2*c^3*e - a^2*b*c^2*g)*x^6 - ((6*b^4*c - 49*a*b^2*c^2 +
28*a^2*c^3)*d + 2*(a*b^3*c + 14*a^2*b*c^2)*f - (19*a^2*b^2*c - 4*a^3*c^2)*h
)*x^5 - 18*(2*a^2*b*c^2*e - a^2*b^2*c*g)*x^4 - ((3*b^5 - 20*a*b^3*c - 4*a^2
*b*c^2)*d + (a*b^4 + 5*a^2*b^2*c + 36*a^3*c^2)*f - (5*a^2*b^3 + 16*a^3*b*c)
*h)*x^3 - 4*(2*(a^2*b^2*c + 5*a^3*c^2)*e - (a^2*b^3 + 5*a^3*b*c)*g)*x^2 + 2
*(a^2*b^3 - 10*a^3*b*c)*e + 2*(a^3*b^2 + 8*a^4*c)*g - ((5*a*b^4 - 37*a^2*b^
2*c + 44*a^3*c^2)*d - (a^2*b^3 - 16*a^3*b*c)*f - 3*(a^3*b^2 + 4*a^4*c)*h)*x
)/((a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)*x^8 + a^4*b^4 - 8*a^5*b^2*c +
 16*a^6*c^2 + 2*(a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3)*x^6 + (a^2*b^6 -
 6*a^3*b^4*c + 32*a^5*c^3)*x^4 + 2*(a^3*b^5 - 8*a^4*b^3*c + 16*a^5*b*c^2)*x
^2) - 1/8*integrate(((12*a^2*b*c*h - 3*(b^3*c - 8*a*b*c^2)*d - (a*b^2*c + 2
0*a^2*c^2)*f)*x^2 - 3*(b^4 - 9*a*b^2*c + 28*a^2*c^2)*d - (a*b^3 - 16*a^2*b*
c)*f - 3*(a^2*b^2 + 4*a^3*c)*h - 24*(2*a^2*c^2*e - a^2*b*c*g)*x)/(c*x^4 + b
*x^2 + a), x)/(a^2*b^4 - 8*a^3*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 6854 vs. $2(627) = 1254$.

Time = 3.09 (sec) , antiderivative size = 6854, normalized size of antiderivative = 10.09

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="giac")
```

```
[Out] 1/32*(3*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
b^7*c - 2*b^8*c + 116*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 +
 26*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^6*c^2 + 34*a*b^6*c^2 + 2*b^7*c^2 - 368*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^
3 - 232*a^2*b^4*c^3 - 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 + 176
*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^
4*c^5 - 352*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b^7 + 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c
- 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2
- 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 -
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 176*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 88*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 11*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 44*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a
*c)*b^6*c - 26*(b^2 - 4*a*c)*a*b^4*c^2 - 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2
 - 4*a*c)*a^2*b^2*c^3 + 22*(b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*
c^4 - 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*a*b^7*c + 144*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 48
*a^2*b^5*c^2 + 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
4*b*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 288*a^3*b^3*c^3 - 44*a^2*
b^4*c^3 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 512*a^4*b*
c^4 + 64*a^3*b^2*c^4 + 320*a^4*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a*b^6 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^2*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a*b^5*c - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^3*b^2*c^2 - 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^2*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a*b^4*c^2 - 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^4*c^3 - 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^3*b*c^3 + 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2
*b^2*c^3 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3
*c^4 + 2*(b^2 - 4*a*c)*a*b^5*c - 40*(b^2 - 4*a*c)*a^2*b^3*c^2 - 2*(b^2 - 4*
a*c)*a*b^4*c^2 + 128*(b^2 - 4*a*c)*a^3*b*c^3 + 36*(b^2 - 4*a*c)*a^2*b^2*c^3
 + 80*(b^2 - 4*a*c)*a^3*c^4)*f + 3*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
```

```
*a^2*b^6 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 2*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*a^2*b^6*c - 16*sqrt(2)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^2*b^4*c^2 + 8*a^3*b^4*c^2 + 2*a^2*b^5*c^2 + 64*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^5*c^3 + 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4
*b*c^3 + 32*a^4*b^2*c^3 + 16*a^3*b^3*c^3 - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a^4*c^4 - 128*a^5*c^4 - 96*a^4*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*(b^2 - 4*a*c)*a^2*b^4*c - 2*(b^2 - 4*a*c)*a
^2*b^3*c^2 - 32*(b^2 - 4*a*c)*a^4*c^3 - 24*(b^2 - 4*a*c)*a^3*b*c^3)*h)*arct
an(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5
 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(
a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a
^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*
b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 +
 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/32*
(3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7*c
 + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 2
32*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 - 176*a^2*
b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896*a^4*c^5
 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*
c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c + 88*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 22*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b
^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(b^2 - 4*
a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 + 224*(b^2 - 4*a*c)*a^3*c^4 +
 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
```

*b^7 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c + 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 - 48*a^2*b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 288*a^3*b^3*c^3 + 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 512*a^4*b*c^4 - 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 2*(b^2 - 4*a*c)*a*b^5*c + 40*(b^2 - 4*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*a*b^4*c^2 - 128*(b^2 - 4*a*c)*a^3*b*c^3 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80*(b^2 - 4*a*c)*a^3*c^4)*f + 3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c + 2*a^2*b^6*c - 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 - 8*a^3*b^4*c^2 - 2*a^2*b^5*c^2 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*c^3 + 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 32*a^4*b^2*c^3 - 16*a^3*b^3*c^3 - 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 128*a^5*c^4 + 96*a^4*b*c^4 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^4*c + 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 32*(b^2 - 4*a*c)*a^4*c^3 + 24*(b^2 - 4*a*c)*a^3*b*c^3)*h)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) - 3/2*(2*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e + (b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5)*sqrt(b^2 - 4*a*c)*g)*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^

```
3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))
)/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*b^6*c - 2*b^7*c +
96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3 - 96*a^2*b^3*c^3
- 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2*c^4 - 64*a^3*c^5)
*c^2) + 3/2*(2*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e - (b
^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5)*sqrt(b^2 - 4*a*c)*g)*log(x^2 + 1/2*
(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^
4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*
c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*
b^6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3
 - 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2
*c^4 - 64*a^3*c^5)*c^2) + 1/8*(3*b^3*c^2*d*x^7 - 24*a*b*c^3*d*x^7 + a*b^2*c
^2*f*x^7 + 20*a^2*c^3*f*x^7 - 12*a^2*b*c^2*h*x^7 + 24*a^2*c^3*e*x^6 - 12*a^
2*b*c^2*g*x^6 + 6*b^4*c*d*x^5 - 49*a*b^2*c^2*d*x^5 + 28*a^2*c^3*d*x^5 + 2*a
*b^3*c*f*x^5 + 28*a^2*b*c^2*f*x^5 - 19*a^2*b^2*c*h*x^5 + 4*a^3*c^2*h*x^5 +
36*a^2*b*c^2*e*x^4 - 18*a^2*b^2*c*g*x^4 + 3*b^5*d*x^3 - 20*a*b^3*c*d*x^3 -
4*a^2*b*c^2*d*x^3 + a*b^4*f*x^3 + 5*a^2*b^2*c*f*x^3 + 36*a^3*c^2*f*x^3 - 5*
a^2*b^3*h*x^3 - 16*a^3*b*c*h*x^3 + 8*a^2*b^2*c*e*x^2 + 40*a^3*c^2*e*x^2 - 4
*a^2*b^3*g*x^2 - 20*a^3*b*c*g*x^2 + 5*a*b^4*d*x - 37*a^2*b^2*c*d*x + 44*a^3
*c^2*d*x - a^2*b^3*f*x + 16*a^3*b*c*f*x - 3*a^3*b^2*h*x - 12*a^4*c*h*x - 2*
a^2*b^3*e + 20*a^3*b*c*e - 2*a^3*b^2*g - 16*a^4*c*g)/((a^2*b^4 - 8*a^3*b^2*
c + 16*a^4*c^2)*(c*x^4 + b*x^2 + a)^2)
```

Mupad [B] (verification not implemented)

Time = 10.96 (sec) , antiderivative size = 23811, normalized size of antiderivative = 35.07

$$\int \frac{d+ex+fx^2+gx^3+hx^4}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] ((9*x^4*(2*b*c^2*e - b^2*c*g))/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (x^2*(b
^3*g - 10*a*c^2*e - 2*b^2*c*e + 5*a*b*c*g))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*
c)) - (b^3*e + a*b^2*g + 8*a^2*c*g - 10*a*b*c*e)/(4*(b^4 + 16*a^2*c^2 - 8*a
*b^2*c)) + (x^5*(28*a^2*c^3*d + 4*a^3*c^2*h + 6*b^4*c*d + 2*a*b^3*c*f - 49*
a*b^2*c^2*d + 28*a^2*b*c^2*f - 19*a^2*b^2*c*h))/(8*a^2*(b^4 + 16*a^2*c^2 -
8*a*b^2*c)) + (x^3*(3*b^5*d + 36*a^3*c^2*f - 5*a^2*b^3*h + a*b^4*f - 20*a*b
^3*c*d - 16*a^3*b*c*h - 4*a^2*b*c^2*d + 5*a^2*b^2*c*f))/(8*a^2*(b^4 + 16*a^
2*c^2 - 8*a*b^2*c)) - (x*(3*a^2*b^2*h - 44*a^2*c^2*d - 5*b^4*d + a*b^3*f +
12*a^3*c*h + 37*a*b^2*c*d - 16*a^2*b*c*f))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2
*c)) + (3*c^2*x^6*(2*c*e - b*g))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^
7*(20*a^2*c^2*f + 3*b^3*c*d - 24*a*b*c^2*d + a*b^2*c*f - 12*a^2*b*c*h))/(8*
a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2
*a*b*x^2 + 2*b*c*x^6) + symsum(log((10368*a*b^5*c^6*d^3 - 8000*a^5*c^7*f^3
```

- 567*b^7*c^5*d^3 + 169344*a^3*b*c^8*d^3 + 193536*a^4*c^8*d*e^2 - 141120*a^
4*c^8*d^2*f + 1728*a^6*b*c^5*h^3 + 315*b^8*c^4*d^2*f + 27648*a^5*c^7*e^2*h
- 135*b^9*c^3*d^2*h - 2880*a^6*c^6*f*h^2 - 67824*a^2*b^3*c^7*d^3 + 35*a^2*b
^6*c^4*f^3 + 84*a^3*b^4*c^5*f^3 - 12720*a^4*b^2*c^6*f^3 + 540*a^4*b^5*c^3*h
^3 + 4320*a^5*b^3*c^4*h^3 - 40320*a^5*c^7*d*f*h - 6237*a*b^6*c^5*d^2*f + 21
0*a*b^7*c^4*d*f^2 + 116160*a^4*b*c^7*d*f^2 - 36864*a^4*b*c^7*e^2*f + 2430*a
*b^7*c^4*d^2*h + 133056*a^4*b*c^7*d^2*h + 27648*a^5*b*c^6*d*h^2 + 26880*a^5
*b*c^6*f^2*h + 6912*a^2*b^4*c^6*d*e^2 - 62208*a^3*b^2*c^7*d*e^2 + 42372*a^2
*b^4*c^6*d^2*f - 1764*a^2*b^5*c^5*d*f^2 - 96048*a^3*b^2*c^7*d^2*f - 4608*a^
3*b^3*c^6*d*f^2 + 1728*a^2*b^6*c^4*d*g^2 + 2304*a^3*b^3*c^6*e^2*f - 15552*a
^3*b^4*c^5*d*g^2 + 48384*a^4*b^2*c^6*d*g^2 - 13716*a^2*b^5*c^5*d^2*h + 405*
a^2*b^7*c^3*d*h^2 + 12096*a^3*b^3*c^6*d^2*h - 5400*a^3*b^5*c^4*d*h^2 + 2894
4*a^4*b^3*c^5*d*h^2 + 576*a^3*b^5*c^4*f*g^2 + 6912*a^4*b^2*c^6*e^2*h - 9216
*a^4*b^3*c^5*f*g^2 - 15*a^2*b^7*c^3*f^2*h - 360*a^3*b^5*c^4*f^2*h + 135*a^3
*b^6*c^3*f*h^2 + 15696*a^4*b^3*c^5*f^2*h - 5580*a^4*b^4*c^4*f*h^2 - 20592*a
^5*b^2*c^5*f*h^2 + 1728*a^4*b^4*c^4*g^2*h + 6912*a^5*b^2*c^5*g^2*h - 193536
*a^4*b*c^7*d*e*g - 90*a*b^8*c^3*d*f*h - 27648*a^5*b*c^6*e*g*h - 6912*a^2*b^
5*c^5*d*e*g + 62208*a^3*b^3*c^6*d*e*g - 270*a^2*b^6*c^4*d*f*h + 16056*a^3*b
^4*c^5*d*f*h - 2304*a^3*b^4*c^5*e*f*g - 127008*a^4*b^2*c^6*d*f*h + 36864*a^
4*b^2*c^6*e*f*g - 6912*a^4*b^3*c^5*e*g*h)/(512*(a^4*b^12 + 4096*a^10*c^6 -
24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 614
4*a^9*b^2*c^5)) - root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^
3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 1932735
28320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b
^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 687194
76736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 46080*a^4*b^14*c*f*h*z^2 - 10598
4*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7*d*
h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2 - 1
321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*a
^7*b^6*c^6*d*f*z^2 - 456130560*a^9*b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c^5*
d*h*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 2
54017536*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^
10*b^2*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 - 61931520*a^7*b^8*c^4*f
*h*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 + 614
4000*a^6*b^10*c^3*f*h*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c
^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 3
68640*a^5*b^11*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^
14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*a^10*b*c^8*f^2*z
^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 46080*a^5
*b^13*c*h^2*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2
 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 15099494
40*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z^2
 + 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*
d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 4
77102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^

10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c^6
*h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 +
146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080*a^
8*b^7*c^4*h^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^2*z
^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a
^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^
2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 29184
0*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c
^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 120795955
2*a^10*c^9*e^2*z^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^
19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 460
8*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 13824*a^2*b^13*c*d*g*h*z
 + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^7*
d*e*h*z - 221773824*a^6*b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z + 1
10886912*a^6*b^4*c^6*d*f*g*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a^7*
b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z + 44236800*a^6*b^5*c^5*d*g*
h*z - 5898240*a^7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*h*z + 2949120*a
^6*b^6*c^4*f*g*h*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*
z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z - 6635520*a
^5*b^7*c^4*d*g*h*z - 5898240*a^6*b^5*c^5*e*f*h*z + 1474560*a^5*b^7*c^4*e*f*
h*z - 276480*a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*
b^11*c^2*d*g*h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z
 + 13271040*a^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4
*b^8*c^4*d*e*h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*
z + 27648*a^2*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3
*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*
z + 7077888*a^9*b*c^6*g*h^2*z - 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*c^
7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b
^12*c^3*d^2*e*z - 198180864*a^8*c^8*d*e*h*z + 1022754816*a^6*b^2*c^8*d^2*e*
z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 3211591
68*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4*b^7*
c^5*d^2*g*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423680*a^7*b^5*c^4*g*h^2*z - 1
105920*a^6*b^7*c^3*g*h^2*z + 138240*a^5*b^9*c^2*g*h^2*z + 25362432*a^7*b^3*
c^6*f^2*g*z + 17694720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z -
 13271040*a^6*b^5*c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z + 3563520*a^5*b
^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z - 506880*a^4*b^9*c^3*f^2*g*z -
 276480*a^5*b^8*c^3*e*h^2*z + 34560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10*c
^2*e*h^2*z + 26542080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z -
46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^
11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z +
 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c
^9*d^2*e*z - 14155776*a^9*c^7*e*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^
14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 2211840*a^6*b*c^6*e*f*g*h + 15482880
*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 4423680*a^5*b^3*c^5*e*f*g*h
+ 138240*a^4*b^5*c^4*e*f*g*h - 13824*a^3*b^7*c^3*e*f*g*h - 16588800*a^5*b^2

*c^6*d*e*g*h + 1658880*a^4*b^4*c^5*d*e*g*h + 124416*a^3*b^6*c^4*d*e*g*h - 4
1472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*c^
5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 37062144*a^5*b*c^7*d^2*f*h - 59857
92*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*c^3*d^2*f*h - 6300*a*b^10*c^2*d*f^2*h +
 16588800*a^5*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^
2*e*g + 13824*a*b^8*c^4*d*e^2*f + 1350*a*b^11*c*d*f*h^2 - 1105920*a^5*b^4*c
^4*f*g^2*h - 552960*a^6*b^2*c^5*f*g^2*h - 34560*a^4*b^6*c^3*f*g^2*h + 3456*
a^3*b^8*c^2*f*g^2*h - 1658880*a^6*b^2*c^5*e*g*h^2 - 829440*a^5*b^4*c^4*e*g*
h^2 - 20736*a^4*b^6*c^3*e*g*h^2 - 4423680*a^5*b^2*c^6*e^2*f*h + 4147200*a^5
*b^3*c^5*d*g^2*h - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4*c^5*e^2*f*h
- 31104*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2
*d*g^2*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 96
30144*a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c
^6*e*f^2*g + 2867328*a^4*b^4*c^5*d*f^2*h - 2095200*a^2*b^7*c^4*d^2*f*h - 14
14080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c
^5*e*f^2*g + 306720*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 1468
80*a^4*b^5*c^4*d*f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*
h^2 - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*
b^4*c^5*d*f*g^2 - 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2
+ 17418240*a^3*b^4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8
*c^3*d*f*g^2 + 41472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 77
41440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^
5*d*e^2*f - 1648128*a^5*b^3*c^5*f^3*h - 898560*a^6*b^3*c^4*f*h^3 - 354240*a
^5*b^5*c^3*f*h^3 - 354240*a^4*b^5*c^4*f^3*h + 43680*a^3*b^7*c^3*f^3*h - 216
00*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 + 16
58880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2*c^7*d^3*h - 12306816*a^3*b^4*c^6
*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037824*a^2*b^6*c^5*d^3*h - 2654208*a
^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h^3 + 1296000*a^5*b^4*c^4*d*h^3 -
155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^2*d^2*h^2 - 8100*a^3*b^8*c^2*d*h^
3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^9*c
^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 173779
2*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 -
 435456*a*b^7*c^5*d^2*e^2 - 2211840*a^6*c^7*e^2*f*h - 9450*b^11*c^2*d^2*f*h
 + 1612800*a^6*c^7*d*f^2*h - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^
3*f - 883200*a^6*b*c^6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^
3 - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c^4*
d^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*
d^3*f + 7350*a*b^9*c^3*d*f^3 + 414720*a^6*b^3*c^4*g^2*h^2 + 207360*a^5*b^5*
c^3*g^2*h^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1264
320*a^5*b^4*c^4*f^2*h^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^
2*h^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e^2*h^2 + 161280*a^
4*b^5*c^4*f^2*g^2 + 20736*a^4*b^5*c^4*e^2*h^2 - 20160*a^3*b^7*c^3*f^2*g^2 +
 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2*c^6*d^2*h^2 + 7962624*a^5*b^2*c
^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 - 1412640*a^3*b^6*c^4*d^2*h^2 + 4
61376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*c^3*d^2*h^2 + 8709120*a^4*b^3*c^

6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*g^2 + 6451
20*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f
^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*
a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 115200*a^7*c^6*f^2*h^2
+ 6096384*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2
+ 5644800*a^5*c^8*d^2*f^2 + 142560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h^4
 + 32400*a^5*b^6*c^2*h^4 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 +
 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 +
1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 +
 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^2*h^2 + 5806
08*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 2025*a^4*b^8*c*h^4 - 734832*a*b^6*
c^6*d^4 + 20736*a^8*c^5*h^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5
308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((983040*a^7*c^8*e*f - 324403
2*a^6*b*c^8*d*e - 884736*a^7*b*c^7*e*h - 491520*a^7*b*c^7*f*g - 4608*a^2*b^
9*c^4*d*e + 87552*a^3*b^7*c^5*d*e - 681984*a^4*b^5*c^6*d*e + 2433024*a^5*b^
3*c^7*d*e + 2304*a^2*b^10*c^3*d*g - 43776*a^3*b^8*c^4*d*g - 1536*a^3*b^8*c^
4*e*f + 340992*a^4*b^6*c^5*d*g + 39936*a^4*b^6*c^5*e*f - 1216512*a^5*b^4*c^
6*d*g - 184320*a^5*b^4*c^6*e*f + 1622016*a^6*b^2*c^7*d*g - 49152*a^6*b^2*c^
7*e*f + 768*a^3*b^9*c^3*f*g - 4608*a^4*b^7*c^4*e*h - 19968*a^4*b^7*c^4*f*g
- 18432*a^5*b^5*c^5*e*h + 92160*a^5*b^5*c^5*f*g + 368640*a^6*b^3*c^6*e*h +
24576*a^6*b^3*c^6*f*g + 2304*a^4*b^8*c^3*g*h + 9216*a^5*b^6*c^4*g*h - 18432
0*a^6*b^4*c^5*g*h + 442368*a^7*b^2*c^6*g*h)/(512*(a^4*b^12 + 4096*a^10*c^6
- 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6
144*a^9*b^2*c^5)) - root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*
c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 19327
3528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10
*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 6871
9476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 46080*a^4*b^14*c*f*h*z^2 - 105
984*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7*
d*h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2 -
 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192
*a^7*b^6*c^6*d*f*z^2 - 456130560*a^9*b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c^
5*d*h*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 +
 254017536*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*
a^10*b^2*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 - 61931520*a^7*b^8*c^4
*f*h*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 + 6
144000*a^6*b^10*c^3*f*h*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9
*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 +
 368640*a^5*b^11*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*
b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*a^10*b*c^8*f^2
*z^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 46080*a
^5*b^13*c*h^2*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z
^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 150994
9440*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z
^2 + 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*

c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 +
477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*
a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c
^6*h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2
+ 146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080*
a^8*b^7*c^4*h^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^2
*z^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912
*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*
f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291
840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10
*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959
552*a^10*c^9*e^2*z^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*
b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 4
608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 13824*a^2*b^13*c*d*g*h
*z + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^
7*d*e*h*z - 221773824*a^6*b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z +
110886912*a^6*b^4*c^6*d*f*g*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a^
7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z + 44236800*a^6*b^5*c^5*d*
g*h*z - 5898240*a^7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*h*z + 2949120
*a^6*b^6*c^4*f*g*h*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*
h*z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z - 6635520
*a^5*b^7*c^4*d*g*h*z - 5898240*a^6*b^5*c^5*e*f*h*z + 1474560*a^5*b^7*c^4*e*
f*h*z - 276480*a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^
3*b^11*c^2*d*g*h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g
*z + 13271040*a^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a
^4*b^8*c^4*d*e*h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*
g*z + 27648*a^2*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a
^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*
g*z + 7077888*a^9*b*c^6*g*h^2*z - 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*
c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a
*b^12*c^3*d^2*e*z - 198180864*a^8*c^8*d*e*h*z + 1022754816*a^6*b^2*c^8*d^2*
e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 32115
9168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4*b^
7*c^5*d^2*g*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423680*a^7*b^5*c^4*g*h^2*z -
1105920*a^6*b^7*c^3*g*h^2*z + 138240*a^5*b^9*c^2*g*h^2*z + 25362432*a^7*b^
3*c^6*f^2*g*z + 17694720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z
- 13271040*a^6*b^5*c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z + 3563520*a^5
*b^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z - 506880*a^4*b^9*c^3*f^2*g*z
- 276480*a^5*b^8*c^3*e*h^2*z + 34560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10
*c^2*e*h^2*z + 26542080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z
- 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*
b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z
+ 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7
*c^9*d^2*e*z - 14155776*a^9*c^7*e*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824*
b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 2211840*a^6*b*c^6*e*f*g*h + 154828

80*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 4423680*a^5*b^3*c^5*e*f*g*h + 138240*a^4*b^5*c^4*e*f*g*h - 13824*a^3*b^7*c^3*e*f*g*h - 16588800*a^5*b^2*c^6*d*e*g*h + 1658880*a^4*b^4*c^5*d*e*g*h + 124416*a^3*b^6*c^4*d*e*g*h - 41472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 37062144*a^5*b*c^7*d^2*f*h - 5985792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*c^3*d^2*f*h - 6300*a*b^10*c^2*d*f^2*h + 16588800*a^5*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f + 1350*a*b^11*c*d*f*h^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 552960*a^6*b^2*c^5*f*g^2*h - 34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^2*h - 1658880*a^6*b^2*c^5*e*g*h^2 - 829440*a^5*b^4*c^4*e*g*h^2 - 20736*a^4*b^6*c^3*e*g*h^2 - 4423680*a^5*b^2*c^6*e^2*f*h + 4147200*a^5*b^3*c^5*d*g^2*h - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4*c^5*e^2*f*h - 31104*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d*g^2*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630144*a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6*e*f^2*g + 2867328*a^4*b^4*c^5*d*f^2*h - 2095200*a^2*b^7*c^4*d^2*f*h - 1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f^2*g + 306720*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d*f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 - 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 41472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f - 1648128*a^5*b^3*c^5*f^3*h - 898560*a^6*b^3*c^4*f*h^3 - 354240*a^5*b^5*c^3*f*h^3 - 354240*a^4*b^5*c^4*f^3*h + 43680*a^3*b^7*c^3*f^3*h - 21600*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 + 1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2*c^7*d^3*h - 12306816*a^3*b^4*c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037824*a^2*b^6*c^5*d^3*h - 2654208*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h^3 + 1296000*a^5*b^4*c^4*d*h^3 - 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^2*d^2*h^2 - 8100*a^3*b^8*c^2*d*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 2211840*a^6*c^7*e^2*f*h - 9450*b^11*c^2*d^2*f*h + 1612800*a^6*c^7*d*f^2*h - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6*b*c^6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c^4*d^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 414720*a^6*b^3*c^4*g^2*h^2 + 207360*a^5*b^5*c^3*g^2*h^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1264320*a^5*b^4*c^4*f^2*h^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e^2*h^2 + 161280*a^4*b^5*c^4*f^2*g^2 + 20736*a^4*b^5*c^4*e^2*h^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2*c^6*d^2*h^2 + 7962624*a^5*b^2

*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 - 1412640*a^3*b^6*c^4*d^2*h^2 +
 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*c^3*d^2*h^2 + 8709120*a^4*b^3*
c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*g^2 + 64
5120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2
*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 1741824
0*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 115200*a^7*c^6*f^2*h^
2 + 6096384*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^
2 + 5644800*a^5*c^8*d^2*f^2 + 142560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h
^4 + 32400*a^5*b^6*c^2*h^4 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4
 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4
+ 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4
 + 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^2*h^2 + 58
0608*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 2025*a^4*b^8*c*h^4 - 734832*a*b^
6*c^6*d^4 + 20736*a^8*c^5*h^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 +
 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((768*a^2*b^14*c^2*d - 3145
728*a^10*c^8*h - 22020096*a^9*c^9*d - 22272*a^3*b^12*c^3*d + 282624*a^4*b^1
0*c^4*d - 2027520*a^5*b^8*c^5*d + 8847360*a^6*b^6*c^6*d - 23396352*a^7*b^4*
c^7*d + 34603008*a^8*b^2*c^8*d + 256*a^3*b^13*c^2*f - 9216*a^4*b^11*c^3*f +
 122880*a^5*b^9*c^4*f - 819200*a^6*b^7*c^5*f + 2949120*a^7*b^5*c^6*f - 5505
024*a^8*b^3*c^7*f + 768*a^4*b^12*c^2*h - 12288*a^5*b^10*c^3*h + 61440*a^6*b
^8*c^4*h - 983040*a^8*b^4*c^6*h + 3145728*a^9*b^2*c^7*h + 4194304*a^9*b*c^8
f)/(512(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280
*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(1572864*a^9*c^9*
e - 1536*a^4*b^10*c^4*e + 30720*a^5*b^8*c^5*e - 245760*a^6*b^6*c^6*e + 9830
40*a^7*b^4*c^7*e - 1966080*a^8*b^2*c^8*e + 768*a^4*b^11*c^3*g - 15360*a^5*b
^9*c^4*g + 122880*a^6*b^7*c^5*g - 491520*a^7*b^5*c^6*g + 983040*a^8*b^3*c^7
*g - 786432*a^9*b*c^8*g))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 2
40*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) +
 (root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920
*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c
^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 35
23215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*
z^4 + 65536*a^5*b^20*z^4 - 46080*a^4*b^14*c*f*h*z^2 - 105984*a^3*b^15*c*d*h
*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7*d*h*z^2 + 15099494
40*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2 - 1321205760*a^9*b^
2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*a^7*b^6*c^6*d*f*z
^2 - 456130560*a^9*b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c^5*d*h*z^2 - 366280
704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 254017536*a^8*b^6
*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^10*b^2*c^7*f*h*z
^2 + 188743680*a^7*b^7*c^5*e*g*z^2 - 61931520*a^7*b^8*c^4*f*h*z^2 + 9658368
0*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 + 6144000*a^6*b^10*c^
3*f*h*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2 + 117
9648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 368640*a^5*b^11*c
^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 -
 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*a^10*b*c^8*f^2*z^2 - 188743680*a

^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 46080*a^5*b^13*c*h^2*z^2
- 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^
6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e
^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z^2 + 4608*a^2*b^17
*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974
720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3
*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z
^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c^6*h^2*z^2 - 17432
5760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^
11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080*a^8*b^7*c^4*h^2*z^
2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^2*z^2 - 1290240*a^6
*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*
z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960
*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f
^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 177
1776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z
^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169
869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 4608*a^3*b^12*c*f*g
*h*z - 9437184*a^8*b*c^7*e*f*h*z - 13824*a^2*b^13*c*d*g*h*z + 9216*a*b^13*c
^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^7*d*e*h*z - 221773
824*a^6*b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z + 110886912*a^6*b^4
*c^6*d*f*g*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a^7*b^2*c^7*d*f*g*z
+ 117964800*a^5*b^5*c^6*d*e*f*z + 44236800*a^6*b^5*c^5*d*g*h*z - 5898240*a^
7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*h*z + 2949120*a^6*b^6*c^4*f*g*h
*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*z - 58982400*a^5
*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z - 6635520*a^5*b^7*c^4*d*g*h
*z - 5898240*a^6*b^5*c^5*e*f*h*z + 1474560*a^5*b^7*c^4*e*f*h*z - 276480*a^4
*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*b^11*c^2*d*g*h*z
 + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z + 13271040*a^5*
b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4*b^8*c^4*d*e*h*z
 - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*z + 27648*a^2*b^
12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z
 - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z + 7077888*a^9*
b*c^6*g*h^2*z - 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*c^7*f^2*g*z - 768*
a^2*b^13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z
- 198180864*a^8*c^8*d*e*h*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^
5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^5*b^5*c^6*d
^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4*b^7*c^5*d^2*g*z - 88
47360*a^8*b^3*c^5*g*h^2*z + 4423680*a^7*b^5*c^4*g*h^2*z - 1105920*a^6*b^7*c
^3*g*h^2*z + 138240*a^5*b^9*c^2*g*h^2*z + 25362432*a^7*b^3*c^6*f^2*g*z + 17
694720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^
5*c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z + 3563520*a^5*b^7*c^4*f^2*g*z +
 2211840*a^6*b^6*c^4*e*h^2*z - 506880*a^4*b^9*c^3*f^2*g*z - 276480*a^5*b^8*
c^3*e*h^2*z + 34560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10*c^2*e*h^2*z + 265
42080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8

*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z +
 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c
^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z - 141
55776*a^9*c^7*e*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z -
 6912*b^15*c*d^2*g*z + 2211840*a^6*b*c^6*e*f*g*h + 15482880*a^5*b*c^7*d*e*f
*g - 13824*a*b^9*c^3*d*e*f*g + 4423680*a^5*b^3*c^5*e*f*g*h + 138240*a^4*b^5
*c^4*e*f*g*h - 13824*a^3*b^7*c^3*e*f*g*h - 16588800*a^5*b^2*c^6*d*e*g*h + 1
658880*a^4*b^4*c^5*d*e*g*h + 124416*a^3*b^6*c^4*d*e*g*h - 41472*a^2*b^8*c^3
*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*c^5*d*e*f*g + 3870
72*a^2*b^7*c^4*d*e*f*g - 37062144*a^5*b*c^7*d^2*f*h - 5985792*a^6*b*c^6*d*f
*h^2 + 206010*a*b^9*c^3*d^2*f*h - 6300*a*b^10*c^2*d*f^2*h + 16588800*a^5*b*
c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824*a*
b^8*c^4*d*e^2*f + 1350*a*b^11*c*d*f*h^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 552
960*a^6*b^2*c^5*f*g^2*h - 34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^
2*h - 1658880*a^6*b^2*c^5*e*g*h^2 - 829440*a^5*b^4*c^4*e*g*h^2 - 20736*a^4*
b^6*c^3*e*g*h^2 - 4423680*a^5*b^2*c^6*e^2*f*h + 4147200*a^5*b^3*c^5*d*g^2*h
 - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4*c^5*e^2*f*h - 31104*a^3*b^7*
c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d*g^2*h + 15630
336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630144*a^3*b^5*c^
5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6*e*f^2*g + 286
7328*a^4*b^4*c^5*d*f^2*h - 2095200*a^2*b^7*c^4*d^2*f*h - 1414080*a^3*b^6*c^
4*d*f^2*h - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^4*c^5*e*f^2*g + 306
720*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d
*f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b
^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2
 - 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b
^4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 4
1472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^
7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f - 1648
128*a^5*b^3*c^5*f^3*h - 898560*a^6*b^3*c^4*f*h^3 - 354240*a^5*b^5*c^3*f*h^3
 - 354240*a^4*b^5*c^4*f^3*h + 43680*a^3*b^7*c^3*f^3*h - 21600*a^4*b^7*c^2*f
*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 + 1658880*a^6*b*c^6*
e^2*h^2 + 16547328*a^4*b^2*c^7*d^3*h - 12306816*a^3*b^4*c^6*d^3*h + 3731097
6*a^3*b^3*c^7*d^3*f + 3037824*a^2*b^6*c^5*d^3*h - 2654208*a^5*b^3*c^5*e*g^3
 + 1949184*a^6*b^2*c^5*d*h^3 + 1296000*a^5*b^4*c^4*d*h^3 - 155520*a^4*b^6*c
^3*d*h^3 - 40500*a*b^10*c^2*d^2*h^2 - 8100*a^3*b^8*c^2*d*h^3 + 3870720*a^5*
b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 806
8032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*
f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^
5*d^2*e^2 - 2211840*a^6*c^7*e^2*f*h - 9450*b^11*c^2*d^2*f*h + 1612800*a^6*c
^7*d*f^2*h - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6
*b*c^6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5
*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c^4*d^3*h + 4050*a^2
*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b
^9*c^3*d*f^3 + 414720*a^6*b^3*c^4*g^2*h^2 + 207360*a^5*b^5*c^3*g^2*h^2 + 51

84*a^4*b^7*c^2*g^2*h^2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1264320*a^5*b^4*c^4*
f^2*h^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 + 967680*a
^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e^2*h^2 + 161280*a^4*b^5*c^4*f^2*g^
2 + 20736*a^4*b^5*c^4*e^2*h^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2
*f^2*g^2 + 11487744*a^5*b^2*c^6*d^2*h^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 355
25376*a^4*b^2*c^7*d^2*f^2 - 1412640*a^3*b^6*c^4*d^2*h^2 + 461376*a^4*b^4*c^
5*d^2*h^2 + 375030*a^2*b^8*c^3*d^2*h^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354
560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e
^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^
3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*
e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 115200*a^7*c^6*f^2*h^2 + 6096384*a^6*c^
7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^
8*d^2*f^2 + 142560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h^4 + 32400*a^5*b^6
*c^2*h^4 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*
c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3
*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 + 28449792*a^5*c^
8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^2*h^2 + 580608*a^7*c^6*d*h^3
 - 39690*b^9*c^4*d^3*f + 2025*a^4*b^8*c*h^4 - 734832*a*b^6*c^6*d^4 + 20736*
a^8*c^5*h^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e
^4 + 35721*b^8*c^5*d^4, z, k)*x*(8388608*a^11*b*c^9 - 512*a^4*b^15*c^2 + 14
336*a^5*b^13*c^3 - 172032*a^6*b^11*c^4 + 1146880*a^7*b^9*c^5 - 4587520*a^8*
b^7*c^6 + 11010048*a^9*b^5*c^7 - 14680064*a^10*b^3*c^8))/(64*(a^4*b^12 + 40
96*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8
*b^4*c^4 - 6144*a^9*b^2*c^5))) + (x*(451584*a^6*c^9*d^2 + 18*b^12*c^3*d^2 -
 25600*a^7*c^8*f^2 + 9216*a^8*c^7*h^2 - 504*a*b^10*c^4*d^2 - 73728*a^6*b*c^
8*e^2 + 6228*a^2*b^8*c^5*d^2 - 42624*a^3*b^6*c^6*d^2 + 176256*a^4*b^4*c^7*d
^2 - 423936*a^5*b^2*c^8*d^2 - 4608*a^4*b^5*c^6*e^2 + 36864*a^5*b^3*c^7*e^2
+ 2*a^2*b^10*c^3*f^2 - 84*a^3*b^8*c^4*f^2 + 3520*a^4*b^6*c^5*f^2 - 26240*a^
5*b^4*c^6*f^2 + 59904*a^6*b^2*c^7*f^2 - 1152*a^4*b^7*c^4*g^2 + 9216*a^5*b^5
*c^5*g^2 - 18432*a^6*b^3*c^6*g^2 + 468*a^4*b^8*c^3*h^2 - 3456*a^5*b^6*c^4*h
^2 + 5760*a^6*b^4*c^5*h^2 + 129024*a^7*c^8*d*h + 12*a*b^11*c^3*d*f - 218112
*a^6*b*c^8*d*f - 9216*a^7*b*c^7*f*h - 420*a^2*b^9*c^4*d*f + 4992*a^3*b^7*c^
5*d*f - 36480*a^4*b^5*c^6*d*f + 144384*a^5*b^3*c^7*d*f + 36*a^2*b^10*c^3*d*
h - 360*a^3*b^8*c^4*d*h + 3456*a^4*b^6*c^5*d*h + 4608*a^4*b^6*c^5*e*g - 115
20*a^5*b^4*c^6*d*h - 36864*a^5*b^4*c^6*e*g - 27648*a^6*b^2*c^7*d*h + 73728*
a^6*b^2*c^7*e*g + 12*a^3*b^9*c^3*f*h - 2304*a^4*b^7*c^4*f*h + 17280*a^5*b^5
*c^5*f*h - 30720*a^6*b^3*c^6*f*h))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b
^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^
2*c^5))) + (x*(13824*a^4*c^8*e^3 - 54*b^7*c^5*d^2*e + 27*b^8*c^4*d^2*g - 17
28*a^4*b^3*c^5*g^3 - 20160*a^4*c^8*d*e*f - 2880*a^5*c^7*e*f*h + 972*a*b^5*c
^6*d^2*e + 24192*a^3*b*c^8*d^2*e - 486*a*b^6*c^5*d^2*g + 6240*a^4*b*c^7*e*f
^2 - 20736*a^4*b*c^7*e^2*g + 1728*a^5*b*c^6*e*h^2 - 7344*a^2*b^3*c^7*d^2*e
+ 3672*a^2*b^4*c^6*d^2*g - 6*a^2*b^5*c^5*e*f^2 - 12096*a^3*b^2*c^7*d^2*g +
192*a^3*b^3*c^6*e*f^2 + 10368*a^4*b^2*c^6*e*g^2 + 3*a^2*b^6*c^4*f^2*g - 96*
a^3*b^4*c^5*f^2*g - 3120*a^4*b^2*c^6*f^2*g + 1296*a^4*b^3*c^5*e*h^2 - 648*a

^4*b^4*c^4*g*h^2 - 864*a^5*b^2*c^5*g*h^2 - 36*a*b^6*c^5*d*e*f + 18*a*b^7*c^
4*d*f*g + 15552*a^4*b*c^7*d*e*h + 10080*a^4*b*c^7*d*f*g + 1440*a^5*b*c^6*f*
g*h + 900*a^2*b^4*c^6*d*e*f - 4896*a^3*b^2*c^7*d*e*f - 108*a^2*b^5*c^5*d*e*
h - 450*a^2*b^5*c^5*d*f*g + 2448*a^3*b^3*c^6*d*f*g + 54*a^2*b^6*c^4*d*g*h -
 36*a^3*b^4*c^5*e*f*h - 7776*a^4*b^2*c^6*d*g*h - 6048*a^4*b^2*c^6*e*f*h + 1
8*a^3*b^5*c^4*f*g*h + 3024*a^4*b^3*c^5*f*g*h))/(64*(a^4*b^12 + 4096*a^10*c^
6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 -
 6144*a^9*b^2*c^5)))*root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14
*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 1932
73528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^1
0*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 687
19476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 46080*a^4*b^14*c*f*h*z^2 - 10
5984*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7
*d*h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2
- 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 73216819
2*a^7*b^6*c^6*d*f*z^2 - 456130560*a^9*b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c
^5*d*h*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2
+ 254017536*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680
*a^10*b^2*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 - 61931520*a^7*b^8*c^
4*f*h*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 +
6144000*a^6*b^10*c^3*f*h*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^
9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2
+ 368640*a^5*b^11*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3
*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 440401920*a^10*b*c^8*f^
2*z^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 46080*
a^5*b^13*c*h^2*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*
z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 15099
49440*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*
z^2 + 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17
*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2
+ 477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888
*a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*
c^6*h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2
 + 146165760*a^4*b^11*c^4*d^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080
*a^8*b^7*c^4*h^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^
2*z^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 29491
2*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4
*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 29
1840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^1
0*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 120795
9552*a^10*c^9*e^2*z^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304
*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z -
4608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 13824*a^2*b^13*c*d*g*
h*z + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c
^7*d*e*h*z - 221773824*a^6*b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z

+ 110886912*a^6*b^4*c^6*d*f*g*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a
^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z + 44236800*a^6*b^5*c^5*d
*g*h*z - 5898240*a^7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*h*z + 294912
0*a^6*b^6*c^4*f*g*h*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g
*h*z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z - 663552
0*a^5*b^7*c^4*d*g*h*z - 5898240*a^6*b^5*c^5*e*f*h*z + 1474560*a^5*b^7*c^4*e
*f*h*z - 276480*a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a
^3*b^11*c^2*d*g*h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*
g*z + 13271040*a^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*
a^4*b^8*c^4*d*e*h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f
*g*z + 27648*a^2*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*
a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2
*g*z + 7077888*a^9*b*c^6*g*h^2*z - 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b
*c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*
a*b^12*c^3*d^2*e*z - 198180864*a^8*c^8*d*e*h*z + 1022754816*a^6*b^2*c^8*d^2
*e*z - 642318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 3211
59168*a^5*b^5*c^6*d^2*g*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4*b
^7*c^5*d^2*g*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423680*a^7*b^5*c^4*g*h^2*z
- 1105920*a^6*b^7*c^3*g*h^2*z + 138240*a^5*b^9*c^2*g*h^2*z + 25362432*a^7*b
^3*c^6*f^2*g*z + 17694720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*
z - 13271040*a^6*b^5*c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z + 3563520*a^
5*b^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z - 506880*a^4*b^9*c^3*f^2*g*
z - 276480*a^5*b^8*c^3*e*h^2*z + 34560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^1
0*c^2*e*h^2*z + 26542080*a^6*b^4*c^6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z
- 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2
*b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*
z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^
7*c^9*d^2*e*z - 14155776*a^9*c^7*e*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824
*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 2211840*a^6*b*c^6*e*f*g*h + 15482
880*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g + 4423680*a^5*b^3*c^5*e*f*g
*h + 138240*a^4*b^5*c^4*e*f*g*h - 13824*a^3*b^7*c^3*e*f*g*h - 16588800*a^5*
b^2*c^6*d*e*g*h + 1658880*a^4*b^4*c^5*d*e*g*h + 124416*a^3*b^6*c^4*d*e*g*h
- 41472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5
*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 37062144*a^5*b*c^7*d^2*f*h - 59
85792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*c^3*d^2*f*h - 6300*a*b^10*c^2*d*f^2*
h + 16588800*a^5*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4
*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f + 1350*a*b^11*c*d*f*h^2 - 1105920*a^5*b^
4*c^4*f*g^2*h - 552960*a^6*b^2*c^5*f*g^2*h - 34560*a^4*b^6*c^3*f*g^2*h + 34
56*a^3*b^8*c^2*f*g^2*h - 1658880*a^6*b^2*c^5*e*g*h^2 - 829440*a^5*b^4*c^4*e
*g*h^2 - 20736*a^4*b^6*c^3*e*g*h^2 - 4423680*a^5*b^2*c^6*e^2*f*h + 4147200*
a^5*b^3*c^5*d*g^2*h - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4*c^5*e^2*f
*h - 31104*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*
c^2*d*g^2*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h +
9630144*a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^
2*c^6*e*f^2*g + 2867328*a^4*b^4*c^5*d*f^2*h - 2095200*a^2*b^7*c^4*d^2*f*h -

1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^7*d^2*e*g - 645120*a^4*b^
4*c^5*e*f^2*g + 306720*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 1
46880*a^4*b^5*c^4*d*f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d
*f*h^2 - 2304*a^2*b^8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a
^4*b^4*c^5*d*f*g^2 - 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g
^2 + 17418240*a^3*b^4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*
b^8*c^3*d*f*g^2 + 41472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g -
7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6
*c^5*d*e^2*f - 1648128*a^5*b^3*c^5*f^3*h - 898560*a^6*b^3*c^4*f*h^3 - 35424
0*a^5*b^5*c^3*f*h^3 - 354240*a^4*b^5*c^4*f^3*h + 43680*a^3*b^7*c^3*f^3*h -
21600*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 +
1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2*c^7*d^3*h - 12306816*a^3*b^4*
c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037824*a^2*b^6*c^5*d^3*h - 265420
8*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h^3 + 1296000*a^5*b^4*c^4*d*h^3
- 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^2*d^2*h^2 - 8100*a^3*b^8*c^2*d
*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108864*a*b^
9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 173
7792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^
3 - 435456*a*b^7*c^5*d^2*e^2 - 2211840*a^6*c^7*e^2*f*h - 9450*b^11*c^2*d^2*
f*h + 1612800*a^6*c^7*d*f^2*h - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8
*d^3*f - 883200*a^6*b*c^6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f
*h^3 - 15482880*a^5*c^8*d*e^2*f - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c
^4*d^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c
^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 414720*a^6*b^3*c^4*g^2*h^2 + 207360*a^5*b
^5*c^3*g^2*h^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1
264320*a^5*b^4*c^4*f^2*h^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2
*f^2*h^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e^2*h^2 + 161280
*a^4*b^5*c^4*f^2*g^2 + 20736*a^4*b^5*c^4*e^2*h^2 - 20160*a^3*b^7*c^3*f^2*g^
2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2*c^6*d^2*h^2 + 7962624*a^5*b^
2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 - 1412640*a^3*b^6*c^4*d^2*h^2
+ 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*c^3*d^2*h^2 + 8709120*a^4*b^3
*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*g^2 + 6
45120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^
2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 174182
40*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 115200*a^7*c^6*f^2*h
^2 + 6096384*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 11025*b^10*c^3*d^2*f
^2 + 5644800*a^5*c^8*d^2*f^2 + 142560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*
h^4 + 32400*a^5*b^6*c^2*h^4 + 20736*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^
4 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4
+ 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^
4 + 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^2*h^2 + 5
80608*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 2025*a^4*b^8*c*h^4 - 734832*a*b
^6*c^6*d^4 + 20736*a^8*c^5*h^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4
+ 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k), k, 1, 4)

3.56 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3} dx$

Optimal result

Integrand size = 40, antiderivative size = 728

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{2acg-b(ce+ai)-(2c^2e-bcg+b^2i-2aci)x^2}{4c(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{\left(6ce-3bg+2ai+\frac{b^2i}{c}\right)(b+2cx^2)}{4(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab(2cd+}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)+\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{}}{\sqrt{b-}}\right.}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)-\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{}}{\sqrt{b+}}\right.}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{(6c^2e-3bcg+b^2i+2aci)\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

```
[Out] 1/4*x*(b^2*d-a*b*f-2*a*(-a*h+c*d)+(a*b*h-2*a*c*f+b*c*d)*x^2)/a/(-4*a*c+b^2)
/(c*x^4+b*x^2+a)^2+1/4*(2*a*c*g-b*(a*i+c*e)-(-2*a*c*i+b^2*i-b*c*g+2*c^2*e)*
x^2)/c/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1/4*(6*c*e-3*b*g+2*a*i+b^2*i/c)*(2*c*
x^2+b)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/8*x*(3*b^4*d+a*b^3*f+8*a^2*b*c*f+4*
a^2*c*(a*h+7*c*d)-a*b^2*(7*a*h+25*c*d)+c*(3*b^3*d+a*b^2*f+20*a^2*c*f-12*a*b
*(a*h+2*c*d))*x^2)/a^2/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)-(2*a*c*i+b^2*i-3*b*c*
g+6*c^2*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(5/2)+1/16*
```

```
arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b^3*d+a*b
^2*f+20*a^2*c*f-12*a*b*(a*h+2*c*d)+(3*b^4*d+a*b^3*f-52*a^2*b*c*f-6*a*b^2*(-
3*a*h+5*c*d)+24*a^2*c*(a*h+7*c*d))/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2
^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*
c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b^3*d+a*b^2*f+20*a^2*c*f-12*a*b*(a*h+2*c*d)
+(-3*b^4*d-a*b^3*f+52*a^2*b*c*f+6*a*b^2*(-3*a*h+5*c*d)-24*a^2*c*(a*h+7*c*d)
)/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/
2)
```

Rubi [A] (verified)

Time = 1.85 (sec) , antiderivative size = 728, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 11, $\frac{\text{number of rules}}{\text{integrand size}} = 0.275$, Rules used = {1687, 1692, 1192, 1180, 211, 1677, 1674, 12, 628, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{-52a^2bcf+24a^2c(ah+7cd)+ab^3f-6ab^2(5cd-3ah)+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-12ab(ah+2cd)+\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-52a^2bcf+24a^2c(ah+7cd)+ab^3f-6ab^2(5cd-3ah)+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-12ab(ah+2c\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{x(cx^2(20a^2cf+ab^2f-12ab(ah+2cd)+3b^3d)+8a^2bcf+4a^2c(ah+7cd)+ab^3f-ab^2(7ah+25cd)+}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$
$$-\frac{\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)\left(2aci+b^2i-3bcg+6c^2e\right)}{\left(b^2-4ac\right)^{5/2}}$$
$$+\frac{-(x^2(-2aci+b^2i-bcg+2c^2e))-b(ai+ce)+2acg}{4c\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{x(x^2(abh-2acf+bcd)-abf-2a(cd-ah)+b^2d)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$
$$+\frac{(b+2cx^2)\left(2ai+\frac{b^2i}{c}-3bg+6ce\right)}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] (x*(b^2*d - a*b*f - 2*a*(c*d - a*h) + (b*c*d - 2*a*c*f + a*b*h)*x^2))/(4*a*
(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (2*a*c*g - b*(c*e + a*i) - (2*c^2*e
- b*c*g + b^2*i - 2*a*c*i)*x^2)/(4*c*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) +
 ((6*c*e - 3*b*g + 2*a*i + (b^2*i)/c)*(b + 2*c*x^2))/(4*(b^2 - 4*a*c)^2*(a
+ b*x^2 + c*x^4)) + (x*(3*b^4*d + a*b^3*f + 8*a^2*b*c*f + 4*a^2*c*(7*c*d +
a*h) - a*b^2*(25*c*d + 7*a*h) + c*(3*b^3*d + a*b^2*f + 20*a^2*c*f - 12*a*b*
```

```
(2*c*d + a*h))*x^2))/(8*a^2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (Sqrt[c]
*(3*b^3*d + a*b^2*f + 20*a^2*c*f - 12*a*b*(2*c*d + a*h) + (3*b^4*d + a*b^3*
f - 52*a^2*b*c*f - 6*a*b^2*(5*c*d - 3*a*h) + 24*a^2*c*(7*c*d + a*h))/Sqrt[b
^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sq
rt[2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(3*b^3*d
+ a*b^2*f + 20*a^2*c*f - 12*a*b*(2*c*d + a*h) - (3*b^4*d + a*b^3*f - 52*a^2
*b*c*f - 6*a*b^2*(5*c*d - 3*a*h) + 24*a^2*c*(7*c*d + a*h))/Sqrt[b^2 - 4*a*c
])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*
(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - ((6*c^2*e - 3*b*c*g + b^2*i
+ 2*a*c*i)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
```

```
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
```

```
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2+hx^4}{(a+bx^2+cx^4)^3}\,dx + \int \frac{x(e+gx^2+ix^4)}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2}{(a+bx+cx^2)^3}\,dx, x, x^2\right) - \frac{\int \frac{-3b^2d-abf+2a(7cd+ah)-5(bcd-2acf+abh)x^2}{(a+bx^2+cx^4)^2}\,dx}{4a(b^2-4ac)}$$

$$= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{2acg-b(ce+ai)-(2c^2e-bcg+b^2i-2aci)x^2}{4c(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\int \frac{3b^4d+ab^3f-16a^2bcf-3ab^2(9cd-ah)+12a^2c(7cd+ah)+c(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah))x^2}{a+bx^2+cx^4}\,dx}{8a^2(b^2-4ac)^2} - \frac{\text{Subst}\left(\int \frac{6ce-3bg+2ai+\frac{b^2i}{c}}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{4(b^2-4ac)}$$

$$= \frac{x(b^2d-abf-2a(cd-ah)+(bcd-2acf+abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{2acg-b(ce+ai)-(2c^2e-bcg+b^2i-2aci)x^2}{4c(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(3b^4d+ab^3f+8a^2bcf+4a^2c(7cd+ah)-ab^2(25cd+7ah)+c(3b^3d+ab^2f+20a^2cf-12ab}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} + \frac{\left(c\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)-\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\right)\int \frac{}{\frac{b}{2}}}{16a^2(b^2-4ac)^2} + \frac{\left(c\left(3b^3d+ab^2f+20a^2cf-12ab(2cd+ah)+\frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\right)\int \frac{}{\frac{b}{2}}}{16a^2(b^2-4ac)^2} - \frac{\left(6ce-3bg+2ai+\frac{b^2i}{c}\right)\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{4(b^2-4ac)}$$

$$= \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{4a(b^2 - 4ac)(a + bx^2 + cx^4)^2} + \frac{2acg - b(ce + ai) - (2c^2e - bcg + b^2i - 2aci)x^2}{4c(b^2 - 4ac)(a + bx^2 + cx^4)^2} + \frac{\left(6ce - 3bg + 2ai + \frac{b^2i}{c}\right)(b + 2cx^2)}{4(b^2 - 4ac)^2(a + bx^2 + cx^4)} + \frac{x(3b^4d + ab^3f + 8a^2bcf + 4a^2c(7cd + ah) - ab^2(25cd + 7ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab(}{8a^2(b^2 - 4ac)^2(a + bx^2 + cx^4)} + \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) + \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2(b^2 - 4ac)^2\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) - \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2(b^2 - 4ac)^2\sqrt{b + \sqrt{b^2 - 4ac}}} + \frac{(6c^2e - 3bcg + b^2i + 2aci)\,\mathrm{Subst}\left(\int \frac{1}{a + bx + cx^2}\,dx, x, x^2\right)}{2(b^2 - 4ac)^2}$$

$$= \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{4a(b^2 - 4ac)(a + bx^2 + cx^4)^2} + \frac{2acg - b(ce + ai) - (2c^2e - bcg + b^2i - 2aci)x^2}{4c(b^2 - 4ac)(a + bx^2 + cx^4)^2} + \frac{\left(6ce - 3bg + 2ai + \frac{b^2i}{c}\right)(b + 2cx^2)}{4(b^2 - 4ac)^2(a + bx^2 + cx^4)} + \frac{x(3b^4d + ab^3f + 8a^2bcf + 4a^2c(7cd + ah) - ab^2(25cd + 7ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab(}{8a^2(b^2 - 4ac)^2(a + bx^2 + cx^4)} + \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) + \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2(b^2 - 4ac)^2\sqrt{b - \sqrt{b^2 - 4ac}}} + \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) - \frac{3b^4d + ab^3f - 52a^2bcf - 6ab^2(5cd - 3ah) + 24a^2c(7cd + ah)}{\sqrt{b^2 - 4ac}}\right)\tan^{-1}}{8\sqrt{2}a^2(b^2 - 4ac)^2\sqrt{b + \sqrt{b^2 - 4ac}}} - \frac{(6c^2e - 3bcg + b^2i + 2aci)\,\mathrm{Subst}\left(\int \frac{1}{b^2 - 4ac - x^2}\,dx, x, b + 2cx^2\right)}{(b^2 - 4ac)^2}$$

$$= \frac{x(b^2d - abf - 2a(cd - ah) + (bcd - 2acf + abh)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{2acg - b(ce + ai) - (2c^2e - bcg + b^2i - 2aci)\,x^2}{4c(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{\left(6ce - 3bg + 2ai + \frac{b^2i}{c}\right)(b + 2cx^2)}{4(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{x(3b^4d + ab^3f + 8a^2bcf + 4a^2c(7cd + ah) - ab^2(25cd + 7ah) + c(3b^3d + ab^2f + 20a^2cf - 12ab}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) + \frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-}}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d + ab^2f + 20a^2cf - 12ab(2cd + ah) - \frac{3b^4d+ab^3f-52a^2bcf-6ab^2(5cd-3ah)+24a^2c(7cd+ah)}{\sqrt{b^2-4ac}}\right)\tan^{-}}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{(6c^2e - 3bcg + b^2i + 2aci)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

Mathematica [A] (verified)

Time = 5.35 (sec) , antiderivative size = 821, normalized size of antiderivative = 1.13

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + ix^5}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{1}{16}\left(\frac{2(3b^3cdx(b + cx^2) + 4a^3c(bi + cx(h + 2ix)) + abcx(b^2f - 24c^2dx^2 + bc(-25d + fx^2)) + a^2(2b^3i + 4}{a^2c(b^2-4ac)^2(a + bx}\right.$$
$$+ \frac{4(-bcdx(b + cx^2) + a^2(bi - 2c(g + x(h + ix))) + a(b^2ix^2 + 2c^2x(d + x(e + fx)) + bc(e + x(f - x(g +}{ac(-b^2+4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{\sqrt{2}\sqrt{c}\left(3b^4d + b^3\left(3\sqrt{b^2-4ac}d + af\right) + 4a^2c\left(42cd + 5\sqrt{b^2-4ac}f + 6ah\right) + ab^2\left(-30cd + \sqrt{b^2-4ac}f\right.\right.}{a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-}}}$$
$$- \frac{\sqrt{2}\sqrt{c}\left(3b^4d + b^3\left(-3\sqrt{b^2-4ac}d + af\right) + 4a^2c\left(42cd - 5\sqrt{b^2-4ac}f + 6ah\right) + ab^2\left(-30cd - \sqrt{b^2-4ac}\right.\right.}{a^2(b^2-4ac)^{5/2}\sqrt{b+\sqrt{b^2}}}$$
$$+ \frac{8(6c^2e - 3bcg + b^2i + 2aci)\log\left(-b + \sqrt{b^2-4ac} - 2cx^2\right)}{(b^2-4ac)^{5/2}}$$
$$\left. - \frac{8(6c^2e - 3bcg + b^2i + 2aci)\log\left(b + \sqrt{b^2-4ac} + 2cx^2\right)}{(b^2-4ac)^{5/2}}\right)$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^3,x
]
```

```
[Out] ((2*(3*b^3*c*d*x*(b + c*x^2) + 4*a^3*c*(b*i + c*x*(h + 2*i*x)) + a*b*c*x*(b
^2*f - 24*c^2*d*x^2 + b*c*(-25*d + f*x^2)) + a^2*(2*b^3*i + 4*c^3*x*(7*d +
x*(6*e + 5*f*x)) + b^2*c*(-6*g + x*(-7*h + 4*i*x)) + 4*b*c^2*(3*e + x*(2*f
- 3*x*(g + h*x))))))/(a^2*c*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (4*(-(b*
c*d*x*(b + c*x^2)) + a^2*(b*i - 2*c*(g + x*(h + i*x))) + a*(b^2*i*x^2 + 2*c
^2*x*(d + x*(e + f*x)) + b*c*(e + x*(f - x*(g + h*x))))))/(a*c*(-b^2 + 4*a*
c)*(a + b*x^2 + c*x^4)^2) + (Sqrt[2]*Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4
*a*c]*d + a*f) + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f + 6*a*h) + a*b^2*(
-30*c*d + Sqrt[b^2 - 4*a*c]*f + 18*a*h) - 4*a*b*(6*c*Sqrt[b^2 - 4*a*c]*d +
13*a*c*f + 3*a*Sqrt[b^2 - 4*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sq
rt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) -
(Sqrt[2]*Sqrt[c]*(3*b^4*d + b^3*(-3*Sqrt[b^2 - 4*a*c]*d + a*f) + 4*a^2*c*(4
2*c*d - 5*Sqrt[b^2 - 4*a*c]*f + 6*a*h) + a*b^2*(-30*c*d - Sqrt[b^2 - 4*a*c]
*f + 18*a*h) + 4*a*b*(6*c*Sqrt[b^2 - 4*a*c]*d - 13*a*c*f + 3*a*Sqrt[b^2 - 4
*a*c]*h))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a^2*(b^
2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + (8*(6*c^2*e - 3*b*c*g + b^2
*i + 2*a*c*i)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) -
(8*(6*c^2*e - 3*b*c*g + b^2*i + 2*a*c*i)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^
2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.30 (sec) , antiderivative size = 795, normalized size of antiderivative = 1.09

method	result
risch	$\frac{-\frac{c^{2}\left(12a^{2}bh-20a^{2}cf-a\,b^{2}f+24abcd-3b^{3}d\right)x^{7}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{c\left(2aci+b^{2}i-3gbc+6e\,c^{2}\right)x^{6}}{32a^{2}c^{2}-16a\,b^{2}c+2b^{4}}+\frac{c\left(4a^{3}ch-19a^{2}b^{2}h+28a^{2}bcf+28a^{2}c^{2}d+2a\,b^{3}f-49a\,b^{2}cd+6d\,b^{4}\right)x^{5}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}}{}$
default	Expression too large to display

```
[In] int((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBO
SE)
```

```
[Out] (-1/8*c^2*(12*a^2*b*h-20*a^2*c*f-a*b^2*f+24*a*b*c*d-3*b^3*d)/a^2/(16*a^2*c^
2-8*a*b^2*c+b^4)*x^7+1/2*c*(2*a*c*i+b^2*i-3*b*c*g+6*c^2*e)/(16*a^2*c^2-8*a*
b^2*c+b^4)*x^6+1/8/a^2*c*(4*a^3*c*h-19*a^2*b^2*h+28*a^2*b*c*f+28*a^2*c^2*d+
2*a*b^3*f-49*a*b^2*c*d+6*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^5+3/4*b*(2*a*c
*i+b^2*i-3*b*c*g+6*c^2*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^4-1/8*(16*a^3*b*c*h-
36*a^3*c^2*f+5*a^2*b^3*h-5*a^2*b^2*c*f+4*a^2*b*c^2*d-a*b^4*f+20*a*b^3*c*d-3
*b^5*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^3-1/2*(2*a^2*c*i-5*a*b^2*i+5*a*b*c
*g-10*a*c^2*e+b^3*g-2*b^2*c*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2-1/8*(12*a^3*c
```

```
*h+3*a^2*b^2*h-16*a^2*b*c*f-44*a^2*c^2*d+a*b^3*f+37*a*b^2*c*d-5*b^4*d)/(16*
a^2*c^2-8*a*b^2*c+b^4)/a*x+1/4*(6*a^2*b*i-8*a^2*c*g-a*b^2*g+10*a*b*c*e-b^3*
e)/(16*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((-c*(12*a^2*b*h-2
0*a^2*c*f-a*b^2*f+24*a*b*c*d-3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2+8
*(2*a*c*i+b^2*i-3*b*c*g+6*c^2*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*_R+(12*a^3*c*h+
3*a^2*b^2*h-16*a^2*b*c*f+84*a^2*c^2*d+a*b^3*f-27*a*b^2*c*d+3*b^4*d)/a^2/(16
*a^2*c^2-8*a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a
))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="f
ricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**3,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx = \int \frac{ix^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="m
axima")
```

```
[Out] -1/8*((12*a^2*b*c^2*h - 3*(b^3*c^2 - 8*a*b*c^3)*d - (a*b^2*c^2 + 20*a^2*c^3
)*f)*x^7 - 4*(6*a^2*c^3*e - 3*a^2*b*c^2*g + (a^2*b^2*c + 2*a^3*c^2)*i)*x^6
- 12*a^4*b*i - ((6*b^4*c - 49*a*b^2*c^2 + 28*a^2*c^3)*d + 2*(a*b^3*c + 14*a
^2*b*c^2)*f - (19*a^2*b^2*c - 4*a^3*c^2)*h)*x^5 - 6*(6*a^2*b*c^2*e - 3*a^2*
b^2*c*g + (a^2*b^3 + 2*a^3*b*c)*i)*x^4 - ((3*b^5 - 20*a*b^3*c - 4*a^2*b*c^2
```

```
)*d + (a*b^4 + 5*a^2*b^2*c + 36*a^3*c^2)*f - (5*a^2*b^3 + 16*a^3*b*c)*h)*x^
3 - 4*(2*(a^2*b^2*c + 5*a^3*c^2)*e - (a^2*b^3 + 5*a^3*b*c)*g + (5*a^3*b^2 -
 2*a^4*c)*i)*x^2 + 2*(a^2*b^3 - 10*a^3*b*c)*e + 2*(a^3*b^2 + 8*a^4*c)*g - (
(5*a*b^4 - 37*a^2*b^2*c + 44*a^3*c^2)*d - (a^2*b^3 - 16*a^3*b*c)*f - 3*(a^3
*b^2 + 4*a^4*c)*h)*x)/((a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)*x^8 + a^4
*b^4 - 8*a^5*b^2*c + 16*a^6*c^2 + 2*(a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c
^3)*x^6 + (a^2*b^6 - 6*a^3*b^4*c + 32*a^5*c^3)*x^4 + 2*(a^3*b^5 - 8*a^4*b^3
*c + 16*a^5*b*c^2)*x^2) - 1/8*integrate(((12*a^2*b*c*h - 3*(b^3*c - 8*a*b*c
^2)*d - (a*b^2*c + 20*a^2*c^2)*f)*x^2 - 3*(b^4 - 9*a*b^2*c + 28*a^2*c^2)*d
- (a*b^3 - 16*a^2*b*c)*f - 3*(a^2*b^2 + 4*a^3*c)*h - 8*(6*a^2*c^2*e - 3*a^2
*b*c*g + (a^2*b^2 + 2*a^3*c)*i)*x)/(c*x^4 + b*x^2 + a), x)/(a^2*b^4 - 8*a^3
*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 7340 vs. $2(676) = 1352$.

Time = 3.38 (sec) , antiderivative size = 7340, normalized size of antiderivative = 10.08

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x, algorithm="g
iac")
```

```
[Out] 1/32*(3*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
b^7*c - 2*b^8*c + 116*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 +
 26*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^6*c^2 + 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^
3 - 232*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 - 176
*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^
4*c^5 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b^5*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c
+ 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2
+ 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 +
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)
```

*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a*c)*b^6*c - 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2 - 4*a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*c^4 + 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 48*a^2*b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 288*a^3*b^3*c^3 + 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 512*a^4*b*c^4 - 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 2*(b^2 - 4*a*c)*a*b^5*c - 40*(b^2 - 4*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*a*b^4*c^2 + 128*(b^2 - 4*a*c)*a^3*b*c^3 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80*(b^2 - 4*a*c)*a^3*c^4)*f + 3*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^6 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*a^2*b^6*c - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 8*a^3*b^4*c^2 - 2*a^2*b^5*c^2 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*c^3 + 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 + 32*a^4*b^2*c^3 - 16*a^3*b^3*c^3 - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 - 128*a^5*c^4 + 96*a^4*b*c^4 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*(b^2 - 4*a*c)*a^2*b^4*c + 2*(b^2 - 4*a*c)*a^2*b^3*c^2 - 32*(b^2 - 4*a*c)*a^4*c^3 + 24*(b^2 - 4*a*c)*a^3*b*c^3)*h)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 +

256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/32*
(3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^7*c
 + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 + 2
32*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 - 176*a^2*
b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896*a^4*c^5
 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*
c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6*c + 88*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 22*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b
^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(b^2 - 4*
a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 + 224*(b^2 - 4*a*c)*a^3*c^4 +
 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
*b^7 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c + 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 - 48*a^2*
b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c
^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 288*a^3*b^3*c^3 + 44*a^2*b^4*c
^3 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 512*a^4*b*c^4 -
 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*
c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*
c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4
 - 2*(b^2 - 4*a*c)*a*b^5*c + 40*(b^2 - 4*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*
a*b^4*c^2 - 128*(b^2 - 4*a*c)*a^3*b*c^3 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80

```
*(b^2 - 4*a*c)*a^3*c^4)*f + 3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*
b^6 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c - 2*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c + 2*a^2*b^6*c - 16*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^2*b^4*c^2 - 8*a^3*b^4*c^2 - 2*a^2*b^5*c^2 + 64*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^5*c^3 + 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^
3 - 32*a^4*b^2*c^3 - 16*a^3*b^3*c^3 - 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^4*c^4 + 128*a^5*c^4 + 96*a^4*b*c^4 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^4*b*c^2 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^2*b^3*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^4*c + 2*(b^2 - 4*a*c)*a^2*b^
3*c^2 + 32*(b^2 - 4*a*c)*a^4*c^3 + 24*(b^2 - 4*a*c)*a^3*b*c^3)*h)*arctan(2*
sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*
a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b
^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^
3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c
^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*
a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/2*(6*(b^
2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e + 3*(b^3*c^3 - 4*a*b*c
^4 - 2*b^2*c^4 + b*c^5)*sqrt(b^2 - 4*a*c)*g - (b^4*c^2 - 2*a*b^2*c^3 - 2*b^
3*c^3 - 8*a^2*c^4 - 4*a*b*c^4 + b^2*c^4 + 2*a*c^5)*sqrt(b^2 - 4*a*c)*i)*log
(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^
3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c -
 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((
b^8 - 16*a*b^6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*
a^3*b^2*c^3 - 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 +
 48*a^2*b^2*c^4 - 64*a^3*c^5)*c^2) + 1/4*(6*(b^5*c^3 - 8*a*b^3*c^4 - 2*b^4*
c^4 + 16*a^2*b*c^5 + 8*a*b^2*c^5 + b^3*c^5 - 4*a*b*c^6 - (b^4*c^3 - 6*a*b^2
*c^4 - 2*b^3*c^4 + 8*a^2*c^5 + 4*a*b*c^5 + b^2*c^5 - 2*a*c^6)*sqrt(b^2 - 4*
a*c))*e - 3*(b^6*c^2 - 8*a*b^4*c^3 - 2*b^5*c^3 + 16*a^2*b^2*c^4 + 8*a*b^3*c
^4 + b^4*c^4 - 4*a*b^2*c^5 - (b^5*c^2 - 6*a*b^3*c^3 - 2*b^4*c^3 + 8*a^2*b*c
^4 + 4*a*b^2*c^4 + b^3*c^4 - 2*a*b*c^5)*sqrt(b^2 - 4*a*c))*g + (b^7*c - 6*a
*b^5*c^2 - 2*b^6*c^2 + 4*a*b^4*c^3 + b^5*c^3 + 32*a^3*b*c^4 + 16*a^2*b^2*c^
4 - 2*a*b^3*c^4 - 8*a^2*b*c^5 - (b^6*c - 4*a*b^4*c^2 - 2*b^5*c^2 - 4*a^2*b^
2*c^3 + b^4*c^3 + 16*a^3*c^4 + 8*a^2*b*c^4 - 4*a^2*c^5)*sqrt(b^2 - 4*a*c))*
i)*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*
a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b
^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^
3))/((a*b^8 - 16*a^2*b^6*c - 2*a*b^7*c + 96*a^3*b^4*c^2 + 24*a^2*b^5*c^2 +
a*b^6*c^2 - 256*a^4*b^2*c^3 - 96*a^3*b^3*c^3 - 12*a^2*b^4*c^3 + 256*a^5*c^4
 + 128*a^4*b*c^4 + 48*a^3*b^2*c^4 - 64*a^4*c^5)*c^2) + 1/8*(3*b^3*c^2*d*x^7
```

```
 - 24*a*b*c^3*d*x^7 + a*b^2*c^2*f*x^7 + 20*a^2*c^3*f*x^7 - 12*a^2*b*c^2*h*x
^7 + 24*a^2*c^3*e*x^6 - 12*a^2*b*c^2*g*x^6 + 4*a^2*b^2*c*i*x^6 + 8*a^3*c^2*
i*x^6 + 6*b^4*c*d*x^5 - 49*a*b^2*c^2*d*x^5 + 28*a^2*c^3*d*x^5 + 2*a*b^3*c*f
*x^5 + 28*a^2*b*c^2*f*x^5 - 19*a^2*b^2*c*h*x^5 + 4*a^3*c^2*h*x^5 + 36*a^2*b
*c^2*e*x^4 - 18*a^2*b^2*c*g*x^4 + 6*a^2*b^3*i*x^4 + 12*a^3*b*c*i*x^4 + 3*b^
5*d*x^3 - 20*a*b^3*c*d*x^3 - 4*a^2*b*c^2*d*x^3 + a*b^4*f*x^3 + 5*a^2*b^2*c*
f*x^3 + 36*a^3*c^2*f*x^3 - 5*a^2*b^3*h*x^3 - 16*a^3*b*c*h*x^3 + 8*a^2*b^2*c
*e*x^2 + 40*a^3*c^2*e*x^2 - 4*a^2*b^3*g*x^2 - 20*a^3*b*c*g*x^2 + 20*a^3*b^2
*i*x^2 - 8*a^4*c*i*x^2 + 5*a*b^4*d*x - 37*a^2*b^2*c*d*x + 44*a^3*c^2*d*x -
a^2*b^3*f*x + 16*a^3*b*c*f*x - 3*a^3*b^2*h*x - 12*a^4*c*h*x - 2*a^2*b^3*e +
 20*a^3*b*c*e - 2*a^3*b^2*g - 16*a^4*c*g + 12*a^4*b*i)/((a^2*b^4 - 8*a^3*b^
2*c + 16*a^4*c^2)*(c*x^4 + b*x^2 + a)^2)
```

Mupad [B] (verification not implemented)

Time = 12.37 (sec) , antiderivative size = 36653, normalized size of antiderivative = 50.35

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] ((x^5*(28*a^2*c^3*d + 4*a^3*c^2*h + 6*b^4*c*d + 2*a*b^3*c*f - 49*a*b^2*c^2*
d + 28*a^2*b*c^2*f - 19*a^2*b^2*c*h))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)
) - (x^2*(b^3*g - 10*a*c^2*e - 2*b^2*c*e - 5*a*b^2*i + 2*a^2*c*i + 5*a*b*c*
g))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (b^3*e + a*b^2*g + 8*a^2*c*g - 6*a
^2*b*i - 10*a*b*c*e)/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (3*b*x^4*(6*c^2*e
 + b^2*i - 3*b*c*g + 2*a*c*i))/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^6*
(6*c^2*e + b^2*i - 3*b*c*g + 2*a*c*i))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) +
 (x^3*(3*b^5*d + 36*a^3*c^2*f - 5*a^2*b^3*h + a*b^4*f - 20*a*b^3*c*d - 16*a
^3*b*c*h - 4*a^2*b*c^2*d + 5*a^2*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b
^2*c)) - (x*(3*a^2*b^2*h - 44*a^2*c^2*d - 5*b^4*d + a*b^3*f + 12*a^3*c*h +
37*a*b^2*c*d - 16*a^2*b*c*f))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^7
*(20*a^2*c^2*f + 3*b^3*c*d - 24*a*b*c^2*d + a*b^2*c*f - 12*a^2*b*c*h))/(8*a
^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2*
a*b*x^2 + 2*b*c*x^6) + symsum(log((10368*a*b^5*c^6*d^3 - 8000*a^5*c^7*f^3 -
 567*b^7*c^5*d^3 + 169344*a^3*b*c^8*d^3 + 193536*a^4*c^8*d*e^2 - 141120*a^4
*c^8*d^2*f + 1728*a^6*b*c^5*h^3 + 315*b^8*c^4*d^2*f + 27648*a^5*c^7*e^2*h +
 21504*a^6*c^6*d*i^2 - 135*b^9*c^3*d^2*h - 2880*a^6*c^6*f*h^2 + 3072*a^7*c^
5*h*i^2 - 67824*a^2*b^3*c^7*d^3 + 35*a^2*b^6*c^4*f^3 + 84*a^3*b^4*c^5*f^3 -
 12720*a^4*b^2*c^6*f^3 + 540*a^4*b^5*c^3*h^3 + 4320*a^5*b^3*c^4*h^3 + 12902
4*a^5*c^7*d*e*i - 40320*a^5*c^7*d*f*h + 18432*a^6*c^6*e*h*i - 6237*a*b^6*c^
5*d^2*f + 210*a*b^7*c^4*d*f^2 + 116160*a^4*b*c^7*d*f^2 - 36864*a^4*b*c^7*e^
2*f + 2430*a*b^7*c^4*d^2*h + 133056*a^4*b*c^7*d^2*h + 27648*a^5*b*c^6*d*h^2
 + 26880*a^5*b*c^6*f^2*h - 4096*a^6*b*c^5*f*i^2 + 6912*a^2*b^4*c^6*d*e^2 -
```

62208*a^3*b^2*c^7*d*e^2 + 42372*a^2*b^4*c^6*d^2*f - 1764*a^2*b^5*c^5*d*f^2
- 96048*a^3*b^2*c^7*d^2*f - 4608*a^3*b^3*c^6*d*f^2 + 1728*a^2*b^6*c^4*d*g^2
+ 2304*a^3*b^3*c^6*e^2*f - 15552*a^3*b^4*c^5*d*g^2 + 48384*a^4*b^2*c^6*d*g
^2 - 13716*a^2*b^5*c^5*d^2*h + 405*a^2*b^7*c^3*d*h^2 + 12096*a^3*b^3*c^6*d^
2*h - 5400*a^3*b^5*c^4*d*h^2 + 28944*a^4*b^3*c^5*d*h^2 + 192*a^2*b^8*c^2*d*
i^2 + 576*a^3*b^5*c^4*f*g^2 - 960*a^3*b^6*c^3*d*i^2 + 6912*a^4*b^2*c^6*e^2*
h - 9216*a^4*b^3*c^5*f*g^2 - 768*a^4*b^4*c^4*d*i^2 + 14592*a^5*b^2*c^5*d*i^
2 - 15*a^2*b^7*c^3*f^2*h - 360*a^3*b^5*c^4*f^2*h + 135*a^3*b^6*c^3*f*h^2 +
15696*a^4*b^3*c^5*f^2*h - 5580*a^4*b^4*c^4*f*h^2 - 20592*a^5*b^2*c^5*f*h^2
+ 64*a^3*b^7*c^2*f*i^2 + 1728*a^4*b^4*c^4*g^2*h - 768*a^4*b^5*c^3*f*i^2 + 6
912*a^5*b^2*c^5*g^2*h - 3840*a^5*b^3*c^4*f*i^2 + 192*a^4*b^6*c^2*h*i^2 + 15
36*a^5*b^4*c^3*h*i^2 + 3840*a^6*b^2*c^4*h*i^2 - 193536*a^4*b*c^7*d*e*g - 90
*a*b^8*c^3*d*f*h - 64512*a^5*b*c^6*d*g*i - 24576*a^5*b*c^6*e*f*i - 27648*a^
5*b*c^6*e*g*h - 9216*a^6*b*c^5*g*h*i - 6912*a^2*b^5*c^5*d*e*g + 62208*a^3*b
^3*c^6*d*e*g + 2304*a^2*b^6*c^4*d*e*i - 270*a^2*b^6*c^4*d*f*h - 16128*a^3*b
^4*c^5*d*e*i + 16056*a^3*b^4*c^5*d*f*h - 2304*a^3*b^4*c^5*e*f*g + 23040*a^4
*b^2*c^6*d*e*i - 127008*a^4*b^2*c^6*d*f*h + 36864*a^4*b^2*c^6*e*f*g - 1152*
a^2*b^7*c^3*d*g*i + 8064*a^3*b^5*c^4*d*g*i + 768*a^3*b^5*c^4*e*f*i - 11520*
a^4*b^3*c^5*d*g*i - 10752*a^4*b^3*c^5*e*f*i - 6912*a^4*b^3*c^5*e*g*h - 384*
a^3*b^6*c^3*f*g*i + 2304*a^4*b^4*c^4*e*h*i + 5376*a^4*b^4*c^4*f*g*i + 13824
*a^5*b^2*c^5*e*h*i + 12288*a^5*b^2*c^5*f*g*i - 1152*a^4*b^5*c^3*g*h*i - 691
2*a^5*b^3*c^4*g*h*i)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a
^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + roo
t(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*
b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^
4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215
360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 +
65536*a^5*b^20*z^4 + 196608*a^5*b^13*c*g*i*z^2 - 46080*a^4*b^14*c*f*h*z^2
- 105984*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3
*c^7*d*h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*
z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 7321
68192*a^7*b^6*c^6*d*f*z^2 - 603979776*a^10*b^2*c^7*e*i*z^2 - 456130560*a^9*
b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c^5*d*h*z^2 + 301989888*a^10*b^3*c^6*g*
i*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 254
017536*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^10
*b^2*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 125829120*a^8*b^6*c^5*e*
i*z^2 - 62914560*a^8*b^7*c^4*g*i*z^2 - 61931520*a^7*b^8*c^4*f*h*z^2 + 23592
960*a^7*b^9*c^3*g*i*z^2 - 47185920*a^7*b^8*c^4*e*i*z^2 - 3538944*a^6*b^11*c
^2*g*i*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 +
7077888*a^6*b^10*c^3*e*i*z^2 + 6144000*a^6*b^10*c^3*f*h*z^2 - 393216*a^5*b
^12*c^2*e*i*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2
+ 1179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 368640*a^5*
b^11*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f
*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 402653184*a^11*b*c^7*g*i*z^2 - 44040
1920*a^10*b*c^8*f^2*z^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^

9*d*f*z^2 + 524288*a^6*b^12*c*i^2*z^2 + 46080*a^5*b^13*c*h^2*z^2 - 14080*a^
3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*
d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 + 805306368*a^11*c^8*e*i*z^2 - 15099
49440*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*
z^2 + 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17
*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2
+ 477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888
*a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*
c^6*h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2
 + 146165760*a^4*b^11*c^4*d^2*z^2 - 50331648*a^10*b^4*c^5*i^2*z^2 - 3355443
2*a^11*b^2*c^6*i^2*z^2 + 20971520*a^9*b^6*c^4*i^2*z^2 - 47185920*a^7*b^8*c^
4*g^2*z^2 - 26542080*a^8*b^7*c^4*h^2*z^2 - 2752512*a^7*b^10*c^2*i^2*z^2 + 2
621440*a^8*b^8*c^3*i^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*
c^5*h^2*z^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 -
 294912*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b
^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^
2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a
^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 +
1207959552*a^10*c^9*e^2*z^2 + 134217728*a^12*c^7*i^2*z^2 - 32768*a^5*b^14*i
^2*z^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 +
 169869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 3145728*a^9*b*c
^6*f*h*i*z - 27648*a^4*b^11*c*f*h*i*z + 56623104*a^8*b*c^7*d*f*i*z - 50688*
a^3*b^12*c*d*h*i*z - 4608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z -
55296*a^2*b^13*c*d*f*i*z - 13824*a^2*b^13*c*d*g*h*z + 9216*a*b^13*c^2*d*e*f
*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^7*d*e*h*z - 221773824*a^6*
b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z + 110886912*a^6*b^4*c^6*d*f
*g*z + 40108032*a^8*b^2*c^6*d*h*i*z + 2359296*a^8*b^3*c^5*f*h*i*z - 491520*
a^6*b^7*c^3*f*h*i*z + 184320*a^5*b^9*c^2*f*h*i*z - 88473600*a^6*b^4*c^6*d*e
*h*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 45613
056*a^7*b^3*c^6*d*f*i*z + 44236800*a^6*b^5*c^5*d*g*h*z - 10321920*a^6*b^6*c
^4*d*h*i*z + 7077888*a^7*b^4*c^5*d*h*i*z - 5898240*a^7*b^4*c^5*f*g*h*z + 47
18592*a^8*b^2*c^6*f*g*h*z + 2949120*a^6*b^6*c^4*f*g*h*z + 2396160*a^5*b^8*c
^3*d*h*i*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*z - 2764
8*a^4*b^10*c^2*d*h*i*z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^
6*e*f*h*z + 8847360*a^5*b^7*c^4*d*f*i*z - 6635520*a^5*b^7*c^4*d*g*h*z - 589
8240*a^6*b^5*c^5*e*f*h*z - 3809280*a^4*b^9*c^3*d*f*i*z + 2359296*a^6*b^5*c^
5*d*f*i*z + 1474560*a^5*b^7*c^4*e*f*h*z + 681984*a^3*b^11*c^2*d*f*i*z - 276
480*a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*b^11*c^2*
d*g*h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z + 132710
40*a^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4*b^8*c^4*
d*e*h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*z + 27648
*a^2*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*
d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z - 41472
*a^5*b^10*c*h^2*i*z + 7077888*a^9*b*c^6*g*h^2*z - 11008*a^3*b^12*c*f^2*i*z
- 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2

*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z - 198180864*a^
8*c^8*d*e*h*z - 66060288*a^9*c^7*d*h*i*z + 1536*a^3*b^13*f*h*i*z + 4608*a^2
*b^14*d*h*i*z - 66816*a*b^14*c*d^2*i*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 6
42318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^
5*b^5*c^6*d^2*g*z + 225312768*a^7*b^2*c^7*d^2*i*z + 223395840*a^4*b^6*c^6*d
^2*e*z - 111697920*a^4*b^7*c^5*d^2*g*z + 3538944*a^9*b^2*c^5*h^2*i*z - 7372
80*a^7*b^6*c^3*h^2*i*z + 276480*a^6*b^8*c^2*h^2*i*z - 10354688*a^8*b^2*c^6*
f^2*i*z - 43646976*a^6*b^4*c^6*d^2*i*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423
680*a^7*b^5*c^4*g*h^2*z + 2048000*a^6*b^6*c^4*f^2*i*z - 1105920*a^6*b^7*c^3
*g*h^2*z - 849920*a^5*b^8*c^3*f^2*i*z + 393216*a^7*b^4*c^5*f^2*i*z + 145920
*a^4*b^10*c^2*f^2*i*z + 138240*a^5*b^9*c^2*g*h^2*z - 32587776*a^5*b^6*c^5*d
^2*i*z + 25362432*a^7*b^3*c^6*f^2*g*z + 21657600*a^4*b^8*c^4*d^2*i*z + 1769
4720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*
c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z - 5810688*a^3*b^10*c^3*d^2*i*z +
3563520*a^5*b^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z + 845568*a^2*b^12
*c^2*d^2*i*z - 506880*a^4*b^9*c^3*f^2*g*z - 276480*a^5*b^8*c^3*e*h^2*z + 34
560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10*c^2*e*h^2*z + 26542080*a^6*b^4*c^
6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7
127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8
*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 593
0496*a^2*b^10*c^4*d^2*e*z + 1536*a*b^15*d*f*i*z - 693633024*a^7*c^9*d^2*e*z
 - 231211008*a^8*c^8*d^2*i*z - 4718592*a^10*c^6*h^2*i*z + 2304*a^4*b^12*h^2
*i*z + 13107200*a^9*c^7*f^2*i*z + 256*a^2*b^14*f^2*i*z - 14155776*a^9*c^7*e
*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^
2*g*z + 2304*b^16*d^2*i*z + 737280*a^7*b*c^5*f*g*h*i - 2304*a^3*b^9*c*f*g*h
*i - 6912*a^2*b^10*c*d*g*h*i + 11059200*a^6*b*c^6*d*e*h*i + 5160960*a^6*b*c
^6*d*f*g*i + 2211840*a^6*b*c^6*e*f*g*h + 4608*a*b^10*c^2*d*e*f*i + 15482880
*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g - 2304*a*b^11*c*d*f*g*i + 1843
200*a^6*b^3*c^4*f*g*h*i + 783360*a^5*b^5*c^3*f*g*h*i + 18432*a^4*b^7*c^2*f*
g*h*i - 5529600*a^6*b^2*c^5*d*g*h*i - 3686400*a^6*b^2*c^5*e*f*h*i - 2211840
*a^5*b^4*c^4*d*g*h*i - 1566720*a^5*b^4*c^4*e*f*h*i + 317952*a^4*b^6*c^3*d*g
*h*i - 36864*a^4*b^6*c^3*e*f*h*i + 6912*a^3*b^8*c^2*d*g*h*i + 4608*a^3*b^8*
c^2*e*f*h*i + 5160960*a^5*b^3*c^5*d*f*g*i + 4423680*a^5*b^3*c^5*e*f*g*h + 4
423680*a^5*b^3*c^5*d*e*h*i - 635904*a^4*b^5*c^4*d*e*h*i - 354816*a^3*b^7*c^
3*d*f*g*i + 322560*a^4*b^5*c^4*d*f*g*i + 138240*a^4*b^5*c^4*e*f*g*h + 59904
*a^2*b^9*c^2*d*f*g*i - 13824*a^3*b^7*c^3*e*f*g*h - 13824*a^3*b^7*c^3*d*e*h*
i + 13824*a^2*b^9*c^2*d*e*h*i - 16588800*a^5*b^2*c^6*d*e*g*h - 10321920*a^5
*b^2*c^6*d*e*f*i + 1658880*a^4*b^4*c^5*d*e*g*h + 709632*a^3*b^6*c^4*d*e*f*i
 - 645120*a^4*b^4*c^5*d*e*f*i + 124416*a^3*b^6*c^4*d*e*g*h - 119808*a^2*b^8
*c^3*d*e*f*i - 41472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 29
03040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 3456*a^4*b^8*c*g*h
^2*i - 2304*a^4*b^8*c*f*h*i^2 + 1105920*a^7*b*c^5*e*h^2*i - 384*a^2*b^10*c*
f^2*g*i - 10616832*a^6*b*c^6*e^2*g*i - 3538944*a^7*b*c^5*e*g*i^2 + 1843200*
a^7*b*c^5*d*h*i^2 + 1152*a^3*b^9*c*d*h*i^2 - 37062144*a^5*b*c^7*d^2*f*h + 2
580480*a^6*b*c^6*e*f^2*i + 65664*a*b^10*c^2*d^2*g*i + 23224320*a^5*b*c^7*d^

2*e*i - 9216*a^2*b^10*c*d*f*i^2 - 5985792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*
c^3*d^2*f*h - 131328*a*b^9*c^3*d^2*e*i - 6300*a*b^10*c^2*d*f^2*h + 16588800
*a^5*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 1
3824*a*b^8*c^4*d*e^2*f - 1474560*a^7*c^6*e*f*h*i - 10321920*a^6*c^7*d*e*f*i
 + 1350*a*b^11*c*d*f*h^2 - 552960*a^7*b^2*c^4*g*h^2*i - 552960*a^6*b^4*c^3*
g*h^2*i - 145152*a^5*b^6*c^2*g*h^2*i - 737280*a^7*b^2*c^4*f*h*i^2 - 568320*
a^6*b^4*c^3*f*h*i^2 - 136704*a^5*b^6*c^2*f*h*i^2 - 1290240*a^6*b^2*c^5*f^2*
g*i + 1105920*a^6*b^3*c^4*e*h^2*i - 860160*a^5*b^4*c^4*f^2*g*i + 290304*a^5
*b^5*c^3*e*h^2*i - 80640*a^4*b^6*c^3*f^2*g*i + 12672*a^3*b^8*c^2*f^2*g*i +
6912*a^4*b^7*c^2*e*h^2*i + 5308416*a^6*b^2*c^5*e*g^2*i - 5308416*a^5*b^3*c^
5*e^2*g*i - 3538944*a^6*b^3*c^4*e*g*i^2 + 2654208*a^5*b^4*c^4*e*g^2*i + 165
8880*a^6*b^3*c^4*d*h*i^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 884736*a^5*b^5*c^3
*e*g*i^2 - 552960*a^6*b^2*c^5*f*g^2*h + 262656*a^5*b^5*c^3*d*h*i^2 - 55296*
a^4*b^7*c^2*d*h*i^2 - 34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^2*h
- 11612160*a^5*b^2*c^6*d^2*g*i + 1720320*a^5*b^3*c^5*e*f^2*i - 1658880*a^6*
b^2*c^5*e*g*h^2 + 1596672*a^3*b^6*c^4*d^2*g*i - 829440*a^5*b^4*c^4*e*g*h^2
- 508032*a^2*b^8*c^3*d^2*g*i + 161280*a^4*b^5*c^4*e*f^2*i - 25344*a^3*b^7*c
^3*e*f^2*i - 20736*a^4*b^6*c^3*e*g*h^2 + 768*a^2*b^9*c^2*e*f^2*i - 4423680*
a^5*b^2*c^6*e^2*f*h + 4147200*a^5*b^3*c^5*d*g^2*h - 2580480*a^6*b^2*c^5*d*f
*i^2 - 967680*a^5*b^4*c^4*d*f*i^2 - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4
*b^4*c^5*e^2*f*h + 64512*a^4*b^6*c^3*d*f*i^2 + 39168*a^3*b^8*c^2*d*f*i^2 -
31104*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d
*g^2*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630
144*a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6
*e*f^2*g - 3193344*a^3*b^5*c^5*d^2*e*i + 2867328*a^4*b^4*c^5*d*f^2*h - 2095
200*a^2*b^7*c^4*d^2*f*h - 1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^
7*d^2*e*g + 1016064*a^2*b^7*c^4*d^2*e*i - 645120*a^4*b^4*c^5*e*f^2*g + 3067
20*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d*
f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b^
8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 -
 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^
4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 41
472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7
*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 18432
0*a^8*b*c^4*h^2*i^2 + 25344*a^5*b^7*c*h^2*i^2 - 884736*a^6*b^3*c^4*g^3*i -
589824*a^7*b^3*c^3*g*i^3 - 442368*a^5*b^5*c^3*g^3*i - 294912*a^6*b^5*c^2*g*
i^3 + 430080*a^7*b*c^5*f^2*i^2 - 1984*a^3*b^9*c*f^2*i^2 + 3538944*a^5*b^2*c
^6*e^3*i - 1648128*a^5*b^3*c^5*f^3*h + 1179648*a^7*b^2*c^4*e*i^3 - 898560*a
^6*b^3*c^4*f*h^3 + 589824*a^6*b^4*c^3*e*i^3 - 354240*a^5*b^5*c^3*f*h^3 - 35
4240*a^4*b^5*c^4*f^3*h + 98304*a^5*b^6*c^2*e*i^3 + 43680*a^3*b^7*c^3*f^3*h
- 21600*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2
 + 3870720*a^6*b*c^6*d^2*i^2 + 1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2
*c^7*d^3*h - 12306816*a^3*b^4*c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037
824*a^2*b^6*c^5*d^3*h - 2654208*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h
^3 + 1296000*a^5*b^4*c^4*d*h^3 - 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^

2*d^2*h^2 - 8100*a^3*b^8*c^2*d*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a
^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5
623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2
*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 245760*a^8*c^5
*f*h*i^2 + 384*a^3*b^10*f*h*i^2 + 1152*a^2*b^11*d*h*i^2 - 2211840*a^6*c^7*e
^2*f*h - 1720320*a^7*c^6*d*f*i^2 - 9450*b^11*c^2*d^2*f*h + 6912*b^11*c^2*d^
2*e*i + 1612800*a^6*c^7*d*f^2*h - 393216*a^8*b*c^4*g*i^3 - 49152*a^5*b^7*c*
g*i^3 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6*b*c^
6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5*c^8*
d*e^2*f - 9792*a*b^11*c*d^2*i^2 - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c
^4*d^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c
^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 276480*a^7*b^3*c^3*h^2*i^2 + 140544*a^6*b
^5*c^2*h^2*i^2 + 884736*a^7*b^2*c^4*g^2*i^2 + 884736*a^6*b^4*c^3*g^2*i^2 +
221184*a^5*b^6*c^2*g^2*i^2 + 501760*a^6*b^3*c^4*f^2*i^2 + 414720*a^6*b^3*c^
4*g^2*h^2 + 207360*a^5*b^5*c^3*g^2*h^2 + 170240*a^5*b^5*c^3*f^2*i^2 + 9216*
a^4*b^7*c^2*f^2*i^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 3538944*a^6*b^2*c^5*e^2*i^
2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1264320*a^5*b^4*c^4*f^2*h^2 + 884736*a^5*
b^4*c^4*e^2*i^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 +
1935360*a^5*b^3*c^5*d^2*i^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c
^5*e^2*h^2 - 532224*a^4*b^5*c^4*d^2*i^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 9676
8*a^3*b^7*c^3*d^2*i^2 + 62784*a^2*b^9*c^2*d^2*i^2 + 20736*a^4*b^5*c^4*e^2*h
^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2
*c^6*d^2*h^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 -
 1412640*a^3*b^6*c^4*d^2*h^2 + 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*
c^3*d^2*h^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 9
79776*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*
e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784
*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d
^2*e^2 - 3456*b^12*c*d^2*g*i + 384*a*b^12*d*f*i^2 + 576*a^4*b^9*h^2*i^2 + 3
538944*a^7*c^6*e^2*i^2 + 115200*a^7*c^6*f^2*h^2 + 64*a^2*b^11*f^2*i^2 + 609
6384*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 131072*a^8*b^2*c^3*i^4 + 983
04*a^7*b^4*c^2*i^4 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 142
560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h^4 + 32400*a^5*b^6*c^2*h^4 + 2073
6*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 35145
6*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728
*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 + 7077888*a^6*c^7*e^3*i + 786432
*a^8*c^5*e*i^3 + 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*
c*d^2*h^2 + 580608*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 32768*a^6*b^6*c*i^
4 + 2025*a^4*b^8*c*h^4 - 734832*a*b^6*c^6*d^4 + 576*b^13*d^2*i^2 + 65536*a^
9*c^4*i^4 + 20736*a^8*c^5*h^4 + 4096*a^5*b^8*i^4 + 49787136*a^4*c^9*d^4 + 1
60000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, l)*(root(56
371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16
*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 -
128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*
a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 655

36*a^5*b^20*z^4 + 196608*a^5*b^13*c*g*i*z^2 - 46080*a^4*b^14*c*f*h*z^2 - 10
5984*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7
*d*h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2
- 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 73216819
2*a^7*b^6*c^6*d*f*z^2 - 603979776*a^10*b^2*c^7*e*i*z^2 - 456130560*a^9*b^4*
c^6*f*h*z^2 + 390463488*a^7*b^7*c^5*d*h*z^2 + 301989888*a^10*b^3*c^6*g*i*z^
2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 2540175
36*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^10*b^2
*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 125829120*a^8*b^6*c^5*e*i*z^
2 - 62914560*a^8*b^7*c^4*g*i*z^2 - 61931520*a^7*b^8*c^4*f*h*z^2 + 23592960*
a^7*b^9*c^3*g*i*z^2 - 47185920*a^7*b^8*c^4*e*i*z^2 - 3538944*a^6*b^11*c^2*g
*i*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 + 707
7888*a^6*b^10*c^3*e*i*z^2 + 6144000*a^6*b^10*c^3*f*h*z^2 - 393216*a^5*b^12*
c^2*e*i*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2 + 1
179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 368640*a^5*b^11
*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2
 - 1207959552*a^10*b*c^8*e*g*z^2 - 402653184*a^11*b*c^7*g*i*z^2 - 440401920
*a^10*b*c^8*f^2*z^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*
f*z^2 + 524288*a^6*b^12*c*i^2*z^2 + 46080*a^5*b^13*c*h^2*z^2 - 14080*a^3*b^
15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*
z^2 - 3963617280*a^9*b*c^9*d^2*z^2 + 805306368*a^11*c^8*e*i*z^2 - 150994944
0*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z^2
+ 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d
^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 47
7102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^1
0*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c^6*
h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 1
46165760*a^4*b^11*c^4*d^2*z^2 - 50331648*a^10*b^4*c^5*i^2*z^2 - 33554432*a^
11*b^2*c^6*i^2*z^2 + 20971520*a^9*b^6*c^4*i^2*z^2 - 47185920*a^7*b^8*c^4*g^
2*z^2 - 26542080*a^8*b^7*c^4*h^2*z^2 - 2752512*a^7*b^10*c^2*i^2*z^2 + 26214
40*a^8*b^8*c^3*i^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*
h^2*z^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294
912*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c
^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 +
291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b
^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207
959552*a^10*c^9*e^2*z^2 + 134217728*a^12*c^7*i^2*z^2 - 32768*a^5*b^14*i^2*z
^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169
869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 3145728*a^9*b*c^6*f
*h*i*z - 27648*a^4*b^11*c*f*h*i*z + 56623104*a^8*b*c^7*d*f*i*z - 50688*a^3*
b^12*c*d*h*i*z - 4608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 5529
6*a^2*b^13*c*d*f*i*z - 13824*a^2*b^13*c*d*g*h*z + 9216*a*b^13*c^2*d*e*f*z -
 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^7*d*e*h*z - 221773824*a^6*b^3*
c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z + 110886912*a^6*b^4*c^6*d*f*g*z
 + 40108032*a^8*b^2*c^6*d*h*i*z + 2359296*a^8*b^3*c^5*f*h*i*z - 491520*a^6*

b^7*c^3*f*h*i*z + 184320*a^5*b^9*c^2*f*h*i*z - 88473600*a^6*b^4*c^6*d*e*h*z
 - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 45613056*
a^7*b^3*c^6*d*f*i*z + 44236800*a^6*b^5*c^5*d*g*h*z - 10321920*a^6*b^6*c^4*d
*h*i*z + 7077888*a^7*b^4*c^5*d*h*i*z - 5898240*a^7*b^4*c^5*f*g*h*z + 471859
2*a^8*b^2*c^6*f*g*h*z + 2949120*a^6*b^6*c^4*f*g*h*z + 2396160*a^5*b^8*c^3*d
*h*i*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*z - 27648*a^
4*b^10*c^2*d*h*i*z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*
f*h*z + 8847360*a^5*b^7*c^4*d*f*i*z - 6635520*a^5*b^7*c^4*d*g*h*z - 5898240
*a^6*b^5*c^5*e*f*h*z - 3809280*a^4*b^9*c^3*d*f*i*z + 2359296*a^6*b^5*c^5*d*
f*i*z + 1474560*a^5*b^7*c^4*e*f*h*z + 681984*a^3*b^11*c^2*d*f*i*z - 276480*
a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*b^11*c^2*d*g*
h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z + 13271040*a
^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4*b^8*c^4*d*e*
h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*z + 27648*a^2
*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*
f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z - 41472*a^5
*b^10*c*h^2*i*z + 7077888*a^9*b*c^6*g*h^2*z - 11008*a^3*b^12*c*f^2*i*z - 69
12*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z
 + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z - 198180864*a^8*c^
8*d*e*h*z - 66060288*a^9*c^7*d*h*i*z + 1536*a^3*b^13*f*h*i*z + 4608*a^2*b^1
4*d*h*i*z - 66816*a*b^14*c*d^2*i*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 64231
8336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^5*b^
5*c^6*d^2*g*z + 225312768*a^7*b^2*c^7*d^2*i*z + 223395840*a^4*b^6*c^6*d^2*e
*z - 111697920*a^4*b^7*c^5*d^2*g*z + 3538944*a^9*b^2*c^5*h^2*i*z - 737280*a
^7*b^6*c^3*h^2*i*z + 276480*a^6*b^8*c^2*h^2*i*z - 10354688*a^8*b^2*c^6*f^2*
i*z - 43646976*a^6*b^4*c^6*d^2*i*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423680*
a^7*b^5*c^4*g*h^2*z + 2048000*a^6*b^6*c^4*f^2*i*z - 1105920*a^6*b^7*c^3*g*h
^2*z - 849920*a^5*b^8*c^3*f^2*i*z + 393216*a^7*b^4*c^5*f^2*i*z + 145920*a^4
*b^10*c^2*f^2*i*z + 138240*a^5*b^9*c^2*g*h^2*z - 32587776*a^5*b^6*c^5*d^2*i
*z + 25362432*a^7*b^3*c^6*f^2*g*z + 21657600*a^4*b^8*c^4*d^2*i*z + 17694720
*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*c^5*
f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z - 5810688*a^3*b^10*c^3*d^2*i*z + 3563
520*a^5*b^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z + 845568*a^2*b^12*c^2
*d^2*i*z - 506880*a^4*b^9*c^3*f^2*g*z - 276480*a^5*b^8*c^3*e*h^2*z + 34560*
a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10*c^2*e*h^2*z + 26542080*a^6*b^4*c^6*e*
f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 71270
40*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4
*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496
*a^2*b^10*c^4*d^2*e*z + 1536*a*b^15*d*f*i*z - 693633024*a^7*c^9*d^2*e*z - 2
31211008*a^8*c^8*d^2*i*z - 4718592*a^10*c^6*h^2*i*z + 2304*a^4*b^12*h^2*i*z
 + 13107200*a^9*c^7*f^2*i*z + 256*a^2*b^14*f^2*i*z - 14155776*a^9*c^7*e*h^2
*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*
z + 2304*b^16*d^2*i*z + 737280*a^7*b*c^5*f*g*h*i - 2304*a^3*b^9*c*f*g*h*i -
 6912*a^2*b^10*c*d*g*h*i + 11059200*a^6*b*c^6*d*e*h*i + 5160960*a^6*b*c^6*d
*f*g*i + 2211840*a^6*b*c^6*e*f*g*h + 4608*a*b^10*c^2*d*e*f*i + 15482880*a^5

*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g - 2304*a*b^11*c*d*f*g*i + 1843200*
a^6*b^3*c^4*f*g*h*i + 783360*a^5*b^5*c^3*f*g*h*i + 18432*a^4*b^7*c^2*f*g*h*
i - 5529600*a^6*b^2*c^5*d*g*h*i - 3686400*a^6*b^2*c^5*e*f*h*i - 2211840*a^5
*b^4*c^4*d*g*h*i - 1566720*a^5*b^4*c^4*e*f*h*i + 317952*a^4*b^6*c^3*d*g*h*i
 - 36864*a^4*b^6*c^3*e*f*h*i + 6912*a^3*b^8*c^2*d*g*h*i + 4608*a^3*b^8*c^2*
e*f*h*i + 5160960*a^5*b^3*c^5*d*f*g*i + 4423680*a^5*b^3*c^5*e*f*g*h + 44236
80*a^5*b^3*c^5*d*e*h*i - 635904*a^4*b^5*c^4*d*e*h*i - 354816*a^3*b^7*c^3*d*
f*g*i + 322560*a^4*b^5*c^4*d*f*g*i + 138240*a^4*b^5*c^4*e*f*g*h + 59904*a^2
*b^9*c^2*d*f*g*i - 13824*a^3*b^7*c^3*e*f*g*h - 13824*a^3*b^7*c^3*d*e*h*i +
13824*a^2*b^9*c^2*d*e*h*i - 16588800*a^5*b^2*c^6*d*e*g*h - 10321920*a^5*b^2
*c^6*d*e*f*i + 1658880*a^4*b^4*c^5*d*e*g*h + 709632*a^3*b^6*c^4*d*e*f*i - 6
45120*a^4*b^4*c^5*d*e*f*i + 124416*a^3*b^6*c^4*d*e*g*h - 119808*a^2*b^8*c^3
*d*e*f*i - 41472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 290304
0*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 3456*a^4*b^8*c*g*h^2*i
 - 2304*a^4*b^8*c*f*h*i^2 + 1105920*a^7*b*c^5*e*h^2*i - 384*a^2*b^10*c*f^2*
g*i - 10616832*a^6*b*c^6*e^2*g*i - 3538944*a^7*b*c^5*e*g*i^2 + 1843200*a^7*
b*c^5*d*h*i^2 + 1152*a^3*b^9*c*d*h*i^2 - 37062144*a^5*b*c^7*d^2*f*h + 25804
80*a^6*b*c^6*e*f^2*i + 65664*a*b^10*c^2*d^2*g*i + 23224320*a^5*b*c^7*d^2*e*
i - 9216*a^2*b^10*c*d*f*i^2 - 5985792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*c^3*
d^2*f*h - 131328*a*b^9*c^3*d^2*e*i - 6300*a*b^10*c^2*d*f^2*h + 16588800*a^5
*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824
*a*b^8*c^4*d*e^2*f - 1474560*a^7*c^6*e*f*h*i - 10321920*a^6*c^7*d*e*f*i + 1
350*a*b^11*c*d*f*h^2 - 552960*a^7*b^2*c^4*g*h^2*i - 552960*a^6*b^4*c^3*g*h^
2*i - 145152*a^5*b^6*c^2*g*h^2*i - 737280*a^7*b^2*c^4*f*h*i^2 - 568320*a^6*
b^4*c^3*f*h*i^2 - 136704*a^5*b^6*c^2*f*h*i^2 - 1290240*a^6*b^2*c^5*f^2*g*i
+ 1105920*a^6*b^3*c^4*e*h^2*i - 860160*a^5*b^4*c^4*f^2*g*i + 290304*a^5*b^5
*c^3*e*h^2*i - 80640*a^4*b^6*c^3*f^2*g*i + 12672*a^3*b^8*c^2*f^2*g*i + 6912
*a^4*b^7*c^2*e*h^2*i + 5308416*a^6*b^2*c^5*e*g^2*i - 5308416*a^5*b^3*c^5*e^
2*g*i - 3538944*a^6*b^3*c^4*e*g*i^2 + 2654208*a^5*b^4*c^4*e*g^2*i + 1658880
*a^6*b^3*c^4*d*h*i^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 884736*a^5*b^5*c^3*e*g
*i^2 - 552960*a^6*b^2*c^5*f*g^2*h + 262656*a^5*b^5*c^3*d*h*i^2 - 55296*a^4*
b^7*c^2*d*h*i^2 - 34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^2*h - 11
612160*a^5*b^2*c^6*d^2*g*i + 1720320*a^5*b^3*c^5*e*f^2*i - 1658880*a^6*b^2*
c^5*e*g*h^2 + 1596672*a^3*b^6*c^4*d^2*g*i - 829440*a^5*b^4*c^4*e*g*h^2 - 50
8032*a^2*b^8*c^3*d^2*g*i + 161280*a^4*b^5*c^4*e*f^2*i - 25344*a^3*b^7*c^3*e
*f^2*i - 20736*a^4*b^6*c^3*e*g*h^2 + 768*a^2*b^9*c^2*e*f^2*i - 4423680*a^5*
b^2*c^6*e^2*f*h + 4147200*a^5*b^3*c^5*d*g^2*h - 2580480*a^6*b^2*c^5*d*f*i^2
 - 967680*a^5*b^4*c^4*d*f*i^2 - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4
*c^5*e^2*f*h + 64512*a^4*b^6*c^3*d*f*i^2 + 39168*a^3*b^8*c^2*d*f*i^2 - 3110
4*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d*g^2
*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630144*
a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6*e*f
^2*g - 3193344*a^3*b^5*c^5*d^2*e*i + 2867328*a^4*b^4*c^5*d*f^2*h - 2095200*
a^2*b^7*c^4*d^2*f*h - 1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^7*d^
2*e*g + 1016064*a^2*b^7*c^4*d^2*e*i - 645120*a^4*b^4*c^5*e*f^2*g + 306720*a

^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d*f*h^
2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b^8*c^
3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 - 165
8880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^4*c^
6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 41472*
a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e
^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 184320*a^
8*b*c^4*h^2*i^2 + 25344*a^5*b^7*c*h^2*i^2 - 884736*a^6*b^3*c^4*g^3*i - 5898
24*a^7*b^3*c^3*g*i^3 - 442368*a^5*b^5*c^3*g^3*i - 294912*a^6*b^5*c^2*g*i^3
+ 430080*a^7*b*c^5*f^2*i^2 - 1984*a^3*b^9*c*f^2*i^2 + 3538944*a^5*b^2*c^6*e
^3*i - 1648128*a^5*b^3*c^5*f^3*h + 1179648*a^7*b^2*c^4*e*i^3 - 898560*a^6*b
^3*c^4*f*h^3 + 589824*a^6*b^4*c^3*e*i^3 - 354240*a^5*b^5*c^3*f*h^3 - 354240
*a^4*b^5*c^4*f^3*h + 98304*a^5*b^6*c^2*e*i^3 + 43680*a^3*b^7*c^3*f^3*h - 21
600*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 + 3
870720*a^6*b*c^6*d^2*i^2 + 1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2*c^7
*d^3*h - 12306816*a^3*b^4*c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037824*
a^2*b^6*c^5*d^3*h - 2654208*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h^3 +
 1296000*a^5*b^4*c^4*d*h^3 - 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^2*d^
2*h^2 - 8100*a^3*b^8*c^2*d*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b
*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 56232
96*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2
 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 245760*a^8*c^5*f*h
*i^2 + 384*a^3*b^10*f*h*i^2 + 1152*a^2*b^11*d*h*i^2 - 2211840*a^6*c^7*e^2*f
*h - 1720320*a^7*c^6*d*f*i^2 - 9450*b^11*c^2*d^2*f*h + 6912*b^11*c^2*d^2*e*
i + 1612800*a^6*c^7*d*f^2*h - 393216*a^8*b*c^4*g*i^3 - 49152*a^5*b^7*c*g*i^
3 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6*b*c^6*f^
3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5*c^8*d*e^
2*f - 9792*a*b^11*c*d^2*i^2 - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c^4*d
^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d
^3*f + 7350*a*b^9*c^3*d*f^3 + 276480*a^7*b^3*c^3*h^2*i^2 + 140544*a^6*b^5*c
^2*h^2*i^2 + 884736*a^7*b^2*c^4*g^2*i^2 + 884736*a^6*b^4*c^3*g^2*i^2 + 2211
84*a^5*b^6*c^2*g^2*i^2 + 501760*a^6*b^3*c^4*f^2*i^2 + 414720*a^6*b^3*c^4*g^
2*h^2 + 207360*a^5*b^5*c^3*g^2*h^2 + 170240*a^5*b^5*c^3*f^2*i^2 + 9216*a^4*
b^7*c^2*f^2*i^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 3538944*a^6*b^2*c^5*e^2*i^2 +
1684224*a^6*b^2*c^5*f^2*h^2 + 1264320*a^5*b^4*c^4*f^2*h^2 + 884736*a^5*b^4*
c^4*e^2*i^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 + 1935
360*a^5*b^3*c^5*d^2*i^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e
^2*h^2 - 532224*a^4*b^5*c^4*d^2*i^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 96768*a^
3*b^7*c^3*d^2*i^2 + 62784*a^2*b^9*c^2*d^2*i^2 + 20736*a^4*b^5*c^4*e^2*h^2 -
 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2*c^6
*d^2*h^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 - 141
2640*a^3*b^6*c^4*d^2*h^2 + 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*c^3*
d^2*h^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 97977
6*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*
f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2

*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e
^2 - 3456*b^12*c*d^2*g*i + 384*a*b^12*d*f*i^2 + 576*a^4*b^9*h^2*i^2 + 35389
44*a^7*c^6*e^2*i^2 + 115200*a^7*c^6*f^2*h^2 + 64*a^2*b^11*f^2*i^2 + 6096384
*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 131072*a^8*b^2*c^3*i^4 + 98304*a
^7*b^4*c^2*i^4 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 142560*
a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h^4 + 32400*a^5*b^6*c^2*h^4 + 20736*b^
9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 351456*a^
4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3
*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 + 7077888*a^6*c^7*e^3*i + 786432*a^8
*c^5*e*i^3 + 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^
2*h^2 + 580608*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 32768*a^6*b^6*c*i^4 +
2025*a^4*b^8*c*h^4 - 734832*a*b^6*c^6*d^4 + 576*b^13*d^2*i^2 + 65536*a^9*c^
4*i^4 + 20736*a^8*c^5*h^4 + 4096*a^5*b^8*i^4 + 49787136*a^4*c^9*d^4 + 16000
0*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, l)*((768*a^2*b^
14*c^2*d - 3145728*a^10*c^8*h - 22020096*a^9*c^9*d - 22272*a^3*b^12*c^3*d +
 282624*a^4*b^10*c^4*d - 2027520*a^5*b^8*c^5*d + 8847360*a^6*b^6*c^6*d - 23
396352*a^7*b^4*c^7*d + 34603008*a^8*b^2*c^8*d + 256*a^3*b^13*c^2*f - 9216*a
^4*b^11*c^3*f + 122880*a^5*b^9*c^4*f - 819200*a^6*b^7*c^5*f + 2949120*a^7*b
^5*c^6*f - 5505024*a^8*b^3*c^7*f + 768*a^4*b^12*c^2*h - 12288*a^5*b^10*c^3*
h + 61440*a^6*b^8*c^4*h - 983040*a^8*b^4*c^6*h + 3145728*a^9*b^2*c^7*h + 41
94304*a^9*b*c^8*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6
*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(1
572864*a^9*c^9*e + 524288*a^10*c^8*i - 1536*a^4*b^10*c^4*e + 30720*a^5*b^8*
c^5*e - 245760*a^6*b^6*c^6*e + 983040*a^7*b^4*c^7*e - 1966080*a^8*b^2*c^8*e
 + 768*a^4*b^11*c^3*g - 15360*a^5*b^9*c^4*g + 122880*a^6*b^7*c^5*g - 491520
*a^7*b^5*c^6*g + 983040*a^8*b^3*c^7*g - 256*a^4*b^12*c^2*i + 4608*a^5*b^10*
c^3*i - 30720*a^6*b^8*c^4*i + 81920*a^7*b^6*c^5*i - 393216*a^9*b^2*c^7*i -
786432*a^9*b*c^8*g))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^
6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (roo
t(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*
b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^
4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215
360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 +
 65536*a^5*b^20*z^4 + 196608*a^5*b^13*c*g*i*z^2 - 46080*a^4*b^14*c*f*h*z^2
- 105984*a^3*b^15*c*d*h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3
*c^7*d*h*z^2 + 1509949440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*
z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 7321
68192*a^7*b^6*c^6*d*f*z^2 - 603979776*a^10*b^2*c^7*e*i*z^2 - 456130560*a^9*
b^4*c^6*f*h*z^2 + 390463488*a^7*b^7*c^5*d*h*z^2 + 301989888*a^10*b^3*c^6*g*
i*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 254
017536*a^8*b^6*c^5*f*h*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^10
*b^2*c^7*f*h*z^2 + 188743680*a^7*b^7*c^5*e*g*z^2 + 125829120*a^8*b^6*c^5*e*
i*z^2 - 62914560*a^8*b^7*c^4*g*i*z^2 - 61931520*a^7*b^8*c^4*f*h*z^2 + 23592
960*a^7*b^9*c^3*g*i*z^2 - 47185920*a^7*b^8*c^4*e*i*z^2 - 3538944*a^6*b^11*c
^2*g*i*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 +

7077888*a^6*b^10*c^3*e*i*z^2 + 6144000*a^6*b^10*c^3*f*h*z^2 - 393216*a^5*b^12*c^2*e*i*z^2 + 61440*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 368640*a^5*b^11*c^3*d*h*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10*b*c^8*e*g*z^2 - 402653184*a^11*b*c^7*g*i*z^2 - 440401920*a^10*b*c^8*f^2*z^2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 524288*a^6*b^12*c*i^2*z^2 + 46080*a^5*b^13*c*h^2*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 + 805306368*a^11*c^8*e*i*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z^2 + 4608*a^2*b^17*d*h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2 + 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c^6*h^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 - 50331648*a^10*b^4*c^5*i^2*z^2 - 33554432*a^11*b^2*c^6*i^2*z^2 + 20971520*a^9*b^6*c^4*i^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080*a^8*b^7*c^4*h^2*z^2 - 2752512*a^7*b^10*c^2*i^2*z^2 + 2621440*a^8*b^8*c^3*i^2*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^2*z^2 - 1290240*a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 134217728*a^12*c^7*i^2*z^2 - 32768*a^5*b^14*i^2*z^2 + 2304*a^4*b^15*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 3145728*a^9*b*c^6*f*h*i*z - 27648*a^4*b^11*c*f*h*i*z + 56623104*a^8*b*c^7*d*f*i*z - 50688*a^3*b^12*c*d*h*i*z - 4608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 55296*a^2*b^13*c*d*f*i*z - 13824*a^2*b^13*c*d*g*h*z + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f*g*z + 219414528*a^7*b^2*c^7*d*e*h*z - 221773824*a^6*b^3*c^7*d*e*f*z - 109707264*a^7*b^3*c^6*d*g*h*z + 110886912*a^6*b^4*c^6*d*f*g*z + 40108032*a^8*b^2*c^6*d*h*i*z + 2359296*a^8*b^3*c^5*f*h*i*z - 491520*a^6*b^7*c^3*f*h*i*z + 184320*a^5*b^9*c^2*f*h*i*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a^7*b^2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 45613056*a^7*b^3*c^6*d*f*i*z + 44236800*a^6*b^5*c^5*d*g*h*z - 10321920*a^6*b^6*c^4*d*h*i*z + 7077888*a^7*b^4*c^5*d*h*i*z - 5898240*a^7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*h*z + 2949120*a^6*b^6*c^4*f*g*h*z + 2396160*a^5*b^8*c^3*d*h*i*z - 737280*a^5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*z - 27648*a^4*b^10*c^2*d*h*i*z - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z + 8847360*a^5*b^7*c^4*d*f*i*z - 6635520*a^5*b^7*c^4*d*g*h*z - 5898240*a^6*b^5*c^5*e*f*h*z - 3809280*a^4*b^9*c^3*d*f*i*z + 2359296*a^6*b^5*c^5*d*f*i*z + 1474560*a^5*b^7*c^4*e*f*h*z + 681984*a^3*b^11*c^2*d*f*i*z - 276480*a^4*b^9*c^3*d*g*h*z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*b^11*c^2*d*g*h*z + 9216*a^3*b^11*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z + 132710

40*a^5*b^6*c^5*d*e*h*z - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4*b^8*c^4*
d*e*h*z - 359424*a^3*b^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*z + 27648
*a^2*b^12*c^2*d*e*h*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*
d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z - 41472
*a^5*b^10*c*h^2*i*z + 7077888*a^9*b*c^6*g*h^2*z - 11008*a^3*b^12*c*f^2*i*z
- 6912*a^4*b^11*c*g*h^2*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2
*g*z + 214272*a*b^13*c^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z - 198180864*a^
8*c^8*d*e*h*z - 66060288*a^9*c^7*d*h*i*z + 1536*a^3*b^13*f*h*i*z + 4608*a^2
*b^14*d*h*i*z - 66816*a*b^14*c*d^2*i*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 6
42318336*a^5*b^4*c^7*d^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^
5*b^5*c^6*d^2*g*z + 225312768*a^7*b^2*c^7*d^2*i*z + 223395840*a^4*b^6*c^6*d
^2*e*z - 111697920*a^4*b^7*c^5*d^2*g*z + 3538944*a^9*b^2*c^5*h^2*i*z - 7372
80*a^7*b^6*c^3*h^2*i*z + 276480*a^6*b^8*c^2*h^2*i*z - 10354688*a^8*b^2*c^6*
f^2*i*z - 43646976*a^6*b^4*c^6*d^2*i*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423
680*a^7*b^5*c^4*g*h^2*z + 2048000*a^6*b^6*c^4*f^2*i*z - 1105920*a^6*b^7*c^3
*g*h^2*z - 849920*a^5*b^8*c^3*f^2*i*z + 393216*a^7*b^4*c^5*f^2*i*z + 145920
*a^4*b^10*c^2*f^2*i*z + 138240*a^5*b^9*c^2*g*h^2*z - 32587776*a^5*b^6*c^5*d
^2*i*z + 25362432*a^7*b^3*c^6*f^2*g*z + 21657600*a^4*b^8*c^4*d^2*i*z + 1769
4720*a^8*b^2*c^6*e*h^2*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*
c^5*f^2*g*z - 8847360*a^7*b^4*c^5*e*h^2*z - 5810688*a^3*b^10*c^3*d^2*i*z +
3563520*a^5*b^7*c^4*f^2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z + 845568*a^2*b^12
*c^2*d^2*i*z - 506880*a^4*b^9*c^3*f^2*g*z - 276480*a^5*b^8*c^3*e*h^2*z + 34
560*a^3*b^11*c^2*f^2*g*z + 13824*a^4*b^10*c^2*e*h^2*z + 26542080*a^6*b^4*c^
6*e*f^2*z + 23362560*a^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7
127040*a^5*b^6*c^5*e*f^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8
*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 593
0496*a^2*b^10*c^4*d^2*e*z + 1536*a*b^15*d*f*i*z - 693633024*a^7*c^9*d^2*e*z
- 231211008*a^8*c^8*d^2*i*z - 4718592*a^10*c^6*h^2*i*z + 2304*a^4*b^12*h^2
*i*z + 13107200*a^9*c^7*f^2*i*z + 256*a^2*b^14*f^2*i*z - 14155776*a^9*c^7*e
*h^2*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^
2*g*z + 2304*b^16*d^2*i*z + 737280*a^7*b*c^5*f*g*h*i - 2304*a^3*b^9*c*f*g*h
*i - 6912*a^2*b^10*c*d*g*h*i + 11059200*a^6*b*c^6*d*e*h*i + 5160960*a^6*b*c
^6*d*f*g*i + 2211840*a^6*b*c^6*e*f*g*h + 4608*a*b^10*c^2*d*e*f*i + 15482880
*a^5*b*c^7*d*e*f*g - 13824*a*b^9*c^3*d*e*f*g - 2304*a*b^11*c*d*f*g*i + 1843
200*a^6*b^3*c^4*f*g*h*i + 783360*a^5*b^5*c^3*f*g*h*i + 18432*a^4*b^7*c^2*f*
g*h*i - 5529600*a^6*b^2*c^5*d*g*h*i - 3686400*a^6*b^2*c^5*e*f*h*i - 2211840
*a^5*b^4*c^4*d*g*h*i - 1566720*a^5*b^4*c^4*e*f*h*i + 317952*a^4*b^6*c^3*d*g
*h*i - 36864*a^4*b^6*c^3*e*f*h*i + 6912*a^3*b^8*c^2*d*g*h*i + 4608*a^3*b^8*
c^2*e*f*h*i + 5160960*a^5*b^3*c^5*d*f*g*i + 4423680*a^5*b^3*c^5*e*f*g*h + 4
423680*a^5*b^3*c^5*d*e*h*i - 635904*a^4*b^5*c^4*d*e*h*i - 354816*a^3*b^7*c^
3*d*f*g*i + 322560*a^4*b^5*c^4*d*f*g*i + 138240*a^4*b^5*c^4*e*f*g*h + 59904
*a^2*b^9*c^2*d*f*g*i - 13824*a^3*b^7*c^3*e*f*g*h - 13824*a^3*b^7*c^3*d*e*h*
i + 13824*a^2*b^9*c^2*d*e*h*i - 16588800*a^5*b^2*c^6*d*e*g*h - 10321920*a^5
*b^2*c^6*d*e*f*i + 1658880*a^4*b^4*c^5*d*e*g*h + 709632*a^3*b^6*c^4*d*e*f*i
- 645120*a^4*b^4*c^5*d*e*f*i + 124416*a^3*b^6*c^4*d*e*g*h - 119808*a^2*b^8

*c^3*d*e*f*i - 41472*a^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 29
03040*a^3*b^5*c^5*d*e*f*g + 387072*a^2*b^7*c^4*d*e*f*g - 3456*a^4*b^8*c*g*h
^2*i - 2304*a^4*b^8*c*f*h*i^2 + 1105920*a^7*b*c^5*e*h^2*i - 384*a^2*b^10*c*
f^2*g*i - 10616832*a^6*b*c^6*e^2*g*i - 3538944*a^7*b*c^5*e*g*i^2 + 1843200*
a^7*b*c^5*d*h*i^2 + 1152*a^3*b^9*c*d*h*i^2 - 37062144*a^5*b*c^7*d^2*f*h + 2
580480*a^6*b*c^6*e*f^2*i + 65664*a*b^10*c^2*d^2*g*i + 23224320*a^5*b*c^7*d^
2*e*i - 9216*a^2*b^10*c*d*f*i^2 - 5985792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*
c^3*d^2*f*h - 131328*a*b^9*c^3*d^2*e*i - 6300*a*b^10*c^2*d*f^2*h + 16588800
*a^5*b*c^7*d*e^2*h + 3456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 1
3824*a*b^8*c^4*d*e^2*f - 1474560*a^7*c^6*e*f*h*i - 10321920*a^6*c^7*d*e*f*i
 + 1350*a*b^11*c*d*f*h^2 - 552960*a^7*b^2*c^4*g*h^2*i - 552960*a^6*b^4*c^3*
g*h^2*i - 145152*a^5*b^6*c^2*g*h^2*i - 737280*a^7*b^2*c^4*f*h*i^2 - 568320*
a^6*b^4*c^3*f*h*i^2 - 136704*a^5*b^6*c^2*f*h*i^2 - 1290240*a^6*b^2*c^5*f^2*
g*i + 1105920*a^6*b^3*c^4*e*h^2*i - 860160*a^5*b^4*c^4*f^2*g*i + 290304*a^5
*b^5*c^3*e*h^2*i - 80640*a^4*b^6*c^3*f^2*g*i + 12672*a^3*b^8*c^2*f^2*g*i +
6912*a^4*b^7*c^2*e*h^2*i + 5308416*a^6*b^2*c^5*e*g^2*i - 5308416*a^5*b^3*c^
5*e^2*g*i - 3538944*a^6*b^3*c^4*e*g*i^2 + 2654208*a^5*b^4*c^4*e*g^2*i + 165
8880*a^6*b^3*c^4*d*h*i^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 884736*a^5*b^5*c^3
*e*g*i^2 - 552960*a^6*b^2*c^5*f*g^2*h + 262656*a^5*b^5*c^3*d*h*i^2 - 55296*
a^4*b^7*c^2*d*h*i^2 - 34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^2*h
- 11612160*a^5*b^2*c^6*d^2*g*i + 1720320*a^5*b^3*c^5*e*f^2*i - 1658880*a^6*
b^2*c^5*e*g*h^2 + 1596672*a^3*b^6*c^4*d^2*g*i - 829440*a^5*b^4*c^4*e*g*h^2
- 508032*a^2*b^8*c^3*d^2*g*i + 161280*a^4*b^5*c^4*e*f^2*i - 25344*a^3*b^7*c
^3*e*f^2*i - 20736*a^4*b^6*c^3*e*g*h^2 + 768*a^2*b^9*c^2*e*f^2*i - 4423680*
a^5*b^2*c^6*e^2*f*h + 4147200*a^5*b^3*c^5*d*g^2*h - 2580480*a^6*b^2*c^5*d*f
*i^2 - 967680*a^5*b^4*c^4*d*f*i^2 - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4
*b^4*c^5*e^2*f*h + 64512*a^4*b^6*c^3*d*f*i^2 + 39168*a^3*b^8*c^2*d*f*i^2 -
31104*a^3*b^7*c^3*d*g^2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d
*g^2*h + 15630336*a^5*b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630
144*a^3*b^5*c^5*d^2*f*h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6
*e*f^2*g - 3193344*a^3*b^5*c^5*d^2*e*i + 2867328*a^4*b^4*c^5*d*f^2*h - 2095
200*a^2*b^7*c^4*d^2*f*h - 1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^
7*d^2*e*g + 1016064*a^2*b^7*c^4*d^2*e*i - 645120*a^4*b^4*c^5*e*f^2*g + 3067
20*a^3*b^7*c^3*d*f*h^2 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d*
f*h^2 + 80640*a^3*b^6*c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b^
8*c^3*e*f^2*g - 3870720*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 -
 1658880*a^4*b^3*c^6*d*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^
4*c^6*d^2*e*g - 124416*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 41
472*a^2*b^7*c^4*d*e^2*h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7
*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 18432
0*a^8*b*c^4*h^2*i^2 + 25344*a^5*b^7*c*h^2*i^2 - 884736*a^6*b^3*c^4*g^3*i -
589824*a^7*b^3*c^3*g*i^3 - 442368*a^5*b^5*c^3*g^3*i - 294912*a^6*b^5*c^2*g*
i^3 + 430080*a^7*b*c^5*f^2*i^2 - 1984*a^3*b^9*c*f^2*i^2 + 3538944*a^5*b^2*c
^6*e^3*i - 1648128*a^5*b^3*c^5*f^3*h + 1179648*a^7*b^2*c^4*e*i^3 - 898560*a
^6*b^3*c^4*f*h^3 + 589824*a^6*b^4*c^3*e*i^3 - 354240*a^5*b^5*c^3*f*h^3 - 35

4240*a^4*b^5*c^4*f^3*h + 98304*a^5*b^6*c^2*e*i^3 + 43680*a^3*b^7*c^3*f^3*h
- 21600*a^4*b^7*c^2*f*h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2
 + 3870720*a^6*b*c^6*d^2*i^2 + 1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2
*c^7*d^3*h - 12306816*a^3*b^4*c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037
824*a^2*b^6*c^5*d^3*h - 2654208*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h
^3 + 1296000*a^5*b^4*c^4*d*h^3 - 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^
2*d^2*h^2 - 8100*a^3*b^8*c^2*d*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a
^4*b*c^8*d^2*e^2 - 108864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5
623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2
*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 245760*a^8*c^5
*f*h*i^2 + 384*a^3*b^10*f*h*i^2 + 1152*a^2*b^11*d*h*i^2 - 2211840*a^6*c^7*e
^2*f*h - 1720320*a^7*c^6*d*f*i^2 - 9450*b^11*c^2*d^2*f*h + 6912*b^11*c^2*d^
2*e*i + 1612800*a^6*c^7*d*f^2*h - 393216*a^8*b*c^4*g*i^3 - 49152*a^5*b^7*c*
g*i^3 - 20736*b^10*c^3*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6*b*c^
6*f^3*h - 317952*a^7*b*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5*c^8*
d*e^2*f - 9792*a*b^11*c*d^2*i^2 - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c
^4*d^3*h + 4050*a^2*b^10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c
^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 276480*a^7*b^3*c^3*h^2*i^2 + 140544*a^6*b
^5*c^2*h^2*i^2 + 884736*a^7*b^2*c^4*g^2*i^2 + 884736*a^6*b^4*c^3*g^2*i^2 +
221184*a^5*b^6*c^2*g^2*i^2 + 501760*a^6*b^3*c^4*f^2*i^2 + 414720*a^6*b^3*c^
4*g^2*h^2 + 207360*a^5*b^5*c^3*g^2*h^2 + 170240*a^5*b^5*c^3*f^2*i^2 + 9216*
a^4*b^7*c^2*f^2*i^2 + 5184*a^4*b^7*c^2*g^2*h^2 + 3538944*a^6*b^2*c^5*e^2*i^
2 + 1684224*a^6*b^2*c^5*f^2*h^2 + 1264320*a^5*b^4*c^4*f^2*h^2 + 884736*a^5*
b^4*c^4*e^2*i^2 + 126720*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 +
1935360*a^5*b^3*c^5*d^2*i^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c
^5*e^2*h^2 - 532224*a^4*b^5*c^4*d^2*i^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 9676
8*a^3*b^7*c^3*d^2*i^2 + 62784*a^2*b^9*c^2*d^2*i^2 + 20736*a^4*b^5*c^4*e^2*h
^2 - 20160*a^3*b^7*c^3*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2
*c^6*d^2*h^2 + 7962624*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 -
 1412640*a^3*b^6*c^4*d^2*h^2 + 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*
c^3*d^2*h^2 + 8709120*a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 9
79776*a^2*b^7*c^4*d^2*g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*
e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784
*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d
^2*e^2 - 3456*b^12*c*d^2*g*i + 384*a*b^12*d*f*i^2 + 576*a^4*b^9*h^2*i^2 + 3
538944*a^7*c^6*e^2*i^2 + 115200*a^7*c^6*f^2*h^2 + 64*a^2*b^11*f^2*i^2 + 609
6384*a^6*c^7*d^2*h^2 + 5184*b^11*c^2*d^2*g^2 + 131072*a^8*b^2*c^3*i^4 + 983
04*a^7*b^4*c^2*i^4 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 142
560*a^6*b^4*c^3*h^4 + 103680*a^7*b^2*c^4*h^4 + 32400*a^5*b^6*c^2*h^4 + 2073
6*b^9*c^4*d^2*e^2 + 331776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 35145
6*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728
*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 + 7077888*a^6*c^7*e^3*i + 786432
*a^8*c^5*e*i^3 + 28449792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*
c*d^2*h^2 + 580608*a^7*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 32768*a^6*b^6*c*i^
4 + 2025*a^4*b^8*c*h^4 - 734832*a*b^6*c^6*d^4 + 576*b^13*d^2*i^2 + 65536*a^

9*c^4*i^4 + 20736*a^8*c^5*h^4 + 4096*a^5*b^8*i^4 + 49787136*a^4*c^9*d^4 + 1
60000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, l)*x*(83886
08*a^11*b*c^9 - 512*a^4*b^15*c^2 + 14336*a^5*b^13*c^3 - 172032*a^6*b^11*c^4
 + 1146880*a^7*b^9*c^5 - 4587520*a^8*b^7*c^6 + 11010048*a^9*b^5*c^7 - 14680
064*a^10*b^3*c^8))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*
b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (3244
032*a^6*b*c^8*d*e - 327680*a^8*c^7*f*i - 983040*a^7*c^8*e*f + 1081344*a^7*b
*c^7*d*i + 884736*a^7*b*c^7*e*h + 491520*a^7*b*c^7*f*g + 294912*a^8*b*c^6*h
*i + 4608*a^2*b^9*c^4*d*e - 87552*a^3*b^7*c^5*d*e + 681984*a^4*b^5*c^6*d*e
- 2433024*a^5*b^3*c^7*d*e - 2304*a^2*b^10*c^3*d*g + 43776*a^3*b^8*c^4*d*g +
 1536*a^3*b^8*c^4*e*f - 340992*a^4*b^6*c^5*d*g - 39936*a^4*b^6*c^5*e*f + 12
16512*a^5*b^4*c^6*d*g + 184320*a^5*b^4*c^6*e*f - 1622016*a^6*b^2*c^7*d*g +
49152*a^6*b^2*c^7*e*f + 768*a^2*b^11*c^2*d*i - 13056*a^3*b^9*c^3*d*i - 768*
a^3*b^9*c^3*f*g + 84480*a^4*b^7*c^4*d*i + 4608*a^4*b^7*c^4*e*h + 19968*a^4*
b^7*c^4*f*g - 178176*a^5*b^5*c^5*d*i + 18432*a^5*b^5*c^5*e*h - 92160*a^5*b^
5*c^5*f*g - 270336*a^6*b^3*c^6*d*i - 368640*a^6*b^3*c^6*e*h - 24576*a^6*b^3
*c^6*f*g + 256*a^3*b^10*c^2*f*i - 6144*a^4*b^8*c^3*f*i - 2304*a^4*b^8*c^3*g
*h + 17408*a^5*b^6*c^4*f*i - 9216*a^5*b^6*c^4*g*h + 69632*a^6*b^4*c^5*f*i +
 184320*a^6*b^4*c^5*g*h - 147456*a^7*b^2*c^6*f*i - 442368*a^7*b^2*c^6*g*h +
 768*a^4*b^9*c^2*h*i + 4608*a^5*b^7*c^3*h*i - 55296*a^6*b^5*c^4*h*i + 24576
*a^7*b^3*c^5*h*i)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*
b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - (x*(45
1584*a^6*c^9*d^2 + 18*b^12*c^3*d^2 - 25600*a^7*c^8*f^2 + 9216*a^8*c^7*h^2 -
 504*a*b^10*c^4*d^2 - 73728*a^6*b*c^8*e^2 - 8192*a^8*b*c^6*i^2 + 6228*a^2*b
^8*c^5*d^2 - 42624*a^3*b^6*c^6*d^2 + 176256*a^4*b^4*c^7*d^2 - 423936*a^5*b^
2*c^8*d^2 - 4608*a^4*b^5*c^6*e^2 + 36864*a^5*b^3*c^7*e^2 + 2*a^2*b^10*c^3*f
^2 - 84*a^3*b^8*c^4*f^2 + 3520*a^4*b^6*c^5*f^2 - 26240*a^5*b^4*c^6*f^2 + 59
904*a^6*b^2*c^7*f^2 - 1152*a^4*b^7*c^4*g^2 + 9216*a^5*b^5*c^5*g^2 - 18432*a
^6*b^3*c^6*g^2 + 468*a^4*b^8*c^3*h^2 - 3456*a^5*b^6*c^4*h^2 + 5760*a^6*b^4*
c^5*h^2 - 128*a^4*b^9*c^2*i^2 + 512*a^5*b^7*c^3*i^2 + 1536*a^6*b^5*c^4*i^2
- 4096*a^7*b^3*c^5*i^2 + 129024*a^7*c^8*d*h + 12*a*b^11*c^3*d*f - 218112*a^
6*b*c^8*d*f - 49152*a^7*b*c^7*e*i - 9216*a^7*b*c^7*f*h - 420*a^2*b^9*c^4*d*
f + 4992*a^3*b^7*c^5*d*f - 36480*a^4*b^5*c^6*d*f + 144384*a^5*b^3*c^7*d*f +
 36*a^2*b^10*c^3*d*h - 360*a^3*b^8*c^4*d*h + 3456*a^4*b^6*c^5*d*h + 4608*a^
4*b^6*c^5*e*g - 11520*a^5*b^4*c^6*d*h - 36864*a^5*b^4*c^6*e*g - 27648*a^6*b
^2*c^7*d*h + 73728*a^6*b^2*c^7*e*g + 12*a^3*b^9*c^3*f*h - 1536*a^4*b^7*c^4*
e*i - 2304*a^4*b^7*c^4*f*h + 9216*a^5*b^5*c^5*e*i + 17280*a^5*b^5*c^5*f*h -
 30720*a^6*b^3*c^6*f*h + 768*a^4*b^8*c^3*g*i - 4608*a^5*b^6*c^4*g*i + 24576
*a^7*b^2*c^6*g*i))/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*
b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (x*(1
3824*a^4*c^8*e^3 + 512*a^7*c^5*i^3 - 54*b^7*c^5*d^2*e + 27*b^8*c^4*d^2*g +
13824*a^5*c^7*e^2*i + 4608*a^6*c^6*e*i^2 - 9*b^9*c^3*d^2*i - 1728*a^4*b^3*c
^5*g^3 + 64*a^4*b^6*c^2*i^3 + 384*a^5*b^4*c^3*i^3 + 768*a^6*b^2*c^4*i^3 - 2
0160*a^4*c^8*d*e*f - 6720*a^5*c^7*d*f*i - 2880*a^5*c^7*e*f*h - 960*a^6*c^6*
f*h*i + 972*a*b^5*c^6*d^2*e + 24192*a^3*b*c^8*d^2*e - 486*a*b^6*c^5*d^2*g +

6240*a^4*b*c^7*e*f^2 - 20736*a^4*b*c^7*e^2*g + 144*a*b^7*c^4*d^2*i + 8064*
a^4*b*c^7*d^2*i + 1728*a^5*b*c^6*e*h^2 + 2080*a^5*b*c^6*f^2*i - 2304*a^6*b*
c^5*g*i^2 + 576*a^6*b*c^5*h^2*i - 7344*a^2*b^3*c^7*d^2*e + 3672*a^2*b^4*c^6
*d^2*g - 6*a^2*b^5*c^5*e*f^2 - 12096*a^3*b^2*c^7*d^2*g + 192*a^3*b^3*c^6*e*
f^2 + 10368*a^4*b^2*c^6*e*g^2 - 900*a^2*b^5*c^5*d^2*i + 3*a^2*b^6*c^4*f^2*g
 + 1584*a^3*b^3*c^6*d^2*i - 96*a^3*b^4*c^5*f^2*g - 3120*a^4*b^2*c^6*f^2*g +
 1296*a^4*b^3*c^5*e*h^2 + 6912*a^4*b^2*c^6*e^2*i + 1152*a^4*b^4*c^4*e*i^2 +
 4608*a^5*b^2*c^5*e*i^2 - a^2*b^7*c^3*f^2*i + 30*a^3*b^5*c^4*f^2*i + 1104*a
^4*b^3*c^5*f^2*i - 648*a^4*b^4*c^4*g*h^2 - 864*a^5*b^2*c^5*g*h^2 + 1728*a^4
*b^4*c^4*g^2*i - 576*a^4*b^5*c^3*g*i^2 + 3456*a^5*b^2*c^5*g^2*i - 2304*a^5*
b^3*c^4*g*i^2 + 216*a^4*b^5*c^3*h^2*i + 720*a^5*b^3*c^4*h^2*i - 36*a*b^6*c^
5*d*e*f + 18*a*b^7*c^4*d*f*g + 15552*a^4*b*c^7*d*e*h + 10080*a^4*b*c^7*d*f*
g - 6*a*b^8*c^3*d*f*i + 5184*a^5*b*c^6*d*h*i - 13824*a^5*b*c^6*e*g*i + 1440
*a^5*b*c^6*f*g*h + 900*a^2*b^4*c^6*d*e*f - 4896*a^3*b^2*c^7*d*e*f - 108*a^2
*b^5*c^5*d*e*h - 450*a^2*b^5*c^5*d*f*g + 2448*a^3*b^3*c^6*d*f*g + 138*a^2*b
^6*c^4*d*f*i + 54*a^2*b^6*c^4*d*g*h - 516*a^3*b^4*c^5*d*f*i - 36*a^3*b^4*c^
5*e*f*h - 4992*a^4*b^2*c^6*d*f*i - 7776*a^4*b^2*c^6*d*g*h - 6048*a^4*b^2*c^
6*e*f*h - 18*a^2*b^7*c^3*d*h*i - 36*a^3*b^5*c^4*d*h*i + 18*a^3*b^5*c^4*f*g*
h + 2592*a^4*b^3*c^5*d*h*i - 6912*a^4*b^3*c^5*e*g*i + 3024*a^4*b^3*c^5*f*g*
h - 6*a^3*b^6*c^3*f*h*i - 1020*a^4*b^4*c^4*f*h*i - 2496*a^5*b^2*c^5*f*h*i))
/(64*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7
*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)))*root(56371445760*a^11*b^8
*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 1717986
91840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*
b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 -
 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 +
196608*a^5*b^13*c*g*i*z^2 - 46080*a^4*b^14*c*f*h*z^2 - 105984*a^3*b^15*c*d*
h*z^2 - 73728*a^2*b^16*c*d*f*z^2 + 2548039680*a^9*b^3*c^7*d*h*z^2 + 1509949
440*a^9*b^3*c^7*e*g*z^2 - 1401421824*a^8*b^5*c^6*d*h*z^2 - 1321205760*a^9*b
^2*c^8*d*f*z^2 - 754974720*a^8*b^5*c^6*e*g*z^2 + 732168192*a^7*b^6*c^6*d*f*
z^2 - 603979776*a^10*b^2*c^7*e*i*z^2 - 456130560*a^9*b^4*c^6*f*h*z^2 + 3904
63488*a^7*b^7*c^5*d*h*z^2 + 301989888*a^10*b^3*c^6*g*i*z^2 - 366280704*a^6*
b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 254017536*a^8*b^6*c^5*f*h
*z^2 - 1887436800*a^10*b*c^8*d*h*z^2 + 188743680*a^10*b^2*c^7*f*h*z^2 + 188
743680*a^7*b^7*c^5*e*g*z^2 + 125829120*a^8*b^6*c^5*e*i*z^2 - 62914560*a^8*b
^7*c^4*g*i*z^2 - 61931520*a^7*b^8*c^4*f*h*z^2 + 23592960*a^7*b^9*c^3*g*i*z^
2 - 47185920*a^7*b^8*c^4*e*i*z^2 - 3538944*a^6*b^11*c^2*g*i*z^2 + 96583680*
a^5*b^10*c^4*d*f*z^2 - 51609600*a^6*b^9*c^4*d*h*z^2 + 7077888*a^6*b^10*c^3*
e*i*z^2 + 6144000*a^6*b^10*c^3*f*h*z^2 - 393216*a^5*b^12*c^2*e*i*z^2 + 6144
0*a^5*b^12*c^2*f*h*z^2 - 23592960*a^6*b^9*c^4*e*g*z^2 + 1179648*a^5*b^11*c^
3*e*g*z^2 + 829440*a^4*b^13*c^2*d*h*z^2 + 368640*a^5*b^11*c^3*d*h*z^2 - 151
75680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 1207959552*a^10
*b*c^8*e*g*z^2 - 402653184*a^11*b*c^7*g*i*z^2 - 440401920*a^10*b*c^8*f^2*z^
2 - 188743680*a^11*b*c^7*h^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 + 524288*a^6
*b^12*c*i^2*z^2 + 46080*a^5*b^13*c*h^2*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 693

6330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a
^9*b*c^9*d^2*z^2 + 805306368*a^11*c^8*e*i*z^2 - 1509949440*a^9*b^2*c^8*e^2*
z^2 + 251658240*a^11*c^8*f*h*z^2 + 1536*a^3*b^16*f*h*z^2 + 4608*a^2*b^17*d*
h*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720
*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^
7*f^2*z^2 - 377487360*a^9*b^4*c^6*g^2*z^2 + 301989888*a^10*b^2*c^7*g^2*z^2
+ 188743680*a^8*b^6*c^5*g^2*z^2 + 141557760*a^10*b^3*c^6*h^2*z^2 - 17432576
0*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*
c^4*d^2*z^2 - 50331648*a^10*b^4*c^5*i^2*z^2 - 33554432*a^11*b^2*c^6*i^2*z^2
 + 20971520*a^9*b^6*c^4*i^2*z^2 - 47185920*a^7*b^8*c^4*g^2*z^2 - 26542080*a
^8*b^7*c^4*h^2*z^2 - 2752512*a^7*b^10*c^2*i^2*z^2 + 2621440*a^8*b^8*c^3*i^2
*z^2 + 9584640*a^7*b^9*c^3*h^2*z^2 - 2359296*a^9*b^5*c^5*h^2*z^2 - 1290240*
a^6*b^11*c^2*h^2*z^2 + 5898240*a^6*b^10*c^3*g^2*z^2 - 294912*a^5*b^12*c^2*g
^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592
960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^
2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 +
1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^
2*z^2 + 134217728*a^12*c^7*i^2*z^2 - 32768*a^5*b^14*i^2*z^2 + 2304*a^4*b^15
*h^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d
*e*f*z + 99090432*a^8*b*c^7*d*g*h*z - 3145728*a^9*b*c^6*f*h*i*z - 27648*a^4
*b^11*c*f*h*i*z + 56623104*a^8*b*c^7*d*f*i*z - 50688*a^3*b^12*c*d*h*i*z - 4
608*a^3*b^12*c*f*g*h*z - 9437184*a^8*b*c^7*e*f*h*z - 55296*a^2*b^13*c*d*f*i
*z - 13824*a^2*b^13*c*d*g*h*z + 9216*a*b^13*c^2*d*e*f*z - 4608*a*b^14*c*d*f
*g*z + 219414528*a^7*b^2*c^7*d*e*h*z - 221773824*a^6*b^3*c^7*d*e*f*z - 1097
07264*a^7*b^3*c^6*d*g*h*z + 110886912*a^6*b^4*c^6*d*f*g*z + 40108032*a^8*b^
2*c^6*d*h*i*z + 2359296*a^8*b^3*c^5*f*h*i*z - 491520*a^6*b^7*c^3*f*h*i*z +
184320*a^5*b^9*c^2*f*h*i*z - 88473600*a^6*b^4*c^6*d*e*h*z - 84934656*a^7*b^
2*c^7*d*f*g*z + 117964800*a^5*b^5*c^6*d*e*f*z - 45613056*a^7*b^3*c^6*d*f*i*
z + 44236800*a^6*b^5*c^5*d*g*h*z - 10321920*a^6*b^6*c^4*d*h*i*z + 7077888*a
^7*b^4*c^5*d*h*i*z - 5898240*a^7*b^4*c^5*f*g*h*z + 4718592*a^8*b^2*c^6*f*g*
h*z + 2949120*a^6*b^6*c^4*f*g*h*z + 2396160*a^5*b^8*c^3*d*h*i*z - 737280*a^
5*b^8*c^3*f*g*h*z + 92160*a^4*b^10*c^2*f*g*h*z - 27648*a^4*b^10*c^2*d*h*i*z
 - 58982400*a^5*b^6*c^5*d*f*g*z + 11796480*a^7*b^3*c^6*e*f*h*z + 8847360*a^
5*b^7*c^4*d*f*i*z - 6635520*a^5*b^7*c^4*d*g*h*z - 5898240*a^6*b^5*c^5*e*f*h
*z - 3809280*a^4*b^9*c^3*d*f*i*z + 2359296*a^6*b^5*c^5*d*f*i*z + 1474560*a^
5*b^7*c^4*e*f*h*z + 681984*a^3*b^11*c^2*d*f*i*z - 276480*a^4*b^9*c^3*d*g*h*
z - 184320*a^4*b^9*c^3*e*f*h*z + 179712*a^3*b^11*c^2*d*g*h*z + 9216*a^3*b^1
1*c^2*e*f*h*z + 16220160*a^4*b^8*c^4*d*f*g*z + 13271040*a^5*b^6*c^5*d*e*h*z
 - 2396160*a^3*b^10*c^3*d*f*g*z + 552960*a^4*b^8*c^4*d*e*h*z - 359424*a^3*b
^10*c^3*d*e*h*z + 175104*a^2*b^12*c^2*d*f*g*z + 27648*a^2*b^12*c^2*d*e*h*z
- 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b
^11*c^3*d*e*f*z + 346816512*a^7*b*c^8*d^2*g*z - 41472*a^5*b^10*c*h^2*i*z +
7077888*a^9*b*c^6*g*h^2*z - 11008*a^3*b^12*c*f^2*i*z - 6912*a^4*b^11*c*g*h^
2*z - 19660800*a^8*b*c^7*f^2*g*z - 768*a^2*b^13*c*f^2*g*z + 214272*a*b^13*c
^2*d^2*g*z - 428544*a*b^12*c^3*d^2*e*z - 198180864*a^8*c^8*d*e*h*z - 660602

88*a^9*c^7*d*h*i*z + 1536*a^3*b^13*f*h*i*z + 4608*a^2*b^14*d*h*i*z - 66816*
a*b^14*c*d^2*i*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d
^2*e*z - 511377408*a^6*b^3*c^7*d^2*g*z + 321159168*a^5*b^5*c^6*d^2*g*z + 22
5312768*a^7*b^2*c^7*d^2*i*z + 223395840*a^4*b^6*c^6*d^2*e*z - 111697920*a^4
*b^7*c^5*d^2*g*z + 3538944*a^9*b^2*c^5*h^2*i*z - 737280*a^7*b^6*c^3*h^2*i*z
 + 276480*a^6*b^8*c^2*h^2*i*z - 10354688*a^8*b^2*c^6*f^2*i*z - 43646976*a^6
*b^4*c^6*d^2*i*z - 8847360*a^8*b^3*c^5*g*h^2*z + 4423680*a^7*b^5*c^4*g*h^2*
z + 2048000*a^6*b^6*c^4*f^2*i*z - 1105920*a^6*b^7*c^3*g*h^2*z - 849920*a^5*
b^8*c^3*f^2*i*z + 393216*a^7*b^4*c^5*f^2*i*z + 145920*a^4*b^10*c^2*f^2*i*z
+ 138240*a^5*b^9*c^2*g*h^2*z - 32587776*a^5*b^6*c^5*d^2*i*z + 25362432*a^7*
b^3*c^6*f^2*g*z + 21657600*a^4*b^8*c^4*d^2*i*z + 17694720*a^8*b^2*c^6*e*h^2
*z - 50724864*a^7*b^2*c^7*e*f^2*z - 13271040*a^6*b^5*c^5*f^2*g*z - 8847360*
a^7*b^4*c^5*e*h^2*z - 5810688*a^3*b^10*c^3*d^2*i*z + 3563520*a^5*b^7*c^4*f^
2*g*z + 2211840*a^6*b^6*c^4*e*h^2*z + 845568*a^2*b^12*c^2*d^2*i*z - 506880*
a^4*b^9*c^3*f^2*g*z - 276480*a^5*b^8*c^3*e*h^2*z + 34560*a^3*b^11*c^2*f^2*g
*z + 13824*a^4*b^10*c^2*e*h^2*z + 26542080*a^6*b^4*c^6*e*f^2*z + 23362560*a
^3*b^9*c^4*d^2*g*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f
^2*z - 2965248*a^2*b^11*c^3*d^2*g*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a
^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*
e*z + 1536*a*b^15*d*f*i*z - 693633024*a^7*c^9*d^2*e*z - 231211008*a^8*c^8*d
^2*i*z - 4718592*a^10*c^6*h^2*i*z + 2304*a^4*b^12*h^2*i*z + 13107200*a^9*c^
7*f^2*i*z + 256*a^2*b^14*f^2*i*z - 14155776*a^9*c^7*e*h^2*z + 39321600*a^8*
c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z - 6912*b^15*c*d^2*g*z + 2304*b^16*d^2*
i*z + 737280*a^7*b*c^5*f*g*h*i - 2304*a^3*b^9*c*f*g*h*i - 6912*a^2*b^10*c*d
*g*h*i + 11059200*a^6*b*c^6*d*e*h*i + 5160960*a^6*b*c^6*d*f*g*i + 2211840*a
^6*b*c^6*e*f*g*h + 4608*a*b^10*c^2*d*e*f*i + 15482880*a^5*b*c^7*d*e*f*g - 1
3824*a*b^9*c^3*d*e*f*g - 2304*a*b^11*c*d*f*g*i + 1843200*a^6*b^3*c^4*f*g*h*
i + 783360*a^5*b^5*c^3*f*g*h*i + 18432*a^4*b^7*c^2*f*g*h*i - 5529600*a^6*b^
2*c^5*d*g*h*i - 3686400*a^6*b^2*c^5*e*f*h*i - 2211840*a^5*b^4*c^4*d*g*h*i -
 1566720*a^5*b^4*c^4*e*f*h*i + 317952*a^4*b^6*c^3*d*g*h*i - 36864*a^4*b^6*c
^3*e*f*h*i + 6912*a^3*b^8*c^2*d*g*h*i + 4608*a^3*b^8*c^2*e*f*h*i + 5160960*
a^5*b^3*c^5*d*f*g*i + 4423680*a^5*b^3*c^5*e*f*g*h + 4423680*a^5*b^3*c^5*d*e
*h*i - 635904*a^4*b^5*c^4*d*e*h*i - 354816*a^3*b^7*c^3*d*f*g*i + 322560*a^4
*b^5*c^4*d*f*g*i + 138240*a^4*b^5*c^4*e*f*g*h + 59904*a^2*b^9*c^2*d*f*g*i -
 13824*a^3*b^7*c^3*e*f*g*h - 13824*a^3*b^7*c^3*d*e*h*i + 13824*a^2*b^9*c^2*
d*e*h*i - 16588800*a^5*b^2*c^6*d*e*g*h - 10321920*a^5*b^2*c^6*d*e*f*i + 165
8880*a^4*b^4*c^5*d*e*g*h + 709632*a^3*b^6*c^4*d*e*f*i - 645120*a^4*b^4*c^5*
d*e*f*i + 124416*a^3*b^6*c^4*d*e*g*h - 119808*a^2*b^8*c^3*d*e*f*i - 41472*a
^2*b^8*c^3*d*e*g*h + 7741440*a^4*b^3*c^6*d*e*f*g - 2903040*a^3*b^5*c^5*d*e*
f*g + 387072*a^2*b^7*c^4*d*e*f*g - 3456*a^4*b^8*c*g*h^2*i - 2304*a^4*b^8*c*
f*h*i^2 + 1105920*a^7*b*c^5*e*h^2*i - 384*a^2*b^10*c*f^2*g*i - 10616832*a^6
*b*c^6*e^2*g*i - 3538944*a^7*b*c^5*e*g*i^2 + 1843200*a^7*b*c^5*d*h*i^2 + 11
52*a^3*b^9*c*d*h*i^2 - 37062144*a^5*b*c^7*d^2*f*h + 2580480*a^6*b*c^6*e*f^2
*i + 65664*a*b^10*c^2*d^2*g*i + 23224320*a^5*b*c^7*d^2*e*i - 9216*a^2*b^10*
c*d*f*i^2 - 5985792*a^6*b*c^6*d*f*h^2 + 206010*a*b^9*c^3*d^2*f*h - 131328*a

*b^9*c^3*d^2*e*i - 6300*a*b^10*c^2*d*f^2*h + 16588800*a^5*b*c^7*d*e^2*h + 3
456*a*b^10*c^2*d*f*g^2 + 435456*a*b^8*c^4*d^2*e*g + 13824*a*b^8*c^4*d*e^2*f
 - 1474560*a^7*c^6*e*f*h*i - 10321920*a^6*c^7*d*e*f*i + 1350*a*b^11*c*d*f*h
^2 - 552960*a^7*b^2*c^4*g*h^2*i - 552960*a^6*b^4*c^3*g*h^2*i - 145152*a^5*b
^6*c^2*g*h^2*i - 737280*a^7*b^2*c^4*f*h*i^2 - 568320*a^6*b^4*c^3*f*h*i^2 -
136704*a^5*b^6*c^2*f*h*i^2 - 1290240*a^6*b^2*c^5*f^2*g*i + 1105920*a^6*b^3*
c^4*e*h^2*i - 860160*a^5*b^4*c^4*f^2*g*i + 290304*a^5*b^5*c^3*e*h^2*i - 806
40*a^4*b^6*c^3*f^2*g*i + 12672*a^3*b^8*c^2*f^2*g*i + 6912*a^4*b^7*c^2*e*h^2
*i + 5308416*a^6*b^2*c^5*e*g^2*i - 5308416*a^5*b^3*c^5*e^2*g*i - 3538944*a^
6*b^3*c^4*e*g*i^2 + 2654208*a^5*b^4*c^4*e*g^2*i + 1658880*a^6*b^3*c^4*d*h*i
^2 - 1105920*a^5*b^4*c^4*f*g^2*h - 884736*a^5*b^5*c^3*e*g*i^2 - 552960*a^6*
b^2*c^5*f*g^2*h + 262656*a^5*b^5*c^3*d*h*i^2 - 55296*a^4*b^7*c^2*d*h*i^2 -
34560*a^4*b^6*c^3*f*g^2*h + 3456*a^3*b^8*c^2*f*g^2*h - 11612160*a^5*b^2*c^6
*d^2*g*i + 1720320*a^5*b^3*c^5*e*f^2*i - 1658880*a^6*b^2*c^5*e*g*h^2 + 1596
672*a^3*b^6*c^4*d^2*g*i - 829440*a^5*b^4*c^4*e*g*h^2 - 508032*a^2*b^8*c^3*d
^2*g*i + 161280*a^4*b^5*c^4*e*f^2*i - 25344*a^3*b^7*c^3*e*f^2*i - 20736*a^4
*b^6*c^3*e*g*h^2 + 768*a^2*b^9*c^2*e*f^2*i - 4423680*a^5*b^2*c^6*e^2*f*h +
4147200*a^5*b^3*c^5*d*g^2*h - 2580480*a^6*b^2*c^5*d*f*i^2 - 967680*a^5*b^4*
c^4*d*f*i^2 - 414720*a^4*b^5*c^4*d*g^2*h - 138240*a^4*b^4*c^5*e^2*f*h + 645
12*a^4*b^6*c^3*d*f*i^2 + 39168*a^3*b^8*c^2*d*f*i^2 - 31104*a^3*b^7*c^3*d*g^
2*h + 13824*a^3*b^6*c^4*e^2*f*h + 10368*a^2*b^9*c^2*d*g^2*h + 15630336*a^5*
b^2*c^6*d*f^2*h - 14459904*a^4*b^3*c^6*d^2*f*h + 9630144*a^3*b^5*c^5*d^2*f*
h - 8764416*a^5*b^3*c^5*d*f*h^2 - 3870720*a^5*b^2*c^6*e*f^2*g - 3193344*a^3
*b^5*c^5*d^2*e*i + 2867328*a^4*b^4*c^5*d*f^2*h - 2095200*a^2*b^7*c^4*d^2*f*
h - 1414080*a^3*b^6*c^4*d*f^2*h - 34836480*a^4*b^2*c^7*d^2*e*g + 1016064*a^
2*b^7*c^4*d^2*e*i - 645120*a^4*b^4*c^5*e*f^2*g + 306720*a^3*b^7*c^3*d*f*h^2
 + 197820*a^2*b^8*c^3*d*f^2*h + 146880*a^4*b^5*c^4*d*f*h^2 + 80640*a^3*b^6*
c^4*e*f^2*g - 55350*a^2*b^9*c^2*d*f*h^2 - 2304*a^2*b^8*c^3*e*f^2*g - 387072
0*a^5*b^2*c^6*d*f*g^2 - 1935360*a^4*b^4*c^5*d*f*g^2 - 1658880*a^4*b^3*c^6*d
*e^2*h + 725760*a^3*b^6*c^4*d*f*g^2 + 17418240*a^3*b^4*c^6*d^2*e*g - 124416
*a^3*b^5*c^5*d*e^2*h - 96768*a^2*b^8*c^3*d*f*g^2 + 41472*a^2*b^7*c^4*d*e^2*
h - 3919104*a^2*b^6*c^5*d^2*e*g - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3
*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 184320*a^8*b*c^4*h^2*i^2 +
25344*a^5*b^7*c*h^2*i^2 - 884736*a^6*b^3*c^4*g^3*i - 589824*a^7*b^3*c^3*g*i
^3 - 442368*a^5*b^5*c^3*g^3*i - 294912*a^6*b^5*c^2*g*i^3 + 430080*a^7*b*c^5
*f^2*i^2 - 1984*a^3*b^9*c*f^2*i^2 + 3538944*a^5*b^2*c^6*e^3*i - 1648128*a^5
*b^3*c^5*f^3*h + 1179648*a^7*b^2*c^4*e*i^3 - 898560*a^6*b^3*c^4*f*h^3 + 589
824*a^6*b^4*c^3*e*i^3 - 354240*a^5*b^5*c^3*f*h^3 - 354240*a^4*b^5*c^4*f^3*h
 + 98304*a^5*b^6*c^2*e*i^3 + 43680*a^3*b^7*c^3*f^3*h - 21600*a^4*b^7*c^2*f*
h^3 - 1050*a^2*b^9*c^2*f^3*h + 225*a^2*b^10*c*f^2*h^2 + 3870720*a^6*b*c^6*d
^2*i^2 + 1658880*a^6*b*c^6*e^2*h^2 + 16547328*a^4*b^2*c^7*d^3*h - 12306816*
a^3*b^4*c^6*d^3*h + 37310976*a^3*b^3*c^7*d^3*f + 3037824*a^2*b^6*c^5*d^3*h
 - 2654208*a^5*b^3*c^5*e*g^3 + 1949184*a^6*b^2*c^5*d*h^3 + 1296000*a^5*b^4*c
^4*d*h^3 - 155520*a^4*b^6*c^3*d*h^3 - 40500*a*b^10*c^2*d^2*h^2 - 8100*a^3*b
^8*c^2*d*h^3 + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 108

864*a*b^9*c^3*d^2*g^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f
^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*
c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 245760*a^8*c^5*f*h*i^2 + 384*a^3*b^1
0*f*h*i^2 + 1152*a^2*b^11*d*h*i^2 - 2211840*a^6*c^7*e^2*f*h - 1720320*a^7*c
^6*d*f*i^2 - 9450*b^11*c^2*d^2*f*h + 6912*b^11*c^2*d^2*e*i + 1612800*a^6*c^
7*d*f^2*h - 393216*a^8*b*c^4*g*i^3 - 49152*a^5*b^7*c*g*i^3 - 20736*b^10*c^3
*d^2*e*g - 75188736*a^4*b*c^8*d^3*f - 883200*a^6*b*c^6*f^3*h - 317952*a^7*b
*c^5*f*h^3 + 1350*a^3*b^9*c*f*h^3 - 15482880*a^5*c^8*d*e^2*f - 9792*a*b^11*
c*d^2*i^2 - 10616832*a^5*b*c^7*e^3*g - 345060*a*b^8*c^4*d^3*h + 4050*a^2*b^
10*c*d*h^3 - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*
c^3*d*f^3 + 276480*a^7*b^3*c^3*h^2*i^2 + 140544*a^6*b^5*c^2*h^2*i^2 + 88473
6*a^7*b^2*c^4*g^2*i^2 + 884736*a^6*b^4*c^3*g^2*i^2 + 221184*a^5*b^6*c^2*g^2
*i^2 + 501760*a^6*b^3*c^4*f^2*i^2 + 414720*a^6*b^3*c^4*g^2*h^2 + 207360*a^5
*b^5*c^3*g^2*h^2 + 170240*a^5*b^5*c^3*f^2*i^2 + 9216*a^4*b^7*c^2*f^2*i^2 +
5184*a^4*b^7*c^2*g^2*h^2 + 3538944*a^6*b^2*c^5*e^2*i^2 + 1684224*a^6*b^2*c^
5*f^2*h^2 + 1264320*a^5*b^4*c^4*f^2*h^2 + 884736*a^5*b^4*c^4*e^2*i^2 + 1267
20*a^4*b^6*c^3*f^2*h^2 - 13950*a^3*b^8*c^2*f^2*h^2 + 1935360*a^5*b^3*c^5*d^
2*i^2 + 967680*a^5*b^3*c^5*f^2*g^2 + 829440*a^5*b^3*c^5*e^2*h^2 - 532224*a^
4*b^5*c^4*d^2*i^2 + 161280*a^4*b^5*c^4*f^2*g^2 - 96768*a^3*b^7*c^3*d^2*i^2
+ 62784*a^2*b^9*c^2*d^2*i^2 + 20736*a^4*b^5*c^4*e^2*h^2 - 20160*a^3*b^7*c^3
*f^2*g^2 + 576*a^2*b^9*c^2*f^2*g^2 + 11487744*a^5*b^2*c^6*d^2*h^2 + 7962624
*a^5*b^2*c^6*e^2*g^2 + 35525376*a^4*b^2*c^7*d^2*f^2 - 1412640*a^3*b^6*c^4*d
^2*h^2 + 461376*a^4*b^4*c^5*d^2*h^2 + 375030*a^2*b^8*c^3*d^2*h^2 + 8709120*
a^4*b^3*c^6*d^2*g^2 - 4354560*a^3*b^5*c^5*d^2*g^2 + 979776*a^2*b^7*c^4*d^2*
g^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7
*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 -
17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 - 3456*b^12*c*d
^2*g*i + 384*a*b^12*d*f*i^2 + 576*a^4*b^9*h^2*i^2 + 3538944*a^7*c^6*e^2*i^2
+ 115200*a^7*c^6*f^2*h^2 + 64*a^2*b^11*f^2*i^2 + 6096384*a^6*c^7*d^2*h^2 +
5184*b^11*c^2*d^2*g^2 + 131072*a^8*b^2*c^3*i^4 + 98304*a^7*b^4*c^2*i^4 + 1
1025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 142560*a^6*b^4*c^3*h^4 +
103680*a^7*b^2*c^4*h^4 + 32400*a^5*b^6*c^2*h^4 + 20736*b^9*c^4*d^2*e^2 + 33
1776*a^5*b^4*c^4*g^4 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43
120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 644
6304*a^2*b^4*c^7*d^4 + 7077888*a^6*c^7*e^3*i + 786432*a^8*c^5*e*i^3 + 28449
792*a^5*c^8*d^3*h + 17010*b^10*c^3*d^3*h + 2025*b^12*c*d^2*h^2 + 580608*a^7
*c^6*d*h^3 - 39690*b^9*c^4*d^3*f + 32768*a^6*b^6*c*i^4 + 2025*a^4*b^8*c*h^4
- 734832*a*b^6*c^6*d^4 + 576*b^13*d^2*i^2 + 65536*a^9*c^4*i^4 + 20736*a^8*
c^5*h^4 + 4096*a^5*b^8*i^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 53
08416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, l), l, 1, 4)

3.57 $\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx$

Optimal result

Integrand size = 55, antiderivative size = 1150

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx$$

$$= -\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+\left(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al)\right)x^2}{4c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$-\frac{x\left(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+\left(ab^2ck+2ac^2(cf-ak)-ab^3m-bc(c^2d+\cdots\right)\right)}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{\frac{b^3j}{c}+2b(3ce+aj)-16a^2l-\frac{b^4l}{c^2}-b^2\left(3g-\frac{5al}{c}\right)+2(6c^2e-3bcg+b^2j+2acj-3abl)\,x^2}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{x\left(4a^2bc^2(2cf+ak)+ab^3c(cf+2ak)-ab^2c(25c^2d+7ach-11a^2m)+4a^2c^2(7c^2d+ach-9a^2m)+b\cdots\right)}{8a^2c^2\left(b^2-4a\cdots\right)}$$

$$+\frac{\left(ab^2c(cf+3ak)+4a^2c^2(5cf+3ak)+b^3(3c^2d+a^2m)-4abc(6c^2d+3ach+4a^2m)+\frac{ab^3c(cf-3ak)-4a^2b\cdots}{\cdots}\right)}{8\sqrt{2}a^2c^{3/2}\left(b^2-4ac\right)^2\sqrt{b-\cdots}}$$

$$+\frac{\left(ab^2c(cf+3ak)+4a^2c^2(5cf+3ak)+b^3(3c^2d+a^2m)-4abc(6c^2d+3ach+4a^2m)-\frac{ab^3c(cf-3ak)-4a^2b\cdots}{\cdots}\right)}{8\sqrt{2}a^2c^{3/2}\left(b^2-4ac\right)^2\sqrt{b-\cdots}}$$

$$-\frac{(6c^2e-3bcg+b^2j+2acj-3abl)\,\mathrm{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}$$

```
[Out] 1/4*(-b*c*(a*j+c*e)+a*b^2*l+2*a*c*(-a*l+c*g)-(2*c^3*e-c^2*(2*a*j+b*g)-b^3*l
+b*c*(3*a*l+b*j))*x^2)/c^2/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2-1/4*x*(a*b*c*(a*k
+c*f)-b^2*(a^2*m+c^2*d)+2*a*c*(a^2*m-a*c*h+c^2*d)+(a*b^2*c*k+2*a*c^2*(-a*k+
c*f)-a*b^3*m-b*c*(-3*a^2*m+a*c*h+c^2*d))*x^2)/a/c^2/(-4*a*c+b^2)/(c*x^4+b*x
^2+a)^2+1/4*(b^3*j/c+2*b*(a*j+3*c*e)-16*a^2*l-b^4*l/c^2-b^2*(3*g-5*a*l/c)+2
```

```
*(-3*a*b*l+2*a*c*j+b^2*j-3*b*c*g+6*c^2*e)*x^2)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+
a)+1/8*x*(4*a^2*b*c^2*(a*k+2*c*f)+a*b^3*c*(2*a*k+c*f)-a*b^2*c*(-11*a^2*m+7*
a*c*h+25*c^2*d)+4*a^2*c^2*(-9*a^2*m+a*c*h+7*c^2*d)+b^4*(-2*a^2*m+3*c^2*d)+c
*(a*b^2*c*(3*a*k+c*f)+4*a^2*c^2*(3*a*k+5*c*f)+b^3*(a^2*m+3*c^2*d)-4*a*b*c*(
4*a^2*m+3*a*c*h+6*c^2*d))*x^2)/a^2/c^2/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)-(-3*a
*b*l+2*a*c*j+b^2*j-3*b*c*g+6*c^2*e)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))
/(-4*a*c+b^2)^(5/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1
/2))*(a*b^2*c*(3*a*k+c*f)+4*a^2*c^2*(3*a*k+5*c*f)+b^3*(a^2*m+3*c^2*d)-4*a*b
*c*(4*a^2*m+3*a*c*h+6*c^2*d)+(a*b^3*c*(-3*a*k+c*f)-4*a^2*b*c^2*(9*a*k+13*c*
f)-6*a*b^2*c*(-3*a^2*m-3*a*c*h+5*c^2*d)+b^4*(-a^2*m+3*c^2*d)+8*a^2*c^2*(5*a
^2*m+3*a*c*h+21*c^2*d))/(-4*a*c+b^2)^(1/2))/a^2/c^(3/2)/(-4*a*c+b^2)^2*2^(1
/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b
^2)^(1/2))^(1/2))*(a*b^2*c*(3*a*k+c*f)+4*a^2*c^2*(3*a*k+5*c*f)+b^3*(a^2*m+3
*c^2*d)-4*a*b*c*(4*a^2*m+3*a*c*h+6*c^2*d)+(-a*b^3*c*(-3*a*k+c*f)+4*a^2*b*c^
2*(9*a*k+13*c*f)+6*a*b^2*c*(-3*a^2*m-3*a*c*h+5*c^2*d)-b^4*(-a^2*m+3*c^2*d)-
8*a^2*c^2*(5*a^2*m+3*a*c*h+21*c^2*d))/(-4*a*c+b^2)^(1/2))/a^2/c^(3/2)/(-4*a
*c+b^2)^2*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 5.06 (sec) , antiderivative size = 1144, normalized size of antiderivative = 0.99, number of steps used = 11, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.164$, Rules used = {1687, 1692, 1180, 211, 1677, 1674, 652, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{-\frac{lb^4}{c^2}+\frac{jb^3}{c}-\left(3g-\frac{5al}{c}\right)b^2+2(3ce+aj)b+2(jb^2-3cgb-3alb+6c^2e+2acj)\,x^2-16a^2l}{4\,(b^2-4ac)^2\,(cx^4+bx^2+a)}$$

$$+\frac{\left(\left(\frac{ma^2}{c}+3cd\right)b^3+a(cf+3ak)b^2-4a(4ma^2+3cha+6c^2d)\,b+4a^2c(5cf+3ak)+\frac{(3c^2d-a^2m)b^4+ac(cf-3a}{}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2\,\sqrt{b-\sqrt{}}}$$

$$+\frac{\left(\left(\frac{ma^2}{c}+3cd\right)b^3+a(cf+3ak)b^2-4a(4ma^2+3cha+6c^2d)\,b+4a^2c(5cf+3ak)-\frac{(3c^2d-a^2m)b^4+ac(cf-3a}{}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2\,\sqrt{b+\sqrt{}}}$$

$$-\frac{(jb^2-3cgb-3alb+6c^2e+2acj)\,\text{arctanh}\left(\frac{2cx^2+b}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

$$+\frac{x\left(\left(3cd-\frac{2a^2m}{c}\right)b^4+a(cf+2ak)b^3-a(-11ma^2+7cha+25c^2d)\,b^2+4a^2c(2cf+ak)b+((ma^2+3c^2d\right.}{8a^2c\,(b^2-4ac}$$

$$-\frac{-alb^2+c(ce+aj)b+(-lb^3+c(bj+3al)b+2c^3e-c^2(bg+2aj))\,x^2-2ac(cg-al)}{4c^2\,(b^2-4ac)\,(cx^4+bx^2+a)^2}$$

$$-\frac{x(-((ma^2+c^2d)\,b^2)+ac(cf+ak)b+(-amb^3+ackb^2-c(-3ma^2+cha+c^2d)\,b+2ac^2(cf-ak))\,x^2}{4ac^2\,(b^2-4ac)\,(cx^4+bx^2+a)^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4)^3,x]
```

```
[Out] -1/4*(b*c*(c*e + a*j) - a*b^2*l - 2*a*c*(c*g - a*l) + (2*c^3*e - c^2*(b*g +
 2*a*j) - b^3*l + b*c*(b*j + 3*a*l))*x^2)/(c^2*(b^2 - 4*a*c)*(a + b*x^2 + c
*x^4)^2) - (x*(a*b*c*(c*f + a*k) - b^2*(c^2*d + a^2*m) + 2*a*c*(c^2*d - a*c
*h + a^2*m) + (a*b^2*c*k + 2*a*c^2*(c*f - a*k) - a*b^3*m - b*c*(c^2*d + a*c
*h - 3*a^2*m))*x^2))/(4*a*c^2*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + ((b^3*
j)/c + 2*b*(3*c*e + a*j) - 16*a^2*l - (b^4*l)/c^2 - b^2*(3*g - (5*a*l)/c) +
 2*(6*c^2*e - 3*b*c*g + b^2*j + 2*a*c*j - 3*a*b*l)*x^2)/(4*(b^2 - 4*a*c)^2*
(a + b*x^2 + c*x^4)) + (x*(4*a^2*b*c*(2*c*f + a*k) + a*b^3*(c*f + 2*a*k) -
a*b^2*(25*c^2*d + 7*a*c*h - 11*a^2*m) + 4*a^2*c*(7*c^2*d + a*c*h - 9*a^2*m)
 + b^4*(3*c*d - (2*a^2*m)/c) + (a*b^2*c*(c*f + 3*a*k) + 4*a^2*c^2*(5*c*f +
3*a*k) + b^3*(3*c^2*d + a^2*m) - 4*a*b*c*(6*c^2*d + 3*a*c*h + 4*a^2*m))*x^2
))/(8*a^2*c*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + ((a*b^2*(c*f + 3*a*k) +
4*a^2*c*(5*c*f + 3*a*k) - 4*a*b*(6*c^2*d + 3*a*c*h + 4*a^2*m) + b^3*(3*c*d
+ (a^2*m)/c) + (a*b^3*c*(c*f - 3*a*k) - 4*a^2*b*c^2*(13*c*f + 9*a*k) - 6*a*
b^2*c*(5*c^2*d - 3*a*c*h - 3*a^2*m) + b^4*(3*c^2*d - a^2*m) + 8*a^2*c^2*(21
*c^2*d + 3*a*c*h + 5*a^2*m))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(Sqrt[2]*Sqrt[c]
*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*Sqrt[c]*(b^2 - 4*a*c)^2*Sq
rt[b - Sqrt[b^2 - 4*a*c]]) + ((a*b^2*(c*f + 3*a*k) + 4*a^2*c*(5*c*f + 3*a*k
) - 4*a*b*(6*c^2*d + 3*a*c*h + 4*a^2*m) + b^3*(3*c*d + (a^2*m)/c) - (a*b^3*
c*(c*f - 3*a*k) - 4*a^2*b*c^2*(13*c*f + 9*a*k) - 6*a*b^2*c*(5*c^2*d - 3*a*c
*h - 3*a^2*m) + b^4*(3*c^2*d - a^2*m) + 8*a^2*c^2*(21*c^2*d + 3*a*c*h + 5*a
^2*m))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2
- 4*a*c]]])/(8*Sqrt[2]*a^2*Sqrt[c]*(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*
c]]) - ((6*c^2*e - 3*b*c*g + b^2*j + 2*a*c*j - 3*a*b*l)*ArcTanh[(b + 2*c*x^
2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
```

```
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{x(e+gx^2+jx^4+lx^6)}{(a+bx^2+cx^4)^3}\,dx + \int \frac{d+fx^2+hx^4+kx^6+mx^8}{(a+bx^2+cx^4)^3}\,dx$$

$$= -\frac{x(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+(ab^2ck+2ac^2(cf-ak)-ab^3m-\cdots}{4ac^2(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+jx^2+lx^3}{(a+bx+cx^2)^3}\,dx, x, x^2\right) - \frac{\int \frac{-\frac{abc(cf+ak)+b^2\left(3c^2d-a^2m\right)-2ac\left(7c^2d+ach-a^2m\right)}{c^2}+\frac{\left(ab^2ck+2ac^2(5cf+3ak)-ab^3m-bc\left(5c^2d+5ach+a^2m\right)\right)x^2}{c^2}+4a\left(4a-\frac{b^2}{c}\right)mx^4}{(a+bx^2+cx^4)^2}\,dx \cdots}{4a(b^2-4ac)}$$

$$= -\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al))x^2}{4c^2(b^2-4ac)(a+bx^2+cx^4)^2} - \frac{x(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+(ab^2ck+2ac^2(cf-ak)-ab^3m-\cdots}{4ac^2(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x\left(4a^2bc(2cf+ak)+ab^3(cf+2ak)-ab^2(25c^2d+7ach-11a^2m)+4a^2c(7c^2d+ach-9a^2m)\cdots\right.}{8a^2c(b^2\cdots} + \frac{\int \frac{3b^4d+ab^3f-4a^2b(4cf+3ak)+4a^2\left(21c^2d+3ach+5a^2m\right)-ab^2\left(27cd-3ah-\frac{a^2m}{c}\right)+\frac{\left(ab^2c(cf+3ak)+4a^2c^2(5cf+3ak)+b^3\left(3c^2d+a^2m\cdots\right.\right.}{c}\cdots}{a+bx^2+cx^4}\cdots}{8a^2(b^2-4ac)^2} - \frac{\text{Subst}\left(\int \frac{6ce-3bg+2aj-\frac{b^3l}{c^2}+\frac{b(bj+al)}{c}+2\left(4a-\frac{b^2}{c}\right)lx}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{4(b^2-4ac)}$$

$$= -\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+\left(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al)\right)x^2}{4c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$-\frac{x\left(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+\left(ab^2ck+2ac^2(cf-ak)-ab^3m-bc\right.\right.}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{\frac{b^3j}{c}+2b(3ce+aj)-16a^2l-\frac{b^4l}{c^2}-b^2\left(3g-\frac{5al}{c}\right)+2\left(6c^2e-3bcg+b^2j+2acj-3abl\right)x^2}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{x\Big(4a^2bc(2cf+ak)+ab^3(cf+2ak)-ab^2(25c^2d+7ach-11a^2m)+4a^2c(7c^2d+ach-9a^2m)}{8a^2c\left(b^2-\right.}$$

$$+\frac{\left(6c^2e-3bcg+b^2j+2acj-3abl\right)\text{Subst}\left(\int\frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)^2}$$

$$+\frac{\left(ab^2(cf+3ak)+4a^2c(5cf+3ak)-4ab(6c^2d+3ach+4a^2m)+b^3\left(3cd+\frac{a^2m}{c}\right)-\frac{ab^3c(cf-3ak)-}{}\right.}{16a^2\left(b^2-\right.}$$

$$+\frac{\left(ab^2(cf+3ak)+4a^2c(5cf+3ak)-4ab(6c^2d+3ach+4a^2m)+b^3\left(3cd+\frac{a^2m}{c}\right)+\frac{ab^3c(cf-3ak)-}{}\right.}{16a^2\left(b^2-\right.}$$

$$= -\frac{bc(ce+aj)-ab^2l-2ac(cg-al)+\left(2c^3e-c^2(bg+2aj)-b^3l+bc(bj+3al)\right)x^2}{4c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$-\frac{x\left(abc(cf+ak)-b^2(c^2d+a^2m)+2ac(c^2d-ach+a^2m)+\left(ab^2ck+2ac^2(cf-ak)-ab^3m-bc\right.\right.}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{\frac{b^3j}{c}+2b(3ce+aj)-16a^2l-\frac{b^4l}{c^2}-b^2\left(3g-\frac{5al}{c}\right)+2\left(6c^2e-3bcg+b^2j+2acj-3abl\right)x^2}{4\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{x\Big(4a^2bc(2cf+ak)+ab^3(cf+2ak)-ab^2(25c^2d+7ach-11a^2m)+4a^2c(7c^2d+ach-9a^2m)}{8a^2c\left(b^2-\right.}$$

$$+\frac{\left(ab^2(cf+3ak)+4a^2c(5cf+3ak)-4ab(6c^2d+3ach+4a^2m)+b^3\left(3cd+\frac{a^2m}{c}\right)+\frac{ab^3c(cf-3ak)-}{}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2}$$

$$+\frac{\left(ab^2(cf+3ak)+4a^2c(5cf+3ak)-4ab(6c^2d+3ach+4a^2m)+b^3\left(3cd+\frac{a^2m}{c}\right)-\frac{ab^3c(cf-3ak)-}{}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2}$$

$$-\frac{\left(6c^2e-3bcg+b^2j+2acj-3abl\right)\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{\left(b^2-4ac\right)^2}$$

$$= -\frac{bc(ce+aj) - ab^2l - 2ac(cg-al) + (2c^3e - c^2(bg+2aj) - b^3l + bc(bj+3al))\,x^2}{4c^2\,(b^2-4ac)\,(a+bx^2+cx^4)^2}$$

$$- \frac{x(abc(cf+ak) - b^2(c^2d+a^2m) + 2ac(c^2d - ach + a^2m) + (ab^2ck + 2ac^2(cf-ak) - ab^3m -}{4ac^2\,(b^2-4ac)\,(a+bx^2+cx^4)^2}$$

$$+ \frac{\frac{b^3j}{c} + 2b(3ce+aj) - 16a^2l - \frac{b^4l}{c^2} - b^2\left(3g - \frac{5al}{c}\right) + 2(6c^2e - 3bcg + b^2j + 2acj - 3abl)\,x^2}{4\,(b^2-4ac)^2\,(a+bx^2+cx^4)}$$

$$+ \frac{x\Big(4a^2bc(2cf+ak) + ab^3(cf+2ak) - ab^2(25c^2d + 7ach - 11a^2m) + 4a^2c(7c^2d + ach - 9a^2m)}{8a^2c\,(b^2}$$

$$+ \frac{\left(ab^2(cf+3ak) + 4a^2c(5cf+3ak) - 4ab(6c^2d+3ach+4a^2m) + b^3\left(3cd + \frac{a^2m}{c}\right) + \frac{ab^3c(cf-3ak)}{}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2}$$

$$+ \frac{\left(ab^2(cf+3ak) + 4a^2c(5cf+3ak) - 4ab(6c^2d+3ach+4a^2m) + b^3\left(3cd + \frac{a^2m}{c}\right) - \frac{ab^3c(cf-3ak)}{}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2}$$

$$- \frac{(6c^2e - 3bcg + b^2j + 2acj - 3abl)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

Mathematica [A] (verified)

Time = 6.85 (sec) , antiderivative size = 1590, normalized size of antiderivative = 1.38

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx$$

$$= \frac{abc^2e - 2a^2c^2g + a^2bcj - a^2b^2l + 2a^3cl - b^2c^2dx + 2ac^3dx + abc^2fx - 2a^2c^2hx + a^2bckx - a^2b^2mx + 2a}{}$$

$$+ \frac{12a^2bc^3e - 6a^2b^2c^2g + 2a^2b^3cj + 4a^3bc^2j - 2a^2b^4l + 10a^3b^2cl - 32a^4c^2l + 3b^4c^2dx - 25ab^2c^3dx + 28a^2c}{}$$

$$+ \frac{\left(3b^4c^2d - 30ab^2c^3d + 168a^2c^4d + 3b^3c^2\sqrt{b^2-4ac}d - 24abc^3\sqrt{b^2-4ac}d + ab^3c^2f - 52a^2bc^3f + ab^2c^2\sqrt{}\right.}{}$$

$$+ \frac{\left(-3b^4c^2d + 30ab^2c^3d - 168a^2c^4d + 3b^3c^2\sqrt{b^2-4ac}d - 24abc^3\sqrt{b^2-4ac}d - ab^3c^2f + 52a^2bc^3f + ab^2c^2\right.}{}$$

$$+ \frac{(6c^2e - 3bcg + b^2j + 2acj - 3abl)\log\left(-b + \sqrt{b^2-4ac} - 2cx^2\right)}{2\,(b^2-4ac)^{5/2}}$$

$$+ \frac{(-6c^2e + 3bcg - b^2j - 2acj + 3abl)\log\left(b + \sqrt{b^2-4ac} + 2cx^2\right)}{2\,(b^2-4ac)^{5/2}}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)
/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] (a*b*c^2*e - 2*a^2*c^2*g + a^2*b*c*j - a^2*b^2*l + 2*a^3*c*l - b^2*c^2*d*x
+ 2*a*c^3*d*x + a*b*c^2*f*x - 2*a^2*c^2*h*x + a^2*b*c*k*x - a^2*b^2*m*x + 2
*a^3*c*m*x + 2*a*c^3*e*x^2 - a*b*c^2*g*x^2 + a*b^2*c*j*x^2 - 2*a^2*c^2*j*x^
2 - a*b^3*l*x^2 + 3*a^2*b*c*l*x^2 - b*c^3*d*x^3 + 2*a*c^3*f*x^3 - a*b*c^2*h
*x^3 + a*b^2*c*k*x^3 - 2*a^2*c^2*k*x^3 - a*b^3*m*x^3 + 3*a^2*b*c*m*x^3)/(4*
a*c^2*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (12*a^2*b*c^3*e - 6*a^2*b^2*c
^2*g + 2*a^2*b^3*c*j + 4*a^3*b*c^2*j - 2*a^2*b^4*l + 10*a^3*b^2*c*l - 32*a^
4*c^2*l + 3*b^4*c^2*d*x - 25*a*b^2*c^3*d*x + 28*a^2*c^4*d*x + a*b^3*c^2*f*x
 + 8*a^2*b*c^3*f*x - 7*a^2*b^2*c^2*h*x + 4*a^3*c^3*h*x + 2*a^2*b^3*c*k*x +
4*a^3*b*c^2*k*x - 2*a^2*b^4*m*x + 11*a^3*b^2*c*m*x - 36*a^4*c^2*m*x + 24*a^
2*c^4*e*x^2 - 12*a^2*b*c^3*g*x^2 + 4*a^2*b^2*c^2*j*x^2 + 8*a^3*c^3*j*x^2 -
12*a^3*b*c^2*l*x^2 + 3*b^3*c^3*d*x^3 - 24*a*b*c^4*d*x^3 + a*b^2*c^3*f*x^3 +
 20*a^2*c^4*f*x^3 - 12*a^2*b*c^3*h*x^3 + 3*a^2*b^2*c^2*k*x^3 + 12*a^3*c^3*k
*x^3 + a^2*b^3*c*m*x^3 - 16*a^3*b*c^2*m*x^3)/(8*a^2*c^2*(-b^2 + 4*a*c)^2*(a
 + b*x^2 + c*x^4)) + ((3*b^4*c^2*d - 30*a*b^2*c^3*d + 168*a^2*c^4*d + 3*b^3
*c^2*Sqrt[b^2 - 4*a*c]*d - 24*a*b*c^3*Sqrt[b^2 - 4*a*c]*d + a*b^3*c^2*f - 5
2*a^2*b*c^3*f + a*b^2*c^2*Sqrt[b^2 - 4*a*c]*f + 20*a^2*c^3*Sqrt[b^2 - 4*a*c
]*f + 18*a^2*b^2*c^2*h + 24*a^3*c^3*h - 12*a^2*b*c^2*Sqrt[b^2 - 4*a*c]*h -
3*a^2*b^3*c*k - 36*a^3*b*c^2*k + 3*a^2*b^2*c*Sqrt[b^2 - 4*a*c]*k + 12*a^3*c
^2*Sqrt[b^2 - 4*a*c]*k - a^2*b^4*m + 18*a^3*b^2*c*m + 40*a^4*c^2*m + a^2*b^
3*Sqrt[b^2 - 4*a*c]*m - 16*a^3*b*c*Sqrt[b^2 - 4*a*c]*m)*ArcTan[(Sqrt[2]*Sqr
t[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*c^(3/2)*(b^2 - 4*a*c)^
(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((-3*b^4*c^2*d + 30*a*b^2*c^3*d - 168*
a^2*c^4*d + 3*b^3*c^2*Sqrt[b^2 - 4*a*c]*d - 24*a*b*c^3*Sqrt[b^2 - 4*a*c]*d
- a*b^3*c^2*f + 52*a^2*b*c^3*f + a*b^2*c^2*Sqrt[b^2 - 4*a*c]*f + 20*a^2*c^3
*Sqrt[b^2 - 4*a*c]*f - 18*a^2*b^2*c^2*h - 24*a^3*c^3*h - 12*a^2*b*c^2*Sqrt[
b^2 - 4*a*c]*h + 3*a^2*b^3*c*k + 36*a^3*b*c^2*k + 3*a^2*b^2*c*Sqrt[b^2 - 4*
a*c]*k + 12*a^3*c^2*Sqrt[b^2 - 4*a*c]*k + a^2*b^4*m - 18*a^3*b^2*c*m - 40*a
^4*c^2*m + a^2*b^3*Sqrt[b^2 - 4*a*c]*m - 16*a^3*b*c*Sqrt[b^2 - 4*a*c]*m)*Ar
cTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*c^(3/
2)*(b^2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((6*c^2*e - 3*b*c*g +
 b^2*j + 2*a*c*j - 3*a*b*l)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(2*(b^2
- 4*a*c)^(5/2)) + ((-6*c^2*e + 3*b*c*g - b^2*j - 2*a*c*j + 3*a*b*l)*Log[b +
 Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(2*(b^2 - 4*a*c)^(5/2))
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 1.39 (sec) , antiderivative size = 1167, normalized size of antiderivative = 1.01

method	result	size
risch	Expression too large to display	1167
default	Expression too large to display	1987

```
[In] int((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] (-1/8*(16*a^3*b*c*m-12*a^3*c^2*k-a^2*b^3*m-3*a^2*b^2*c*k+12*a^2*b*c^2*h-20*
a^2*c^3*f-a*b^2*c^2*f+24*a*b*c^3*d-3*b^3*c^2*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b
^4)*x^7-1/2*c*(3*a*b*l-2*a*c*j-b^2*j+3*b*c*g-6*c^2*e)/(16*a^2*c^2-8*a*b^2*c
+b^4)*x^6-1/8/a^2*(36*a^4*c^2*m+5*a^3*b^2*c*m-16*a^3*b*c^2*k-4*a^3*c^3*h+a^
2*b^4*m-5*a^2*b^3*c*k+19*a^2*b^2*c^2*h-28*a^2*b*c^3*f-28*a^2*c^4*d-2*a*b^3*
c^2*f+49*a*b^2*c^3*d-6*b^4*c^2*d)/(16*a^2*c^2-8*a*b^2*c+b^4)/c*x^5-1/4*(16*
a^2*c^2*l+a*b^2*c*l-6*a*b*c^2*j+b^4*l-3*b^3*c*j+9*b^2*c^2*g-18*b*c^3*e)/(16
*a^2*c^2-8*a*b^2*c+b^4)/c*x^4-1/8/c*(28*a^4*b*c*m+4*a^4*c^2*k+2*a^3*b^3*m-1
9*a^3*b^2*c*k+16*a^3*b*c^2*h-36*a^3*c^3*f+5*a^2*b^3*c*h-5*a^2*b^2*c^2*f+4*a
^2*b*c^3*d-a*b^4*c*f+20*a*b^3*c^2*d-3*b^5*c*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^
4)*x^3-1/2/c*(5*a^2*b*c*l+2*a^2*c^2*j+a*b^3*l-5*a*b^2*c*j+5*a*b*c^2*g-10*a*
c^3*e+b^3*c*g-2*b^2*c^2*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2-1/8*(20*a^4*c*m+a
^3*b^2*m-12*a^3*b*c*k+12*a^3*c^2*h+3*a^2*b^2*c*h-16*a^2*b*c^2*f-44*a^2*c^3*
d+a*b^3*c*f+37*a*b^2*c^2*d-5*b^4*c*d)/(16*a^2*c^2-8*a*b^2*c+b^4)/c/a*x-1/4/
c*(8*a^3*c*l+a^2*b^2*l-6*a^2*b*c*j+8*a^2*c^2*g+a*b^2*c*g-10*a*b*c^2*e+b^3*c
*e)/(16*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((-(16*a^3*b*c*m-
12*a^3*c^2*k-a^2*b^3*m-3*a^2*b^2*c*k+12*a^2*b*c^2*h-20*a^2*c^3*f-a*b^2*c^2*
f+24*a*b*c^3*d-3*b^3*c^2*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)/c*_R^2-8*(3*a*b*
l-2*a*c*j-b^2*j+3*b*c*g-6*c^2*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*_R+1/c*(20*a^4*
c*m+a^3*b^2*m-12*a^3*b*c*k+12*a^3*c^2*h+3*a^2*b^2*c*h-16*a^2*b*c^2*f+84*a^2
*c^3*d+a*b^3*c*f-27*a*b^2*c^2*d+3*b^4*c*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4))/
(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + jx^5 + kx^6 + lx^7 + mx^8}{(a + bx^2 + cx^4)^3} \, dx = \text{Timed out}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx = \text{Timed out}$$

```
[In] integrate((m*x**8+l*x**7+k*x**6+j*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+
b*x**2+a)**3,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx = \int \frac{mx^8+lx^7+kx^6+jx^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^3,x, algorithm="maxima")
[Out] -1/8*((12*a^2*b*c^3*h - 3*(b^3*c^3 - 8*a*b*c^4)*d - (a*b^2*c^3 + 20*a^2*c^4
)*f - 3*(a^2*b^2*c^2 + 4*a^3*c^3)*k - (a^2*b^3*c - 16*a^3*b*c^2)*m)*x^7 - 1
2*a^4*b*c*j - 4*(6*a^2*c^4*e - 3*a^2*b*c^3*g - 3*a^3*b*c^2*l + (a^2*b^2*c^2
 + 2*a^3*c^3)*j)*x^6 - ((6*b^4*c^2 - 49*a*b^2*c^3 + 28*a^2*c^4)*d + 2*(a*b^
3*c^2 + 14*a^2*b*c^3)*f - (19*a^2*b^2*c^2 - 4*a^3*c^3)*h + (5*a^2*b^3*c + 1
6*a^3*b*c^2)*k - (a^2*b^4 + 5*a^3*b^2*c + 36*a^4*c^2)*m)*x^5 - 2*(18*a^2*b*
c^3*e - 9*a^2*b^2*c^2*g + 3*(a^2*b^3*c + 2*a^3*b*c^2)*j - (a^2*b^4 + a^3*b^
2*c + 16*a^4*c^2)*l)*x^4 - ((3*b^5*c - 20*a*b^3*c^2 - 4*a^2*b*c^3)*d + (a*b
^4*c + 5*a^2*b^2*c^2 + 36*a^3*c^3)*f - (5*a^2*b^3*c + 16*a^3*b*c^2)*h + (19
*a^3*b^2*c - 4*a^4*c^2)*k - 2*(a^3*b^3 + 14*a^4*b*c)*m)*x^3 - 4*(2*(a^2*b^2
*c^2 + 5*a^3*c^3)*e - (a^2*b^3*c + 5*a^3*b*c^2)*g + (5*a^3*b^2*c - 2*a^4*c^
2)*j - (a^3*b^3 + 5*a^4*b*c)*l)*x^2 + 2*(a^2*b^3*c - 10*a^3*b*c^2)*e + 2*(a
^3*b^2*c + 8*a^4*c^2)*g + 2*(a^4*b^2 + 8*a^5*c)*l - (12*a^4*b*c*k + (5*a*b^
4*c - 37*a^2*b^2*c^2 + 44*a^3*c^3)*d - (a^2*b^3*c - 16*a^3*b*c^2)*f - 3*(a^
3*b^2*c + 4*a^4*c^2)*h - (a^4*b^2 + 20*a^5*c)*m)*x)/(a^4*b^4*c - 8*a^5*b^2*
c^2 + 16*a^6*c^3 + (a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5)*x^8 + 2*(a^2*
b^5*c^2 - 8*a^3*b^3*c^3 + 16*a^4*b*c^4)*x^6 + (a^2*b^6*c - 6*a^3*b^4*c^2 +
32*a^5*c^4)*x^4 + 2*(a^3*b^5*c - 8*a^4*b^3*c^2 + 16*a^5*b*c^3)*x^2) - 1/8*i
ntegrate((12*a^3*b*c*k + (12*a^2*b*c^2*h - 3*(b^3*c^2 - 8*a*b*c^3)*d - (a*b
^2*c^2 + 20*a^2*c^3)*f - 3*(a^2*b^2*c + 4*a^3*c^2)*k - (a^2*b^3 - 16*a^3*b*
c)*m)*x^2 - 3*(b^4*c - 9*a*b^2*c^2 + 28*a^2*c^3)*d - (a*b^3*c - 16*a^2*b*c^
2)*f - 3*(a^2*b^2*c + 4*a^3*c^2)*h - (a^3*b^2 + 20*a^4*c)*m - 8*(6*a^2*c^3*
e - 3*a^2*b*c^2*g - 3*a^3*b*c*l + (a^2*b^2*c + 2*a^3*c^2)*j)*x)/(c*x^4 + b*
x^2 + a), x)/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 22429 vs. 2(1098) = 2196.

Time = 3.50 (sec) , antiderivative size = 22429, normalized size of antiderivative = 19.50

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + jx^5 + kx^6 + lx^7 + mx^8}{(a + bx^2 + cx^4)^3}\, dx = \text{Too large to display}$$

```
[In] integrate((m*x^8+l*x^7+k*x^6+j*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)
^3,x, algorithm="giac")
```

```
[Out] 1/64*(3*(2*b^5*c^4 - 24*a*b^3*c^5 + 64*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*b^3*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 16*(b^2 - 4*a*c)*a*b*c^5)*(a
^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*d + (2*a*b^4*c^4 + 32*a^2*b^2*c^5
- 160*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
*b^4*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2
*b^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^
3*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^
4 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 20*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^5 - 2*(b^2
- 4*a*c)*a*b^2*c^4 - 40*(b^2 - 4*a*c)*a^2*c^5)*(a^2*b^4*c - 8*a^3*b^2*c^2 +
 16*a^4*c^3)^2*f - 12*(2*a^2*b^3*c^4 - 8*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b*c^4)*(a^2*b^
4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*h + 3*(2*a^2*b^4*c^3 - 32*a^4*c^5 - sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 8*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 4*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 2*(b^2 - 4*a*c)*a^2*b^2*
c^3 - 8*(b^2 - 4*a*c)*a^3*c^4)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*k
 + (2*a^2*b^5*c^2 - 40*a^3*b^3*c^3 + 128*a^4*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
```

b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 32*(b^2 - 4*a*c)*a^3*b*c^3)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*m + 6*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^10*c^3 - 21*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^8*c^4 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^9*c^4 - 2*a^2*b^10*c^4 + 184*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + 34*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^8*c^5 + 42*a^3*b^8*c^5 - 832*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 232*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 - 17*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 - 368*a^4*b^6*c^6 + 1920*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 736*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 116*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^7 + 1664*a^5*b^4*c^7 - 1792*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*c^8 - 896*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 - 368*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 3840*a^6*b^2*c^8 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*c^9 + 3584*a^7*c^9 + 2*(b^2 - 4*a*c)*a^2*b^8*c^4 - 34*(b^2 - 4*a*c)*a^3*b^6*c^5 + 232*(b^2 - 4*a*c)*a^4*b^4*c^6 - 736*(b^2 - 4*a*c)*a^5*b^2*c^7 + 896*(b^2 - 4*a*c)*a^6*c^8)*d*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^9*c^3 - 28*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^7*c^4 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^8*c^4 - 2*a^3*b^9*c^4 + 240*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^5 + 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^5 + 56*a^4*b^7*c^5 - 832*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 - 288*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 - 480*a^5*b^5*c^6 + 1024*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b*c^7 + 512*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 144*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 1664*a^6*b^3*c^7 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 - 2048*a^7*b*c^8 + 2*(b^2 - 4*a*c)*a^3*b^7*c^4 - 48*(b^2 - 4*a*c)*a^4*b^5*c^5 + 288*(b^2 - 4*a*c)*a^5*b^3*c^6 - 512*(b^2 - 4*a*c)*a^6*b*c^7)*f*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + 6*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^8*c^3 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^6*c^4 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^7*c^4 - 2*a^4*b^8*c^4 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + 16*a^5*b^6*c^5 + 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^6 + 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*c^7 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b*c^7 - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 - 256*a^7*b^2*c^7 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*c^8 + 512*a^8*c^8 + 2*(b^2 - 4*a*c)*a^4*b^6*c^4 - 8*(b^2 - 4*a*c)*a^5*b^4*c^5 - 32

*(b^2 - 4*a*c)*a^6*b^2*c^6 + 128*(b^2 - 4*a*c)*a^7*c^7)*h*abs(a^2*b^4*c - 8
*a^3*b^2*c^2 + 16*a^4*c^3) - 24*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
5*b^7*c^3 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^4 - 2*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^6*c^4 - 2*a^5*b^7*c^4 + 48*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^3*c^5 + 16*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^6*b^4*c^5 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b
^5*c^5 + 24*a^6*b^5*c^5 - 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b*
c^6 - 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^6 - 8*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 - 96*a^7*b^3*c^6 + 16*sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b*c^7 + 128*a^8*b*c^7 + 2*(b^2 - 4*a*c)*a^
5*b^5*c^4 - 16*(b^2 - 4*a*c)*a^6*b^3*c^5 + 32*(b^2 - 4*a*c)*a^7*b*c^6)*k*ab
s(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^5*b^8*c^2 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^6
*c^3 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^3 - 2*a^5*b^8*c^
3 - 192*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^4 - 24*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^4 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^5*b^6*c^4 - 16*a^6*b^6*c^4 + 896*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^8*b^2*c^5 + 288*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^3*
c^5 + 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^4*c^5 + 384*a^7*b^4*
c^5 - 1280*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*c^6 - 640*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b*c^6 - 144*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^7*b^2*c^6 - 1792*a^8*b^2*c^6 + 320*sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^8*c^7 + 2560*a^9*c^7 + 2*(b^2 - 4*a*c)*a^5*b^6*c^3 + 24*(b^2
- 4*a*c)*a^6*b^4*c^4 - 288*(b^2 - 4*a*c)*a^7*b^2*c^5 + 640*(b^2 - 4*a*c)*a
^8*c^6)*m*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + 3*(2*a^4*b^13*c^6 -
52*a^5*b^11*c^7 + 624*a^6*b^9*c^8 - 4224*a^7*b^7*c^9 + 16384*a^8*b^5*c^10
- 33792*a^9*b^3*c^11 + 28672*a^10*b*c^12 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^13*c^4 + 26*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^12*c^5 - 312*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^6 - 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^11*c^6 + 2112*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^7 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^7 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^9*c^7 - 8192*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^8 - 2432*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^8 - 224*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^7*c^8 + 16896*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^9 + 6656*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^9 + 1216*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^5*c^9 - 14336*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^10 - 7168*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^10 - 3328*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^10 + 3584*sqrt

(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b*c^11 - 2*(b^2 - 4*a*c)*a^4*b^11*c^6 + 44*(b^2 - 4*a*c)*a^5*b^9*c^7 - 448*(b^2 - 4*a*c)*a^6*b^7*c^8 + 2432*(b^2 - 4*a*c)*a^7*b^5*c^9 - 6656*(b^2 - 4*a*c)*a^8*b^3*c^10 + 7168*(b^2 - 4*a*c)*a^9*b*c^11)*d + (2*a^5*b^12*c^6 - 136*a^6*b^10*c^7 + 1856*a^7*b^8*c^8 - 10496*a^8*b^6*c^9 + 27136*a^9*b^4*c^10 - 26624*a^10*b^2*c^11 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^4 + 68*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^5 - 928*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^6 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^6 + 5248*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^7 + 1344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^7 + 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^7 - 13568*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^8 - 5120*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^8 - 672*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^8 + 13312*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^9 + 6656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^9 + 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^9 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^10 - 2*(b^2 - 4*a*c)*a^5*b^10*c^6 + 128*(b^2 - 4*a*c)*a^6*b^8*c^7 - 1344*(b^2 - 4*a*c)*a^7*b^6*c^8 + 5120*(b^2 - 4*a*c)*a^8*b^4*c^9 - 6656*(b^2 - 4*a*c)*a^9*b^2*c^10)*f + 6*(6*a^6*b^11*c^6 - 88*a^7*b^9*c^7 + 448*a^8*b^7*c^8 - 768*a^9*b^5*c^9 - 512*a^10*b^3*c^10 + 2048*a^11*b*c^11 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^4 + 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^5 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^5 - 224*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^6 - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^6 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^6 + 384*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^7 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^7 + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^7 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^3*c^8 - 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^8 - 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b*c^9 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^9 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^10 - 6*(b^2 - 4*a*c)*a^6*b^9*c^6 + 64*(b^2 - 4*a*c)*a^7*b^7*c^7 - 192*(b^2 - 4*a*c)*a^8*b^5*c^8 + 512*(b^2 - 4*a*c)*a^10*b*c^10)*h - 3*(2*a^6*b^12*c^5 - 8*a^7*b^10*c^6 - 192*a^8*b^8*c^7 + 1792*a^9*b^6*c^8 - 5632*a^10*b^4*c^9 + 6144*a^11*b^2*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^12*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^10*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^

2 - 4*a*c)*c)*a^6*b^11*c^4 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^8*b^8*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^6*b^10*c^5 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^9*b^6*c^6 - 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^8*b^7*c^6 + 2816*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^10*b^4*c^7 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^9*b^5*c^7 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^8*b^6*c^7 - 3072*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^11*b^2*c^8 - 1536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^10*b^3*c^8 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^8 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^9 - 2*(b^2 - 4*a*c)*a^6*b^10*c^5 + 1
92*(b^2 - 4*a*c)*a^8*b^6*c^7 - 1024*(b^2 - 4*a*c)*a^9*b^4*c^8 + 1536*(b^2 -
 4*a*c)*a^10*b^2*c^9)*k - (2*a^6*b^13*c^4 - 68*a^7*b^11*c^5 + 688*a^8*b^9*c
^6 - 2688*a^9*b^7*c^7 + 2048*a^10*b^5*c^8 + 11264*a^11*b^3*c^9 - 20480*a^12
*b*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^1
3*c^2 + 34*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^
11*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^
12*c^3 - 344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*
b^9*c^4 - 60*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*
b^10*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^
11*c^4 + 1344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9
*b^7*c^5 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
8*b^8*c^5 + 30*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
7*b^9*c^5 - 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^10*b^5*c^6 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^9*b^6*c^6 - 224*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^8*b^7*c^6 - 5632*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^11*b^3*c^7 - 1536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^10*b^4*c^7 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^9*b^5*c^7 + 10240*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^12*b*c^8 + 5120*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^11*b^2*c^8 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^10*b^3*c^8 - 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^11*b*c^9 - 2*(b^2 - 4*a*c)*a^6*b^11*c^4 + 60*(b^2 - 4*
a*c)*a^7*b^9*c^5 - 448*(b^2 - 4*a*c)*a^8*b^7*c^6 + 896*(b^2 - 4*a*c)*a^9*b^
5*c^7 + 1536*(b^2 - 4*a*c)*a^10*b^3*c^8 - 5120*(b^2 - 4*a*c)*a^11*b*c^9)*m)
*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3 + sqrt
((a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3)^2 - 4*(a^3*b^4*c - 8*a^4*b^2*c^
2 + 16*a^5*c^3)*(a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)))/(a^2*b^4*c^2 -
 8*a^3*b^2*c^3 + 16*a^4*c^4)))/((a^5*b^10*c^3 - 20*a^6*b^8*c^4 - 2*a^5*b^9*
c^4 + 160*a^7*b^6*c^5 + 32*a^6*b^7*c^5 + a^5*b^8*c^5 - 640*a^8*b^4*c^6 - 19
2*a^7*b^5*c^6 - 16*a^6*b^6*c^6 + 1280*a^9*b^2*c^7 + 512*a^8*b^3*c^7 + 96*a^
7*b^4*c^7 - 1024*a^10*c^8 - 512*a^9*b*c^8 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*
abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)*abs(c)) - 1/64*(3*(2*b^5*c^4 -

24*a*b^3*c^5 + 64*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 16*(b^2 - 4*a*c)*a*b*c^5)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*d + (2*a*b^4*c^4 + 32*a^2*b^2*c^5 - 160*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c^5 - 2*(b^2 - 4*a*c)*a*b^2*c^4 - 40*(b^2 - 4*a*c)*a^2*c^5)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*f - 12*(2*a^2*b^3*c^4 - 8*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b*c^4)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*h + 3*(2*a^2*b^4*c^3 - 32*a^4*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 2*(b^2 - 4*a*c)*a^2*b^2*c^3 - 8*(b^2 - 4*a*c)*a^3*c^4)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*k + (2*a^2*b^5*c^2 - 40*a^3*b^3*c^3 + 128*a^4*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 32*(b^2 - 4*a*c)*a^3*b*c^3)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)^2*m - 6*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^10*c^3 - 21*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^8*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^9*c^4 + 2*a^2*b^10*c^4 + 184*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + 34*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^5 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^8*c^5 - 42*a^3*b^8*c^5 - 832*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 232*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^

5*c^6 - 17*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 + 368*a^4*b^
6*c^6 + 1920*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 736*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 116*sqrt(2)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^4*b^4*c^7 - 1664*a^5*b^4*c^7 - 1792*sqrt(2)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*a^7*c^8 - 896*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^6*b*c^8 - 368*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 + 3840
*a^6*b^2*c^8 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*c^9 - 3584*a
^7*c^9 - 2*(b^2 - 4*a*c)*a^2*b^8*c^4 + 34*(b^2 - 4*a*c)*a^3*b^6*c^5 - 232*(
b^2 - 4*a*c)*a^4*b^4*c^6 + 736*(b^2 - 4*a*c)*a^5*b^2*c^7 - 896*(b^2 - 4*a*c
)*a^6*c^8)*d*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) - 2*(sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^9*c^3 - 28*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^4*b^7*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^8*c^
4 + 2*a^3*b^9*c^4 + 240*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^5
 + 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^5 - 56*a^4*b^7*c^5 - 832*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 - 288*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^5*b^4*c^6 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^
6 + 480*a^5*b^5*c^6 + 1024*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b*c^
7 + 512*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 144*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 - 1664*a^6*b^3*c^7 - 256*sqrt(2)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 + 2048*a^7*b*c^8 - 2*(b^2 - 4*a*
c)*a^3*b^7*c^4 + 48*(b^2 - 4*a*c)*a^4*b^5*c^5 - 288*(b^2 - 4*a*c)*a^5*b^3*c
^6 + 512*(b^2 - 4*a*c)*a^6*b*c^7)*f*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*
c^3) - 6*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^8*c^3 - 8*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^6*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^4*b^7*c^4 + 2*a^4*b^8*c^4 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^5*b^5*c^5 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5
 - 16*a^5*b^6*c^5 + 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^6
 + 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 - 4*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^8*c^7 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b*c^7 -
16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 256*a^7*b^2*c^7 +
64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*c^8 - 512*a^8*c^8 - 2*(b^2 -
 4*a*c)*a^4*b^6*c^4 + 8*(b^2 - 4*a*c)*a^5*b^4*c^5 + 32*(b^2 - 4*a*c)*a^6*b^
2*c^6 - 128*(b^2 - 4*a*c)*a^7*c^7)*h*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4
c^3) + 24(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^3 - 12*sqrt(2
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^4 - 2*sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^5*b^6*c^4 + 2*a^5*b^7*c^4 + 48*sqrt(2)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^7*b^3*c^5 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b
^4*c^5 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^5 - 24*a^6*b^5*c
^5 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b*c^6 - 32*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^6 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^6*b^3*c^6 + 96*a^7*b^3*c^6 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^7*b*c^7 - 128*a^8*b*c^7 - 2*(b^2 - 4*a*c)*a^5*b^5*c^4 + 16*(b^2 -
 4*a*c)*a^6*b^3*c^5 - 32*(b^2 - 4*a*c)*a^7*b*c^6)*k*abs(a^2*b^4*c - 8*a^3*b

^2*c^2 + 16*a^4*c^3) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^8*c
^2 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^6*c^3 - 2*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^3 + 2*a^5*b^8*c^3 - 192*sqrt(2)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^4 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^6*b^5*c^4 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^6*c^4
+ 16*a^6*b^6*c^4 + 896*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^5
+ 288*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^3*c^5 + 12*sqrt(2)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^4*c^5 - 384*a^7*b^4*c^5 - 1280*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*c^6 - 640*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^8*b*c^6 - 144*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^2*c^
6 + 1792*a^8*b^2*c^6 + 320*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*c^7
- 2560*a^9*c^7 - 2*(b^2 - 4*a*c)*a^5*b^6*c^3 - 24*(b^2 - 4*a*c)*a^6*b^4*c^4
+ 288*(b^2 - 4*a*c)*a^7*b^2*c^5 - 640*(b^2 - 4*a*c)*a^8*c^6)*m*abs(a^2*b^4
*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + 3*(2*a^4*b^13*c^6 - 52*a^5*b^11*c^7 + 62
4*a^6*b^9*c^8 - 4224*a^7*b^7*c^9 + 16384*a^8*b^5*c^10 - 33792*a^9*b^3*c^11
+ 28672*a^10*b*c^12 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^4*b^13*c^4 + 26*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^5*b^11*c^5 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^4*b^12*c^5 - 312*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^6*b^9*c^6 - 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^5*b^10*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^4*b^11*c^6 + 2112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^7*b^7*c^7 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^6*b^8*c^7 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^5*b^9*c^7 - 8192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^8*b^5*c^8 - 2432*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^7*b^6*c^8 - 224*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^6*b^7*c^8 + 16896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^9*b^3*c^9 + 6656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^8*b^4*c^9 + 1216*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^7*b^5*c^9 - 14336*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^10*b*c^10 - 7168*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^10 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^10 + 3584*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b*c^11 - 2*(b^2 - 4*a*c)*a^4*b^11*c^6
+ 44*(b^2 - 4*a*c)*a^5*b^9*c^7 - 448*(b^2 - 4*a*c)*a^6*b^7*c^8 + 2432*(b^2
- 4*a*c)*a^7*b^5*c^9 - 6656*(b^2 - 4*a*c)*a^8*b^3*c^10 + 7168*(b^2 - 4*a*c)
*a^9*b*c^11)*d + (2*a^5*b^12*c^6 - 136*a^6*b^10*c^7 + 1856*a^7*b^8*c^8 - 10
496*a^8*b^6*c^9 + 27136*a^9*b^4*c^10 - 26624*a^10*b^2*c^11 - sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^4 + 68*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^5 + 2*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^5 - 928*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^6 - 128*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^6 - sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^6 + 5248*sqrt(2)*sq

rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^7 + 1344*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^7 + 64*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^7 - 13568*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^8 - 5120*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^8 - 672*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^8 + 13312
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^9 + 6
656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^9 +
 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^9
 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c
^10 - 2*(b^2 - 4*a*c)*a^5*b^10*c^6 + 128*(b^2 - 4*a*c)*a^6*b^8*c^7 - 1344*(
b^2 - 4*a*c)*a^7*b^6*c^8 + 5120*(b^2 - 4*a*c)*a^8*b^4*c^9 - 6656*(b^2 - 4*a
*c)*a^9*b^2*c^10)*f + 6*(6*a^6*b^11*c^6 - 88*a^7*b^9*c^7 + 448*a^8*b^7*c^8
 - 768*a^9*b^5*c^9 - 512*a^10*b^3*c^10 + 2048*a^11*b*c^11 - 3*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^4 + 44*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^5 + 6*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^5 - 224*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^6 - 64*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^6 - 3*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^6 + 384*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^7 + 192*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^7 + 32*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^7 + 256*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^3*c^8 - 96*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^8 - 1024*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b*c^9 - 512*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^9 + 256*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b*c^10 - 6*(b^2 -
 4*a*c)*a^6*b^9*c^6 + 64*(b^2 - 4*a*c)*a^7*b^7*c^7 - 192*(b^2 - 4*a*c)*a^8*
b^5*c^8 + 512*(b^2 - 4*a*c)*a^10*b*c^10)*h - 3*(2*a^6*b^12*c^5 - 8*a^7*b^10
*c^6 - 192*a^8*b^8*c^7 + 1792*a^9*b^6*c^8 - 5632*a^10*b^4*c^9 + 6144*a^11*b
^2*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^1
2*c^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^1
0*c^4 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^1
1*c^4 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^
8*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*
c^5 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^6
*c^6 - 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^
7*c^6 + 2816*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10
*b^4*c^7 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^9*b^5*c^7 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^8*b^6*c^7 - 3072*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^11*b^2*c^8 - 1536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^10*b^3*c^8 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^9*b^4*c^8 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a

*c)*c)*a^10*b^2*c^9 - 2*(b^2 - 4*a*c)*a^6*b^10*c^5 + 192*(b^2 - 4*a*c)*a^8*
b^6*c^7 - 1024*(b^2 - 4*a*c)*a^9*b^4*c^8 + 1536*(b^2 - 4*a*c)*a^10*b^2*c^9)
*k - (2*a^6*b^13*c^4 - 68*a^7*b^11*c^5 + 688*a^8*b^9*c^6 - 2688*a^9*b^7*c^7
 + 2048*a^10*b^5*c^8 + 11264*a^11*b^3*c^9 - 20480*a^12*b*c^10 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^13*c^2 + 34*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^11*c^3 + 2*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^12*c^3 - 344*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^9*c^4 - 60*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^10*c^4 - sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^4 + 1344*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^7*c^5 + 448*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^8*c^5 + 30*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^5 - 1024*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^5*c^6 - 896*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^6*c^6 - 224*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^6 - 5632
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^3*c^7 - 1
536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^4*c^7
 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^7
 + 10240*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b*c
^8 + 5120*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^
2*c^8 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*
b^3*c^8 - 2560*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
11*b*c^9 - 2*(b^2 - 4*a*c)*a^6*b^11*c^4 + 60*(b^2 - 4*a*c)*a^7*b^9*c^5 - 44
8*(b^2 - 4*a*c)*a^8*b^7*c^6 + 896*(b^2 - 4*a*c)*a^9*b^5*c^7 + 1536*(b^2 - 4
*a*c)*a^10*b^3*c^8 - 5120*(b^2 - 4*a*c)*a^11*b*c^9)*m)*arctan(2*sqrt(1/2)*x
/sqrt((a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3 - sqrt((a^2*b^5*c - 8*a^3*b
^3*c^2 + 16*a^4*b*c^3)^2 - 4*(a^3*b^4*c - 8*a^4*b^2*c^2 + 16*a^5*c^3)*(a^2*
b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)))/(a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a
^4*c^4)))/((a^5*b^10*c^3 - 20*a^6*b^8*c^4 - 2*a^5*b^9*c^4 + 160*a^7*b^6*c^5
 + 32*a^6*b^7*c^5 + a^5*b^8*c^5 - 640*a^8*b^4*c^6 - 192*a^7*b^5*c^6 - 16*a^
6*b^6*c^6 + 1280*a^9*b^2*c^7 + 512*a^8*b^3*c^7 + 96*a^7*b^4*c^7 - 1024*a^10
*c^8 - 512*a^9*b*c^8 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*abs(a^2*b^4*c - 8*a^3
*b^2*c^2 + 16*a^4*c^3)*abs(c)) + 1/8*(3*b^3*c^3*d*x^7 - 24*a*b*c^4*d*x^7 +
a*b^2*c^3*f*x^7 + 20*a^2*c^4*f*x^7 - 12*a^2*b*c^3*h*x^7 + 3*a^2*b^2*c^2*k*x
^7 + 12*a^3*c^3*k*x^7 + a^2*b^3*c*m*x^7 - 16*a^3*b*c^2*m*x^7 + 24*a^2*c^4*e
*x^6 - 12*a^2*b*c^3*g*x^6 + 4*a^2*b^2*c^2*j*x^6 + 8*a^3*c^3*j*x^6 - 12*a^3*
b*c^2*l*x^6 + 6*b^4*c^2*d*x^5 - 49*a*b^2*c^3*d*x^5 + 28*a^2*c^4*d*x^5 + 2*a
*b^3*c^2*f*x^5 + 28*a^2*b*c^3*f*x^5 - 19*a^2*b^2*c^2*h*x^5 + 4*a^3*c^3*h*x^
5 + 5*a^2*b^3*c*k*x^5 + 16*a^3*b*c^2*k*x^5 - a^2*b^4*m*x^5 - 5*a^3*b^2*c*m*
x^5 - 36*a^4*c^2*m*x^5 + 36*a^2*b*c^3*e*x^4 - 18*a^2*b^2*c^2*g*x^4 + 6*a^2*
b^3*c*j*x^4 + 12*a^3*b*c^2*j*x^4 - 2*a^2*b^4*l*x^4 - 2*a^3*b^2*c*l*x^4 - 32
*a^4*c^2*l*x^4 + 3*b^5*c*d*x^3 - 20*a*b^3*c^2*d*x^3 - 4*a^2*b*c^3*d*x^3 + a
*b^4*c*f*x^3 + 5*a^2*b^2*c^2*f*x^3 + 36*a^3*c^3*f*x^3 - 5*a^2*b^3*c*h*x^3 -
 16*a^3*b*c^2*h*x^3 + 19*a^3*b^2*c*k*x^3 - 4*a^4*c^2*k*x^3 - 2*a^3*b^3*m*x^

3 - 28*a^4*b*c*m*x^3 + 8*a^2*b^2*c^2*e*x^2 + 40*a^3*c^3*e*x^2 - 4*a^2*b^3*c
*g*x^2 - 20*a^3*b*c^2*g*x^2 + 20*a^3*b^2*c*j*x^2 - 8*a^4*c^2*j*x^2 - 4*a^3*
b^3*l*x^2 - 20*a^4*b*c*l*x^2 + 5*a*b^4*c*d*x - 37*a^2*b^2*c^2*d*x + 44*a^3*
c^3*d*x - a^2*b^3*c*f*x + 16*a^3*b*c^2*f*x - 3*a^3*b^2*c*h*x - 12*a^4*c^2*h
*x + 12*a^4*b*c*k*x - a^4*b^2*m*x - 20*a^5*c*m*x - 2*a^2*b^3*c*e + 20*a^3*b
*c^2*e - 2*a^3*b^2*c*g - 16*a^4*c^2*g + 12*a^4*b*c*j - 2*a^4*b^2*l - 16*a^5
*c*l)/((a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)*(c*x^4 + b*x^2 + a)^2) + 1/
8*(6*(b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5 - (b^2*c^3 - 4*a*c^4 - 2*b*c^
4 + c^5)*sqrt(b^2 - 4*a*c))*e*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) -
3*(b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4 + (b^3*c^2 - 4*a*b*c^3 - 2*
b^2*c^3 + b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^
4*c^3) + (b^5*c - 2*a*b^3*c^2 - 2*b^4*c^2 - 8*a^2*b*c^3 - 4*a*b^2*c^3 + b^3
*c^3 + 2*a*b*c^4 + (b^4*c - 2*a*b^2*c^2 - 2*b^3*c^2 - 8*a^2*c^3 - 4*a*b*c^3
+ b^2*c^3 + 2*a*c^4)*sqrt(b^2 - 4*a*c))*j*abs(a^2*b^4*c - 8*a^3*b^2*c^2 +
16*a^4*c^3) - 3*(a*b^4*c - 4*a^2*b^2*c^2 - 2*a*b^3*c^2 + a*b^2*c^3 - (a*b^3
*c - 4*a^2*b*c^2 - 2*a*b^2*c^2 + a*b*c^3)*sqrt(b^2 - 4*a*c))*l*abs(a^2*b^4*
c - 8*a^3*b^2*c^2 + 16*a^4*c^3) - 6*(a^2*b^7*c^4 - 12*a^3*b^5*c^5 - 2*a^2*b
^6*c^5 + 48*a^4*b^3*c^6 + 16*a^3*b^4*c^6 + a^2*b^5*c^6 - 64*a^5*b*c^7 - 32*
a^4*b^2*c^7 - 8*a^3*b^3*c^7 + 16*a^4*b*c^8 + (a^2*b^6*c^4 - 8*a^3*b^4*c^5 -
2*a^2*b^5*c^5 + 16*a^4*b^2*c^6 + 8*a^3*b^3*c^6 + a^2*b^4*c^6 - 4*a^3*b^2*c
^7)*sqrt(b^2 - 4*a*c))*e + 3*(a^2*b^8*c^3 - 12*a^3*b^6*c^4 - 2*a^2*b^7*c^4
+ 48*a^4*b^4*c^5 + 16*a^3*b^5*c^5 + a^2*b^6*c^5 - 64*a^5*b^2*c^6 - 32*a^4*b
^3*c^6 - 8*a^3*b^4*c^6 + 16*a^4*b^2*c^7 + (a^2*b^7*c^3 - 8*a^3*b^5*c^4 - 2*
a^2*b^6*c^4 + 16*a^4*b^3*c^5 + 8*a^3*b^4*c^5 + a^2*b^5*c^5 - 4*a^3*b^3*c^6)
*sqrt(b^2 - 4*a*c))*g - (a^2*b^9*c^2 - 10*a^3*b^7*c^3 - 2*a^2*b^8*c^3 + 24*
a^4*b^5*c^4 + 12*a^3*b^6*c^4 + a^2*b^7*c^4 + 32*a^5*b^3*c^5 - 6*a^3*b^5*c^5
- 128*a^6*b*c^6 - 64*a^5*b^2*c^6 + 32*a^5*b*c^7 + (a^2*b^8*c^2 - 6*a^3*b^6
*c^3 - 2*a^2*b^7*c^3 + 4*a^3*b^5*c^4 + a^2*b^6*c^4 + 32*a^5*b^2*c^5 + 16*a^
4*b^3*c^5 - 2*a^3*b^4*c^5 - 8*a^4*b^2*c^6)*sqrt(b^2 - 4*a*c))*j + 3*(a^3*b^
8*c^2 - 12*a^4*b^6*c^3 - 2*a^3*b^7*c^3 + 48*a^5*b^4*c^4 + 16*a^4*b^5*c^4 +
a^3*b^6*c^4 - 64*a^6*b^2*c^5 - 32*a^5*b^3*c^5 - 8*a^4*b^4*c^5 + 16*a^5*b^2*
c^6 + (a^3*b^7*c^2 - 8*a^4*b^5*c^3 - 2*a^3*b^6*c^3 + 16*a^5*b^3*c^4 + 8*a^4
*b^4*c^4 + a^3*b^5*c^4 - 4*a^4*b^3*c^5)*sqrt(b^2 - 4*a*c))*l)*log(x^2 + 1/2
*(a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3 + sqrt((a^2*b^5*c - 8*a^3*b^3*c^
2 + 16*a^4*b*c^3)^2 - 4*(a^3*b^4*c - 8*a^4*b^2*c^2 + 16*a^5*c^3)*(a^2*b^4*c
^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)))/(a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^
4))/((a*b^6 - 12*a^2*b^4*c - 2*a*b^5*c + 48*a^3*b^2*c^2 + 16*a^2*b^3*c^2 +
a*b^4*c^2 - 64*a^4*c^3 - 32*a^3*b*c^3 - 8*a^2*b^2*c^3 + 16*a^3*c^4)*c^2*abs
(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)) + 1/8*(6*(b^3*c^3 - 4*a*b*c^4 - 2
*b^2*c^4 + b*c^5 - (b^2*c^3 - 4*a*c^4 - 2*b*c^4 + c^5)*sqrt(b^2 - 4*a*c))*e
*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) - 3*(b^4*c^2 - 4*a*b^2*c^3 - 2
*b^3*c^3 + b^2*c^4 - (b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4)*sqrt(b^2 - 4
*a*c))*g*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) + (b^5*c - 2*a*b^3*c^2
- 2*b^4*c^2 - 8*a^2*b*c^3 - 4*a*b^2*c^3 + b^3*c^3 + 2*a*b*c^4 - (b^4*c - 2
*a*b^2*c^2 - 2*b^3*c^2 - 8*a^2*c^3 - 4*a*b*c^3 + b^2*c^3 + 2*a*c^4)*sqrt(b^

```
2 - 4*a*c))*j*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3) - 3*(a*b^4*c - 4*
a^2*b^2*c^2 - 2*a*b^3*c^2 + a*b^2*c^3 - (a*b^3*c - 4*a^2*b*c^2 - 2*a*b^2*c^
2 + a*b*c^3)*sqrt(b^2 - 4*a*c))*l*abs(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^
3) - 6*(a^2*b^7*c^4 - 12*a^3*b^5*c^5 - 2*a^2*b^6*c^5 + 48*a^4*b^3*c^6 + 16*
a^3*b^4*c^6 + a^2*b^5*c^6 - 64*a^5*b*c^7 - 32*a^4*b^2*c^7 - 8*a^3*b^3*c^7 +
 16*a^4*b*c^8 - (a^2*b^6*c^4 - 8*a^3*b^4*c^5 - 2*a^2*b^5*c^5 + 16*a^4*b^2*c
^6 + 8*a^3*b^3*c^6 + a^2*b^4*c^6 - 4*a^3*b^2*c^7)*sqrt(b^2 - 4*a*c))*e + 3*
(a^2*b^8*c^3 - 12*a^3*b^6*c^4 - 2*a^2*b^7*c^4 + 48*a^4*b^4*c^5 + 16*a^3*b^5
*c^5 + a^2*b^6*c^5 - 64*a^5*b^2*c^6 - 32*a^4*b^3*c^6 - 8*a^3*b^4*c^6 + 16*a
^4*b^2*c^7 + (a^2*b^7*c^3 - 8*a^3*b^5*c^4 - 2*a^2*b^6*c^4 + 16*a^4*b^3*c^5
+ 8*a^3*b^4*c^5 + a^2*b^5*c^5 - 4*a^3*b^3*c^6)*sqrt(b^2 - 4*a*c))*g - (a^2*
b^9*c^2 - 10*a^3*b^7*c^3 - 2*a^2*b^8*c^3 + 24*a^4*b^5*c^4 + 12*a^3*b^6*c^4
+ a^2*b^7*c^4 + 32*a^5*b^3*c^5 - 6*a^3*b^5*c^5 - 128*a^6*b*c^6 - 64*a^5*b^2
*c^6 + 32*a^5*b*c^7 + (a^2*b^8*c^2 - 6*a^3*b^6*c^3 - 2*a^2*b^7*c^3 + 4*a^3*
b^5*c^4 + a^2*b^6*c^4 + 32*a^5*b^2*c^5 + 16*a^4*b^3*c^5 - 2*a^3*b^4*c^5 - 8
*a^4*b^2*c^6)*sqrt(b^2 - 4*a*c))*j + 3*(a^3*b^8*c^2 - 12*a^4*b^6*c^3 - 2*a^
3*b^7*c^3 + 48*a^5*b^4*c^4 + 16*a^4*b^5*c^4 + a^3*b^6*c^4 - 64*a^6*b^2*c^5
- 32*a^5*b^3*c^5 - 8*a^4*b^4*c^5 + 16*a^5*b^2*c^6 - (a^3*b^7*c^2 - 8*a^4*b^
5*c^3 - 2*a^3*b^6*c^3 + 16*a^5*b^3*c^4 + 8*a^4*b^4*c^4 + a^3*b^5*c^4 - 4*a^
4*b^3*c^5)*sqrt(b^2 - 4*a*c))*l)*log(x^2 + 1/2*(a^2*b^5*c - 8*a^3*b^3*c^2 +
 16*a^4*b*c^3 - sqrt((a^2*b^5*c - 8*a^3*b^3*c^2 + 16*a^4*b*c^3)^2 - 4*(a^3*
b^4*c - 8*a^4*b^2*c^2 + 16*a^5*c^3)*(a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c
^4)))/(a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4))/((a*b^6 - 12*a^2*b^4*c -
2*a*b^5*c + 48*a^3*b^2*c^2 + 16*a^2*b^3*c^2 + a*b^4*c^2 - 64*a^4*c^3 - 32*a
^3*b*c^3 - 8*a^2*b^2*c^3 + 16*a^3*c^4)*c^2*abs(a^2*b^4*c - 8*a^3*b^2*c^2 +
16*a^4*c^3))
```

Mupad [B] (verification not implemented)

Time = 32.04 (sec) , antiderivative size = 114377, normalized size of antiderivative = 99.46

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + j*x^5 + k*x^6 + l*x^7 + m*x^8)/(a +
b*x^2 + c*x^4)^3,x)
```

```
[Out] symsum(log(root(56371445760*a^11*b^8*c^9*z^4 - 503316480*a^8*b^14*c^6*z^4 +
 47185920*a^7*b^16*c^5*z^4 - 2621440*a^6*b^18*c^4*z^4 + 65536*a^5*b^20*c^3*
z^4 - 171798691840*a^14*b^2*c^12*z^4 + 193273528320*a^13*b^4*c^11*z^4 - 128
849018880*a^12*b^6*c^10*z^4 - 16911433728*a^10*b^10*c^8*z^4 + 3523215360*a^
9*b^12*c^7*z^4 + 68719476736*a^15*c^13*z^4 + 1536*a^5*b^16*c*k*m*z^2 + 1536
*a*b^18*c^3*d*f*z^2 - 2571632640*a^9*b^5*c^8*d*m*z^2 + 2548039680*a^9*b^3*c
^10*d*h*z^2 + 1509949440*a^10*b^3*c^9*e*l*z^2 + 1509949440*a^9*b^3*c^10*e*g
*z^2 - 1401421824*a^8*b^5*c^9*d*h*z^2 - 1321205760*a^9*b^2*c^11*d*f*z^2 - 2
```

793406464*a^11*b*c^10*d*m*z^2 + 890634240*a^8*b^7*c^7*d*m*z^2 - 754974720*a^10*b^4*c^8*g*l*z^2 - 754974720*a^9*b^5*c^8*e*l*z^2 + 719585280*a^8*b^6*c^8*d*k*z^2 - 707788800*a^9*b^4*c^9*d*k*z^2 - 754974720*a^8*b^5*c^9*e*g*z^2 + 603979776*a^11*b^2*c^9*g*l*z^2 - 581959680*a^10*b^4*c^8*f*m*z^2 + 732168192*a^7*b^6*c^9*d*f*z^2 + 534773760*a^11*b^3*c^8*h*m*z^2 - 456130560*a^11*b^4*c^7*k*m*z^2 - 603979776*a^10*b^2*c^10*e*j*z^2 + 534773760*a^10*b^3*c^9*f*k*z^2 + 384040960*a^9*b^6*c^7*f*m*z^2 + 377487360*a^9*b^6*c^7*g*l*z^2 - 456130560*a^9*b^4*c^9*f*h*z^2 + 301989888*a^11*b^3*c^8*j*l*z^2 - 415236096*a^10*b^2*c^10*d*k*z^2 + 254017536*a^10*b^6*c^6*k*m*z^2 - 330301440*a^10*b^4*c^8*h*k*z^2 + 390463488*a^7*b^7*c^8*d*h*z^2 + 188743680*a^12*b^2*c^8*k*m*z^2 + 301989888*a^10*b^3*c^9*g*j*z^2 - 297861120*a^7*b^8*c^7*d*k*z^2 - 366280704*a^6*b^8*c^8*d*f*z^2 + 188743680*a^11*b^2*c^9*h*k*z^2 - 330301440*a^8*b^4*c^10*d*f*z^2 + 254017536*a^8*b^6*c^8*f*h*z^2 - 1887436800*a^10*b*c^11*d*h*z^2 + 188743680*a^8*b^7*c^7*e*l*z^2 + 153354240*a^9*b^6*c^7*h*k*z^2 - 185303040*a^7*b^9*c^6*d*m*z^2 - 117964800*a^10*b^5*c^7*h*m*z^2 - 61931520*a^9*b^8*c^5*k*m*z^2 + 121634816*a^11*b^2*c^9*f*m*z^2 - 115671040*a^8*b^8*c^6*f*m*z^2 - 62914560*a^9*b^7*c^6*j*l*z^2 + 188743680*a^10*b^2*c^10*f*h*z^2 - 94371840*a^8*b^8*c^6*g*l*z^2 + 6144000*a^8*b^10*c^4*k*m*z^2 - 117964800*a^9*b^5*c^8*f*k*z^2 + 61440*a^7*b^12*c^3*k*m*z^2 - 46080*a^6*b^14*c^2*k*m*z^2 + 23592960*a^8*b^9*c^5*j*l*z^2 + 188743680*a^7*b^7*c^8*e*g*z^2 - 37355520*a^9*b^7*c^6*h*m*z^2 + 125829120*a^8*b^6*c^8*e*j*z^2 + 23101440*a^8*b^9*c^5*h*m*z^2 - 3538944*a^7*b^11*c^4*j*l*z^2 + 196608*a^6*b^13*c^3*j*l*z^2 - 4349952*a^7*b^11*c^4*h*m*z^2 + 337920*a^6*b^13*c^3*h*m*z^2 - 7680*a^5*b^15*c^2*h*m*z^2 - 62914560*a^8*b^7*c^7*g*j*z^2 - 26542080*a^8*b^8*c^6*h*k*z^2 + 17940480*a^7*b^10*c^5*f*m*z^2 + 11796480*a^7*b^10*c^5*g*l*z^2 - 37355520*a^8*b^7*c^7*f*k*z^2 - 1347584*a^6*b^12*c^4*f*m*z^2 + 68272128*a^6*b^10*c^6*d*k*z^2 - 589824*a^6*b^12*c^4*g*l*z^2 + 552960*a^6*b^12*c^4*h*k*z^2 - 147456*a^7*b^10*c^5*h*k*z^2 - 46080*a^5*b^14*c^3*h*k*z^2 + 35840*a^5*b^14*c^3*f*m*z^2 + 23592960*a^7*b^9*c^6*g*j*z^2 - 23592960*a^7*b^9*c^6*e*l*z^2 + 23371776*a^6*b^11*c^5*d*m*z^2 + 23101440*a^7*b^9*c^6*f*k*z^2 - 47185920*a^7*b^8*c^7*e*j*z^2 - 61931520*a^7*b^8*c^7*f*h*z^2 - 4349952*a^6*b^11*c^5*f*k*z^2 - 3538944*a^6*b^11*c^5*g*j*z^2 - 1677312*a^5*b^13*c^4*d*m*z^2 + 1179648*a^6*b^11*c^5*e*l*z^2 + 337920*a^5*b^13*c^4*f*k*z^2 + 196608*a^5*b^13*c^4*g*j*z^2 + 53760*a^4*b^15*c^3*d*m*z^2 - 7680*a^4*b^15*c^3*f*k*z^2 + 96583680*a^5*b^10*c^7*d*f*z^2 - 9179136*a^5*b^12*c^5*d*k*z^2 + 7077888*a^6*b^10*c^6*e*j*z^2 - 51609600*a^6*b^9*c^7*d*h*z^2 + 691200*a^4*b^14*c^4*d*k*z^2 - 393216*a^5*b^12*c^5*e*j*z^2 - 23040*a^3*b^16*c^3*d*k*z^2 + 6144000*a^6*b^10*c^6*f*h*z^2 + 61440*a^5*b^12*c^5*f*h*z^2 - 46080*a^4*b^14*c^4*f*h*z^2 + 1536*a^3*b^16*c^3*f*h*z^2 - 23592960*a^6*b^9*c^7*e*g*z^2 + 1179648*a^5*b^11*c^6*e*g*z^2 + 829440*a^4*b^13*c^5*d*h*z^2 + 368640*a^5*b^11*c^6*d*h*z^2 - 105984*a^3*b^15*c^4*d*h*z^2 + 4608*a^2*b^17*c^3*d*h*z^2 - 15175680*a^4*b^12*c^6*d*f*z^2 + 1428480*a^3*b^14*c^5*d*f*z^2 - 73728*a^2*b^16*c^4*d*f*z^2 + 4108320768*a^10*b^3*c^9*d*m*z^2 - 1207959552*a^11*b*c^10*e*l*z^2 - 1207959552*a^10*b*c^11*e*g*z^2 - 578813952*a^12*b*c^9*h*m*z^2 - 578813952*a^11*b*c^10*f*k*z^2 - 402653184*a^12*b*c^9*j*l*z^2 - 402653184*a^11*b*c^10*g*j*z^2 - 440401920*a^10*b*c^11*f

^2*z^2 - 188743680*a^12*b*c^9*k^2*z^2 - 188743680*a^11*b*c^10*h^2*z^2 + 176
1607680*a^10*c^12*d*f*z^2 - 14080*a^6*b^15*c*m^2*z^2 - 94464*a*b^17*c^4*d^2
*z^2 + 6936330240*a^8*b^3*c^11*d^2*z^2 + 2464874496*a^6*b^7*c^9*d^2*z^2 - 3
963617280*a^9*b*c^12*d^2*z^2 + 1056964608*a^11*c^11*d*k*z^2 + 805306368*a^1
1*c^11*e*j*z^2 + 419430400*a^12*c^10*f*m*z^2 + 251658240*a^13*c^9*k*m*z^2 -
1509949440*a^9*b^2*c^11*e^2*z^2 + 251658240*a^11*c^11*f*h*z^2 + 150994944*
a^12*c^10*h*k*z^2 - 5400428544*a^7*b^5*c^10*d^2*z^2 + 754974720*a^8*b^4*c^1
0*e^2*z^2 - 730054656*a^5*b^9*c^8*d^2*z^2 + 477102080*a^12*b^3*c^7*m^2*z^2
- 377487360*a^11*b^4*c^7*l^2*z^2 + 477102080*a^9*b^3*c^10*f^2*z^2 + 3019898
88*a^12*b^2*c^8*l^2*z^2 - 377487360*a^9*b^4*c^9*g^2*z^2 + 301989888*a^10*b^
2*c^10*g^2*z^2 - 174325760*a^11*b^5*c^6*m^2*z^2 + 188743680*a^10*b^6*c^6*l^
2*z^2 + 141557760*a^11*b^3*c^8*k^2*z^2 + 188743680*a^8*b^6*c^8*g^2*z^2 + 14
1557760*a^10*b^3*c^9*h^2*z^2 - 174325760*a^8*b^5*c^9*f^2*z^2 - 188743680*a^
7*b^6*c^9*e^2*z^2 - 47185920*a^9*b^8*c^5*l^2*z^2 + 11206656*a^10*b^7*c^5*m^
2*z^2 + 8929280*a^9*b^9*c^4*m^2*z^2 - 2600960*a^8*b^11*c^3*m^2*z^2 + 291840
*a^7*b^13*c^2*m^2*z^2 - 50331648*a^10*b^4*c^8*j^2*z^2 + 146165760*a^4*b^11*
c^7*d^2*z^2 - 26542080*a^9*b^7*c^6*k^2*z^2 + 5898240*a^8*b^10*c^4*l^2*z^2 -
294912*a^7*b^12*c^3*l^2*z^2 - 33554432*a^11*b^2*c^9*j^2*z^2 + 9584640*a^8*
b^9*c^5*k^2*z^2 + 20971520*a^9*b^6*c^7*j^2*z^2 - 2359296*a^10*b^5*c^7*k^2*z
^2 - 1290240*a^7*b^11*c^4*k^2*z^2 + 46080*a^6*b^13*c^3*k^2*z^2 + 2304*a^5*b
^15*c^2*k^2*z^2 - 2752512*a^7*b^10*c^5*j^2*z^2 + 2621440*a^8*b^8*c^6*j^2*z^
2 + 524288*a^6*b^12*c^4*j^2*z^2 - 32768*a^5*b^14*c^3*j^2*z^2 - 47185920*a^7
*b^8*c^7*g^2*z^2 - 26542080*a^8*b^7*c^7*h^2*z^2 + 9584640*a^7*b^9*c^6*h^2*z
^2 - 2359296*a^9*b^5*c^8*h^2*z^2 - 1290240*a^6*b^11*c^5*h^2*z^2 + 46080*a^5
*b^13*c^4*h^2*z^2 + 2304*a^4*b^15*c^3*h^2*z^2 + 5898240*a^6*b^10*c^6*g^2*z^
2 - 294912*a^5*b^12*c^5*g^2*z^2 + 11206656*a^7*b^7*c^8*f^2*z^2 + 8929280*a^
6*b^9*c^7*f^2*z^2 + 23592960*a^6*b^8*c^8*e^2*z^2 - 2600960*a^5*b^11*c^6*f^2
*z^2 + 291840*a^4*b^13*c^5*f^2*z^2 - 14080*a^3*b^15*c^4*f^2*z^2 + 256*a^2*b
^17*c^3*f^2*z^2 - 19860480*a^3*b^13*c^6*d^2*z^2 - 1179648*a^5*b^10*c^7*e^2*
z^2 + 1771776*a^2*b^15*c^5*d^2*z^2 - 440401920*a^13*b*c^8*m^2*z^2 + 1207959
552*a^10*c^12*e^2*z^2 + 134217728*a^12*c^10*j^2*z^2 + 256*a^5*b^17*m^2*z^2
+ 2304*b^19*c^3*d^2*z^2 - 23592960*a^10*b*c^8*f*k*l*z + 99090432*a^9*b*c^9*
d*h*l*z + 9437184*a^10*b*c^8*e*k*m*z + 23592960*a^10*b*c^8*g*h*m*z + 141557
760*a^8*b*c^10*d*e*k*z + 47185920*a^9*b*c^9*d*j*k*z - 23592960*a^9*b*c^9*f*
g*k*z + 169869312*a^7*b*c^11*d*e*f*z + 99090432*a^8*b*c^10*d*g*h*z - 314572
8*a^9*b*c^9*f*h*j*z + 56623104*a^8*b*c^10*d*f*j*z + 1536*a*b^15*c^3*d*f*j*z
- 9437184*a^8*b*c^10*e*f*h*z - 4608*a*b^14*c^4*d*f*g*z + 9216*a*b^13*c^5*d
*e*f*z + 412876800*a^8*b^2*c^9*d*e*m*z - 206438400*a^9*b^3*c^7*d*l*m*z + 58
98240*a^10*b^4*c^5*k*l*m*z - 206438400*a^8*b^3*c^8*d*g*m*z - 4718592*a^11*b
^2*c^6*k*l*m*z - 2949120*a^9*b^6*c^4*k*l*m*z + 737280*a^8*b^8*c^3*k*l*m*z -
92160*a^7*b^10*c^2*k*l*m*z + 103219200*a^8*b^5*c^6*d*l*m*z - 29491200*a^10
*b^3*c^6*h*l*m*z - 206438400*a^7*b^4*c^8*d*e*m*z - 2359296*a^10*b^3*c^6*j*k
*m*z + 491520*a^8*b^7*c^4*j*k*m*z - 184320*a^7*b^9*c^3*j*k*m*z + 27648*a^6*
b^11*c^2*j*k*m*z + 14745600*a^9*b^5*c^5*h*l*m*z - 3686400*a^8*b^7*c^4*h*l*m
*z + 460800*a^7*b^9*c^3*h*l*m*z - 23040*a^6*b^11*c^2*h*l*m*z + 88473600*a^8

*b^4*c^7*d*k*l*z + 82575360*a^9*b^2*c^8*d*j*m*z + 11796480*a^10*b^2*c^7*h*j
*m*z + 5898240*a^9*b^4*c^6*g*k*m*z - 4718592*a^10*b^2*c^7*g*k*m*z - 7077888
0*a^9*b^2*c^8*d*k*l*z - 2949120*a^8*b^6*c^5*g*k*m*z - 2457600*a^8*b^6*c^5*h
*j*m*z + 921600*a^7*b^8*c^4*h*j*m*z + 737280*a^7*b^8*c^4*g*k*m*z - 138240*a
^6*b^10*c^3*h*j*m*z - 92160*a^6*b^10*c^3*g*k*m*z + 7680*a^5*b^12*c^2*h*j*m*
z + 4608*a^5*b^12*c^2*g*k*m*z + 29491200*a^9*b^3*c^7*f*k*l*z - 176947200*a^
7*b^3*c^9*d*e*k*z - 109707264*a^8*b^3*c^8*d*h*l*z - 25804800*a^7*b^7*c^5*d*
l*m*z + 103219200*a^7*b^5*c^7*d*g*m*z + 219414528*a^7*b^2*c^10*d*e*h*z - 14
745600*a^8*b^5*c^6*f*k*l*z - 29491200*a^9*b^3*c^7*g*h*m*z - 11796480*a^9*b^
3*c^7*e*k*m*z - 44236800*a^7*b^6*c^6*d*k*l*z + 58982400*a^9*b^2*c^8*e*h*m*z
 + 5898240*a^8*b^5*c^6*e*k*m*z + 3686400*a^7*b^7*c^5*f*k*l*z + 3225600*a^6*
b^9*c^4*d*l*m*z - 1474560*a^7*b^7*c^5*e*k*m*z - 460800*a^6*b^9*c^4*f*k*l*z
+ 184320*a^6*b^9*c^4*e*k*m*z - 161280*a^5*b^11*c^3*d*l*m*z + 23040*a^5*b^11
*c^3*f*k*l*z - 9216*a^5*b^11*c^3*e*k*m*z + 14745600*a^8*b^5*c^6*g*h*m*z + 1
10886912*a^7*b^4*c^8*d*f*l*z - 3686400*a^7*b^7*c^5*g*h*m*z - 221773824*a^6*
b^3*c^10*d*e*f*z + 460800*a^6*b^9*c^4*g*h*m*z - 17203200*a^7*b^6*c^6*d*j*m*
z - 23040*a^5*b^11*c^3*g*h*m*z - 29491200*a^8*b^4*c^7*e*h*m*z - 11796480*a^
9*b^2*c^8*f*j*k*z + 11059200*a^6*b^8*c^5*d*k*l*z + 6451200*a^6*b^8*c^5*d*j*
m*z + 88473600*a^7*b^4*c^8*d*g*k*z + 2457600*a^7*b^6*c^6*f*j*k*z - 35389440
*a^8*b^3*c^8*d*j*k*z - 1382400*a^5*b^10*c^4*d*k*l*z - 84934656*a^8*b^2*c^9*
d*f*l*z - 967680*a^5*b^10*c^4*d*j*m*z - 921600*a^6*b^8*c^5*f*j*k*z + 138240
*a^5*b^10*c^4*f*j*k*z + 69120*a^4*b^12*c^3*d*k*l*z + 53760*a^4*b^12*c^3*d*j
*m*z - 7680*a^4*b^12*c^3*f*j*k*z + 44236800*a^7*b^5*c^7*d*h*l*z + 7372800*a
^7*b^6*c^6*e*h*m*z - 5898240*a^8*b^4*c^7*f*h*l*z + 4718592*a^9*b^2*c^8*f*h*
l*z - 70778880*a^8*b^2*c^9*d*g*k*z + 2949120*a^7*b^6*c^6*f*h*l*z - 921600*a
^6*b^8*c^5*e*h*m*z - 737280*a^6*b^8*c^5*f*h*l*z + 92160*a^5*b^10*c^4*f*h*l*
z + 46080*a^5*b^10*c^4*e*h*m*z - 4608*a^4*b^12*c^3*f*h*l*z + 29491200*a^8*b
^3*c^8*f*g*k*z - 109707264*a^7*b^3*c^9*d*g*h*z - 25804800*a^6*b^7*c^6*d*g*m
*z - 58982400*a^8*b^2*c^9*e*f*k*z - 58982400*a^6*b^6*c^7*d*f*l*z + 7372800*
a^6*b^7*c^6*d*j*k*z + 88473600*a^6*b^5*c^8*d*e*k*z - 2764800*a^5*b^9*c^5*d*
j*k*z + 51609600*a^6*b^6*c^7*d*e*m*z + 414720*a^4*b^11*c^4*d*j*k*z - 23040*
a^3*b^13*c^3*d*j*k*z - 14745600*a^7*b^5*c^7*f*g*k*z - 44236800*a^6*b^6*c^7*
d*g*k*z - 6635520*a^6*b^7*c^6*d*h*l*z + 40108032*a^8*b^2*c^9*d*h*j*z + 3686
400*a^6*b^7*c^6*f*g*k*z + 3225600*a^5*b^9*c^5*d*g*m*z + 2359296*a^8*b^3*c^8
*f*h*j*z - 491520*a^6*b^7*c^6*f*h*j*z - 460800*a^5*b^9*c^5*f*g*k*z - 276480
*a^5*b^9*c^5*d*h*l*z + 184320*a^5*b^9*c^5*f*h*j*z + 179712*a^4*b^11*c^4*d*h
*l*z - 161280*a^4*b^11*c^4*d*g*m*z - 27648*a^4*b^11*c^4*f*h*j*z + 23040*a^4
*b^11*c^4*f*g*k*z - 13824*a^3*b^13*c^3*d*h*l*z + 1536*a^3*b^13*c^3*f*h*j*z
+ 29491200*a^7*b^4*c^8*e*f*k*z + 110886912*a^6*b^4*c^9*d*f*g*z + 16220160*a
^5*b^8*c^6*d*f*l*z - 45613056*a^7*b^3*c^9*d*f*j*z + 11059200*a^5*b^8*c^6*d*
g*k*z - 10321920*a^6*b^6*c^7*d*h*j*z - 7372800*a^6*b^6*c^7*e*f*k*z + 707788
8*a^7*b^4*c^8*d*h*j*z - 6451200*a^5*b^8*c^6*d*e*m*z - 88473600*a^6*b^4*c^9*
d*e*h*z + 2396160*a^5*b^8*c^6*d*h*j*z - 2396160*a^4*b^10*c^5*d*f*l*z - 1382
400*a^4*b^10*c^5*d*g*k*z - 84934656*a^7*b^2*c^10*d*f*g*z + 921600*a^5*b^8*c
^6*e*f*k*z + 117964800*a^5*b^5*c^9*d*e*f*z + 322560*a^4*b^10*c^5*d*e*m*z +

175104*a^3*b^12*c^4*d*f*l*z + 69120*a^3*b^12*c^4*d*g*k*z - 50688*a^3*b^12*c
^4*d*h*j*z - 46080*a^4*b^10*c^5*e*f*k*z - 27648*a^4*b^10*c^5*d*h*j*z + 4608
*a^2*b^14*c^3*d*h*j*z - 4608*a^2*b^14*c^3*d*f*l*z + 44236800*a^6*b^5*c^8*d*
g*h*z - 5898240*a^7*b^4*c^8*f*g*h*z - 22118400*a^5*b^7*c^7*d*e*k*z + 471859
2*a^8*b^2*c^9*f*g*h*z + 2949120*a^6*b^6*c^7*f*g*h*z - 737280*a^5*b^8*c^6*f*
g*h*z + 92160*a^4*b^10*c^5*f*g*h*z - 4608*a^3*b^12*c^4*f*g*h*z + 8847360*a^
5*b^7*c^7*d*f*j*z - 58982400*a^5*b^6*c^8*d*f*g*z - 3809280*a^4*b^9*c^6*d*f*
j*z + 2764800*a^4*b^9*c^6*d*e*k*z + 2359296*a^6*b^5*c^8*d*f*j*z + 681984*a^
3*b^11*c^5*d*f*j*z - 138240*a^3*b^11*c^5*d*e*k*z - 55296*a^2*b^13*c^4*d*f*j
*z + 11796480*a^7*b^3*c^9*e*f*h*z - 6635520*a^5*b^7*c^7*d*g*h*z - 5898240*a
^6*b^5*c^8*e*f*h*z + 1474560*a^5*b^7*c^7*e*f*h*z - 276480*a^4*b^9*c^6*d*g*h
*z - 184320*a^4*b^9*c^6*e*f*h*z + 179712*a^3*b^11*c^5*d*g*h*z - 13824*a^2*b
^13*c^4*d*g*h*z + 9216*a^3*b^11*c^5*e*f*h*z + 16220160*a^4*b^8*c^7*d*f*g*z
+ 13271040*a^5*b^6*c^8*d*e*h*z - 2396160*a^3*b^10*c^6*d*f*g*z + 552960*a^4*
b^8*c^7*d*e*h*z - 359424*a^3*b^10*c^6*d*e*h*z + 175104*a^2*b^12*c^5*d*f*g*z
 + 27648*a^2*b^12*c^5*d*e*h*z - 32440320*a^4*b^7*c^8*d*e*f*z + 4792320*a^3*
b^9*c^7*d*e*f*z - 350208*a^2*b^11*c^6*d*e*f*z + 165150720*a^10*b*c^8*d*l*m*
z + 4608*a^6*b^12*c*k*l*m*z + 23592960*a^11*b*c^7*h*l*m*z + 3145728*a^11*b*
c^7*j*k*m*z - 1536*a^5*b^13*c*j*k*m*z + 165150720*a^9*b*c^9*d*g*m*z + 34681
6512*a^7*b*c^11*d^2*g*z + 19660800*a^12*b*c^6*l*m^2*z - 34560*a^7*b^11*c*l*
m^2*z - 7077888*a^11*b*c^7*k^2*l*z + 11008*a^6*b^12*c*j*m^2*z + 19660800*a^
11*b*c^7*g*m^2*z + 7077888*a^10*b*c^8*h^2*l*z + 768*a^5*b^13*c*g*m^2*z - 19
660800*a^9*b*c^9*f^2*l*z - 7077888*a^10*b*c^8*g*k^2*z - 6912*a*b^15*c^3*d^2
*l*z + 7077888*a^9*b*c^9*g*h^2*z - 19660800*a^8*b*c^10*f^2*g*z - 66816*a*b^
14*c^4*d^2*j*z + 214272*a*b^13*c^5*d^2*g*z - 428544*a*b^12*c^6*d^2*e*z - 33
0301440*a^9*c^10*d*e*m*z - 110100480*a^10*c^9*d*j*m*z - 15728640*a^11*c^8*h
*j*m*z - 47185920*a^10*c^9*e*h*m*z - 198180864*a^8*c^11*d*e*h*z + 15728640*
a^10*c^9*f*j*k*z - 66060288*a^9*c^10*d*h*j*z + 47185920*a^9*c^10*e*f*k*z +
1022754816*a^6*b^2*c^11*d^2*e*z - 642318336*a^5*b^4*c^10*d^2*e*z - 51137740
8*a^7*b^3*c^9*d^2*l*z - 511377408*a^6*b^3*c^10*d^2*g*z + 321159168*a^6*b^5*
c^8*d^2*l*z + 321159168*a^5*b^5*c^9*d^2*g*z + 225312768*a^7*b^2*c^10*d^2*j*
z - 25362432*a^11*b^3*c^5*l*m^2*z + 13271040*a^10*b^5*c^4*l*m^2*z - 3563520
*a^9*b^7*c^3*l*m^2*z + 506880*a^8*b^9*c^2*l*m^2*z + 10354688*a^11*b^2*c^6*j
*m^2*z + 8847360*a^10*b^3*c^6*k^2*l*z - 4423680*a^9*b^5*c^5*k^2*l*z - 20480
00*a^9*b^6*c^4*j*m^2*z + 1105920*a^8*b^7*c^4*k^2*l*z + 849920*a^8*b^8*c^3*j
*m^2*z - 393216*a^10*b^4*c^5*j*m^2*z - 145920*a^7*b^10*c^2*j*m^2*z - 138240
*a^7*b^9*c^3*k^2*l*z + 6912*a^6*b^11*c^2*k^2*l*z - 111697920*a^5*b^7*c^7*d^
2*l*z + 223395840*a^4*b^6*c^9*d^2*e*z - 25362432*a^10*b^3*c^6*g*m^2*z - 353
8944*a^10*b^2*c^7*j*k^2*z + 737280*a^8*b^6*c^5*j*k^2*z + 50724864*a^10*b^2*
c^7*e*m^2*z - 276480*a^7*b^8*c^4*j*k^2*z + 41472*a^6*b^10*c^3*j*k^2*z - 230
4*a^5*b^12*c^2*j*k^2*z + 13271040*a^9*b^5*c^5*g*m^2*z - 8847360*a^9*b^3*c^7
*h^2*l*z + 4423680*a^8*b^5*c^6*h^2*l*z - 3563520*a^8*b^7*c^4*g*m^2*z - 1105
920*a^7*b^7*c^5*h^2*l*z + 506880*a^7*b^9*c^3*g*m^2*z + 138240*a^6*b^9*c^4*h
^2*l*z - 34560*a^6*b^11*c^2*g*m^2*z - 6912*a^5*b^11*c^3*h^2*l*z - 26542080*
a^9*b^4*c^6*e*m^2*z + 25362432*a^8*b^3*c^8*f^2*l*z - 13271040*a^7*b^5*c^7*f

^2*l*z + 8847360*a^9*b^3*c^7*g*k^2*z + 7127040*a^8*b^6*c^5*e*m^2*z - 442368
0*a^8*b^5*c^6*g*k^2*z + 3563520*a^6*b^7*c^6*f^2*l*z + 3538944*a^9*b^2*c^8*h
^2*j*z + 1105920*a^7*b^7*c^5*g*k^2*z - 1013760*a^7*b^8*c^4*e*m^2*z - 737280
*a^7*b^6*c^6*h^2*j*z - 506880*a^5*b^9*c^5*f^2*l*z + 276480*a^6*b^8*c^5*h^2*
j*z - 138240*a^6*b^9*c^4*g*k^2*z + 69120*a^6*b^10*c^3*e*m^2*z - 41472*a^5*b
^10*c^4*h^2*j*z + 34560*a^4*b^11*c^4*f^2*l*z + 6912*a^5*b^11*c^3*g*k^2*z +
2304*a^4*b^12*c^3*h^2*j*z - 1536*a^5*b^12*c^2*e*m^2*z - 768*a^3*b^13*c^3*f^
2*l*z - 111697920*a^4*b^7*c^8*d^2*g*z + 23362560*a^4*b^9*c^6*d^2*l*z - 1769
4720*a^9*b^2*c^8*e*k^2*z - 10354688*a^8*b^2*c^9*f^2*j*z - 43646976*a^6*b^4*
c^9*d^2*j*z + 8847360*a^8*b^4*c^7*e*k^2*z - 2965248*a^3*b^11*c^5*d^2*l*z -
2211840*a^7*b^6*c^6*e*k^2*z + 2048000*a^6*b^6*c^7*f^2*j*z - 849920*a^5*b^8*
c^6*f^2*j*z + 393216*a^7*b^4*c^8*f^2*j*z + 276480*a^6*b^8*c^5*e*k^2*z + 214
272*a^2*b^13*c^4*d^2*l*z + 145920*a^4*b^10*c^5*f^2*j*z - 13824*a^5*b^10*c^4
*e*k^2*z - 11008*a^3*b^12*c^4*f^2*j*z + 256*a^2*b^14*c^3*f^2*j*z - 32587776
*a^5*b^6*c^8*d^2*j*z - 8847360*a^8*b^3*c^8*g*h^2*z + 21657600*a^4*b^8*c^7*d
^2*j*z + 4423680*a^7*b^5*c^7*g*h^2*z - 1105920*a^6*b^7*c^6*g*h^2*z + 138240
*a^5*b^9*c^5*g*h^2*z - 6912*a^4*b^11*c^4*g*h^2*z + 25362432*a^7*b^3*c^9*f^2
*g*z - 5810688*a^3*b^10*c^6*d^2*j*z + 17694720*a^8*b^2*c^9*e*h^2*z + 845568
*a^2*b^12*c^5*d^2*j*z - 50724864*a^7*b^2*c^10*e*f^2*z - 13271040*a^6*b^5*c^
8*f^2*g*z - 8847360*a^7*b^4*c^8*e*h^2*z + 3563520*a^5*b^7*c^7*f^2*g*z + 221
1840*a^6*b^6*c^7*e*h^2*z - 506880*a^4*b^9*c^6*f^2*g*z - 276480*a^5*b^8*c^6*
e*h^2*z + 34560*a^3*b^11*c^5*f^2*g*z + 13824*a^4*b^10*c^5*e*h^2*z - 768*a^2
*b^13*c^4*f^2*g*z + 26542080*a^6*b^4*c^9*e*f^2*z + 23362560*a^3*b^9*c^7*d^2
*g*z - 46725120*a^3*b^8*c^8*d^2*e*z - 7127040*a^5*b^6*c^8*e*f^2*z - 2965248
*a^2*b^11*c^6*d^2*g*z + 1013760*a^4*b^8*c^7*e*f^2*z - 69120*a^3*b^10*c^6*e*
f^2*z + 1536*a^2*b^12*c^5*e*f^2*z + 5930496*a^2*b^10*c^7*d^2*e*z + 34681651
2*a^8*b*c^10*d^2*l*z - 693633024*a^7*c^12*d^2*e*z - 231211008*a^8*c^11*d^2*
j*z + 768*a^6*b^13*l*m^2*z - 13107200*a^12*c^7*j*m^2*z - 256*a^5*b^14*j*m^2
*z + 4718592*a^11*c^8*j*k^2*z - 39321600*a^11*c^8*e*m^2*z - 4718592*a^10*c^
9*h^2*j*z + 14155776*a^10*c^9*e*k^2*z + 13107200*a^9*c^10*f^2*j*z + 2304*b^
16*c^3*d^2*j*z - 14155776*a^9*c^10*e*h^2*z + 39321600*a^8*c^11*e*f^2*z - 69
12*b^15*c^4*d^2*g*z + 13824*b^14*c^5*d^2*e*z + 737280*a^10*b*c^5*j*k*l*m -
2304*a^6*b^9*c*j*k*l*m + 2211840*a^9*b*c^6*e*k*l*m + 1228800*a^9*b*c^6*f*j*
l*m + 737280*a^9*b*c^6*g*j*k*m + 442368*a^9*b*c^6*h*j*k*l + 36*a^3*b^12*c*f
*h*k*m + 3096576*a^8*b*c^7*d*j*k*l - 12745728*a^8*b*c^7*d*h*k*m + 3686400*a
^8*b*c^7*e*f*l*m + 3391488*a^8*b*c^7*e*h*j*m + 2211840*a^8*b*c^7*e*g*k*m +
1327104*a^8*b*c^7*e*h*k*l + 1228800*a^8*b*c^7*f*g*j*m + 737280*a^8*b*c^7*f*
h*j*l + 442368*a^8*b*c^7*g*h*j*k + 108*a^2*b^13*c*d*h*k*m + 16367616*a^7*b*
c^8*d*e*j*m + 9289728*a^7*b*c^8*d*e*k*l + 5160960*a^7*b*c^8*d*f*j*l + 33914
88*a^7*b*c^8*e*f*j*k + 3096576*a^7*b*c^8*d*g*j*k - 19307520*a^7*b*c^8*d*f*h
*m + 3686400*a^7*b*c^8*e*f*g*m + 2211840*a^7*b*c^8*e*f*h*l + 1327104*a^7*b*
c^8*e*g*h*k + 737280*a^7*b*c^8*f*g*h*j - 180*a*b^13*c^2*d*f*h*m - 540*a*b^1
2*c^3*d*f*h*k + 15482880*a^6*b*c^9*d*e*f*l + 11059200*a^6*b*c^9*d*e*h*j + 9
289728*a^6*b*c^9*d*e*g*k + 5160960*a^6*b*c^9*d*f*g*j - 2304*a*b^11*c^4*d*f*
g*j + 2211840*a^6*b*c^9*e*f*g*h + 4608*a*b^10*c^5*d*e*f*j + 15482880*a^5*b*

c^10*d*e*f*g - 13824*a*b^9*c^6*d*e*f*g + 36*a*b^14*c*d*f*k*m + 1843200*a^9*
b^3*c^4*j*k*l*m + 783360*a^8*b^5*c^3*j*k*l*m + 18432*a^7*b^7*c^2*j*k*l*m -
2211840*a^8*b^4*c^4*g*k*l*m - 1695744*a^9*b^2*c^5*h*j*l*m - 1400832*a^8*b^4
*c^4*h*j*l*m - 1105920*a^9*b^2*c^5*g*k*l*m - 253440*a^7*b^6*c^3*h*j*l*m - 6
9120*a^7*b^6*c^3*g*k*l*m + 11520*a^6*b^8*c^2*h*j*l*m + 6912*a^6*b^8*c^2*g*k
*l*m + 4423680*a^8*b^3*c^5*e*k*l*m + 2506752*a^8*b^3*c^5*f*j*l*m + 1843200*
a^8*b^3*c^5*g*j*k*m + 1327104*a^8*b^3*c^5*h*j*k*l + 838656*a^7*b^5*c^4*f*j*
l*m + 783360*a^7*b^5*c^4*g*j*k*m + 691200*a^7*b^5*c^4*h*j*k*l + 138240*a^7*
b^5*c^4*e*k*l*m + 69120*a^6*b^7*c^3*h*j*k*l - 53760*a^6*b^7*c^3*f*j*l*m + 1
8432*a^6*b^7*c^3*g*j*k*m - 13824*a^6*b^7*c^3*e*k*l*m - 2304*a^5*b^9*c^2*g*j
*k*m + 2543616*a^8*b^3*c^5*g*h*l*m + 829440*a^7*b^5*c^4*g*h*l*m - 34560*a^6
*b^7*c^3*g*h*l*m - 8183808*a^8*b^2*c^6*d*j*l*m - 3686400*a^8*b^2*c^6*e*j*k*
m - 2285568*a^7*b^4*c^5*d*j*l*m - 1695744*a^8*b^2*c^6*f*j*k*l - 1566720*a^7
*b^4*c^5*e*j*k*m - 1400832*a^7*b^4*c^5*f*j*k*l + 741888*a^6*b^6*c^4*d*j*l*m
 - 253440*a^6*b^6*c^4*f*j*k*l - 80640*a^5*b^8*c^3*d*j*l*m - 36864*a^6*b^6*c
^4*e*j*k*m + 11520*a^5*b^8*c^3*f*j*k*l + 4608*a^5*b^8*c^3*e*j*k*m + 6700032
*a^8*b^2*c^6*f*h*k*m + 5103360*a^7*b^4*c^5*f*h*k*m - 5087232*a^8*b^2*c^6*e*
h*l*m - 2838528*a^7*b^4*c^5*f*g*l*m - 1843200*a^8*b^2*c^6*f*g*l*m - 1695744
*a^8*b^2*c^6*g*h*j*m - 1658880*a^7*b^4*c^5*g*h*k*l - 1658880*a^7*b^4*c^5*e*
h*l*m - 1400832*a^7*b^4*c^5*g*h*j*m - 663552*a^8*b^2*c^6*g*h*k*l + 483840*a
^6*b^6*c^4*f*h*k*m - 253440*a^6*b^6*c^4*g*h*j*m - 207360*a^6*b^6*c^4*g*h*k*
l + 161280*a^6*b^6*c^4*f*g*l*m + 69120*a^6*b^6*c^4*e*h*l*m - 50040*a^5*b^8*
c^3*f*h*k*m + 11520*a^5*b^8*c^3*g*h*j*m + 180*a^4*b^10*c^2*f*h*k*m + 420249
6*a^7*b^3*c^6*d*j*k*l + 635904*a^6*b^5*c^5*d*j*k*l - 276480*a^5*b^7*c^4*d*j
*k*l + 34560*a^4*b^9*c^3*d*j*k*l - 16671744*a^7*b^3*c^6*d*h*k*m + 12275712*
a^7*b^3*c^6*d*g*l*m + 5677056*a^7*b^3*c^6*e*f*l*m + 4423680*a^7*b^3*c^6*e*g
*k*m + 3317760*a^7*b^3*c^6*e*h*k*l + 2801664*a^7*b^3*c^6*e*h*j*m - 2709504*
a^6*b^5*c^5*d*g*l*m + 2543616*a^7*b^3*c^6*f*g*k*l + 2506752*a^7*b^3*c^6*f*g
*j*m + 1843200*a^7*b^3*c^6*f*h*j*l + 1327104*a^7*b^3*c^6*g*h*j*k + 838656*a
^6*b^5*c^5*f*g*j*m + 829440*a^6*b^5*c^5*f*g*k*l + 783360*a^6*b^5*c^5*f*h*j*
l + 691200*a^6*b^5*c^5*g*h*j*k + 665280*a^5*b^7*c^4*d*h*k*m + 506880*a^6*b^
5*c^5*e*h*j*m + 414720*a^6*b^5*c^5*e*h*k*l - 322560*a^6*b^5*c^5*e*f*l*m + 2
41920*a^5*b^7*c^4*d*g*l*m + 138240*a^6*b^5*c^5*e*g*k*m - 108540*a^4*b^9*c^3
*d*h*k*m + 69120*a^5*b^7*c^4*g*h*j*k - 53760*a^5*b^7*c^4*f*g*j*m - 51840*a^
6*b^5*c^5*d*h*k*m - 34560*a^5*b^7*c^4*f*g*k*l - 23040*a^5*b^7*c^4*e*h*j*m +
 18432*a^5*b^7*c^4*f*h*j*l - 13824*a^5*b^7*c^4*e*g*k*m - 2304*a^4*b^9*c^3*f
*h*j*l + 1296*a^3*b^11*c^2*d*h*k*m + 31924224*a^7*b^2*c^7*d*f*k*m - 2455142
4*a^7*b^2*c^7*d*e*l*m + 10616832*a^7*b^2*c^7*e*g*j*l - 8183808*a^7*b^2*c^7*
d*g*j*m - 5529600*a^7*b^2*c^7*d*h*j*l + 5419008*a^6*b^4*c^6*d*e*l*m + 53084
16*a^6*b^4*c^6*e*g*j*l - 5087232*a^7*b^2*c^7*e*f*k*l - 5013504*a^7*b^2*c^7*
e*f*j*m + 4868352*a^6*b^4*c^6*d*f*k*m - 4644864*a^7*b^2*c^7*d*g*k*l - 39813
12*a^6*b^4*c^6*d*g*k*l - 2654208*a^7*b^2*c^7*e*h*j*k - 2367360*a^5*b^6*c^5*
d*f*k*m - 2285568*a^6*b^4*c^6*d*g*j*m - 2211840*a^6*b^4*c^6*d*h*j*l - 16957
44*a^7*b^2*c^7*f*g*j*k - 1677312*a^6*b^4*c^6*e*f*j*m - 1658880*a^6*b^4*c^6*
e*f*k*l - 1400832*a^6*b^4*c^6*f*g*j*k - 1382400*a^6*b^4*c^6*e*h*j*k + 10368

00*a^5*b^6*c^5*d*g*k*l + 741888*a^5*b^6*c^5*d*g*j*m - 483840*a^5*b^6*c^5*d*
e*l*m + 317952*a^5*b^6*c^5*d*h*j*l + 268920*a^4*b^8*c^4*d*f*k*m - 253440*a^
5*b^6*c^5*f*g*j*k - 138240*a^5*b^6*c^5*e*h*j*k + 107520*a^5*b^6*c^5*e*f*j*m
 - 103680*a^4*b^8*c^4*d*g*k*l - 80640*a^4*b^8*c^4*d*g*j*m + 69120*a^5*b^6*c
^5*e*f*k*l + 11520*a^4*b^8*c^4*f*g*j*k + 6912*a^4*b^8*c^4*d*h*j*l - 6912*a^
3*b^10*c^3*d*h*j*l + 6120*a^3*b^10*c^3*d*f*k*m - 1368*a^2*b^12*c^2*d*f*k*m
- 5087232*a^7*b^2*c^7*e*g*h*m - 2211840*a^6*b^4*c^6*f*g*h*l - 1658880*a^6*b
^4*c^6*e*g*h*m - 1105920*a^7*b^2*c^7*f*g*h*l - 69120*a^5*b^6*c^5*f*g*h*l +
69120*a^5*b^6*c^5*e*g*h*m + 6912*a^4*b^8*c^4*f*g*h*l + 7962624*a^6*b^3*c^7*
d*e*k*l - 22164480*a^6*b^3*c^7*d*f*h*m + 5160960*a^6*b^3*c^7*d*f*j*l + 4571
136*a^6*b^3*c^7*d*e*j*m + 4202496*a^6*b^3*c^7*d*g*j*k + 2801664*a^6*b^3*c^7
*e*f*j*k - 2073600*a^5*b^5*c^6*d*e*k*l - 1483776*a^5*b^5*c^6*d*e*j*m + 6359
04*a^5*b^5*c^6*d*g*j*k + 506880*a^5*b^5*c^6*e*f*j*k - 354816*a^4*b^7*c^5*d*
f*j*l + 322560*a^5*b^5*c^6*d*f*j*l - 276480*a^4*b^7*c^5*d*g*j*k + 207360*a^
4*b^7*c^5*d*e*k*l + 161280*a^4*b^7*c^5*d*e*j*m + 59904*a^3*b^9*c^4*d*f*j*l
+ 34560*a^3*b^9*c^4*d*g*j*k - 23040*a^4*b^7*c^5*e*f*j*k - 2304*a^2*b^11*c^3
*d*f*j*l + 8294400*a^6*b^3*c^7*d*g*h*l + 5677056*a^6*b^3*c^7*e*f*g*m + 4423
680*a^6*b^3*c^7*e*f*h*l + 3317760*a^6*b^3*c^7*e*g*h*k + 2805120*a^5*b^5*c^6
*d*f*h*m + 1843200*a^6*b^3*c^7*f*g*h*j - 829440*a^5*b^5*c^6*d*g*h*l + 78336
0*a^5*b^5*c^6*f*g*h*j + 437184*a^4*b^7*c^5*d*f*h*m + 414720*a^5*b^5*c^6*e*g
*h*k - 322560*a^5*b^5*c^6*e*f*g*m - 146268*a^3*b^9*c^4*d*f*h*m + 138240*a^5
*b^5*c^6*e*f*h*l - 62208*a^4*b^7*c^5*d*g*h*l + 20736*a^3*b^9*c^4*d*g*h*l +
18432*a^4*b^7*c^5*f*g*h*j - 13824*a^4*b^7*c^5*e*f*h*l + 9360*a^2*b^11*c^3*d
*f*h*m - 2304*a^3*b^9*c^4*f*g*h*j - 8404992*a^6*b^2*c^8*d*e*j*k - 24551424*
a^6*b^2*c^8*d*e*g*m + 21150720*a^6*b^2*c^8*d*f*h*k - 1271808*a^5*b^4*c^7*d*
e*j*k + 552960*a^4*b^6*c^6*d*e*j*k - 69120*a^3*b^8*c^5*d*e*j*k - 16588800*a
^6*b^2*c^8*d*e*h*l - 7741440*a^6*b^2*c^8*d*f*g*l + 6946560*a^5*b^4*c^7*d*f*
h*k - 5529600*a^6*b^2*c^8*d*g*h*j + 5419008*a^5*b^4*c^7*d*e*g*m - 5087232*a
^6*b^2*c^8*e*f*g*k - 3870720*a^5*b^4*c^7*d*f*g*l - 3686400*a^6*b^2*c^8*e*f*
h*j - 2211840*a^5*b^4*c^7*d*g*h*j - 1755648*a^4*b^6*c^6*d*f*h*k - 1658880*a
^5*b^4*c^7*e*f*g*k + 1658880*a^5*b^4*c^7*d*e*h*l - 1566720*a^5*b^4*c^7*e*f*
h*j + 1451520*a^4*b^6*c^6*d*f*g*l - 483840*a^4*b^6*c^6*d*e*g*m + 317952*a^4
*b^6*c^6*d*g*h*j - 193536*a^3*b^8*c^5*d*f*g*l + 124416*a^4*b^6*c^6*d*e*h*l
+ 114696*a^3*b^8*c^5*d*f*h*k + 69120*a^4*b^6*c^6*e*f*g*k - 41472*a^3*b^8*c^
5*d*e*h*l - 36864*a^4*b^6*c^6*e*f*h*j + 14580*a^2*b^10*c^4*d*f*h*k + 6912*a
^3*b^8*c^5*d*g*h*j - 6912*a^2*b^10*c^4*d*g*h*j + 6912*a^2*b^10*c^4*d*f*g*l
+ 4608*a^3*b^8*c^5*e*f*h*j + 7962624*a^5*b^3*c^8*d*e*g*k + 7741440*a^5*b^3*
c^8*d*e*f*l + 5160960*a^5*b^3*c^8*d*f*g*j + 4423680*a^5*b^3*c^8*d*e*h*j - 2
903040*a^4*b^5*c^7*d*e*f*l - 2073600*a^4*b^5*c^7*d*e*g*k - 635904*a^4*b^5*c
^7*d*e*h*j + 387072*a^3*b^7*c^6*d*e*f*l - 354816*a^3*b^7*c^6*d*f*g*j + 3225
60*a^4*b^5*c^7*d*f*g*j + 207360*a^3*b^7*c^6*d*e*g*k + 59904*a^2*b^9*c^5*d*f
*g*j - 13824*a^3*b^7*c^6*d*e*h*j + 13824*a^2*b^9*c^5*d*e*h*j - 13824*a^2*b^
9*c^5*d*e*f*l + 4423680*a^5*b^3*c^8*e*f*g*h + 138240*a^4*b^5*c^7*e*f*g*h -
13824*a^3*b^7*c^6*e*f*g*h - 10321920*a^5*b^2*c^9*d*e*f*j + 709632*a^3*b^6*c
^7*d*e*f*j - 645120*a^4*b^4*c^8*d*e*f*j - 119808*a^2*b^8*c^6*d*e*f*j - 1658

8800*a^5*b^2*c^9*d*e*g*h + 1658880*a^4*b^4*c^8*d*e*g*h + 124416*a^3*b^6*c^7
*d*e*g*h - 41472*a^2*b^8*c^6*d*e*g*h + 7741440*a^4*b^3*c^9*d*e*f*g - 290304
0*a^3*b^5*c^8*d*e*f*g + 387072*a^2*b^7*c^7*d*e*f*g + 3456*a^7*b^8*c*k*l^2*m
 + 12672*a^7*b^8*c*j*l*m^2 + 384*a^5*b^10*c*j^2*k*m - 1635840*a^10*b*c^5*h*
k*m^2 - 1009152*a^9*b*c^6*h^2*k*m + 3690*a^6*b^9*c*h*k*m^2 + 1152*a^6*b^9*c
*g*l*m^2 - 540*a^5*b^10*c*h*k^2*m + 54*a^4*b^11*c*h^2*k*m + 565248*a^9*b*c^
6*h*j^2*m - 39771648*a^7*b*c^8*d^2*k*m - 2496000*a^8*b*c^7*f^2*k*m - 154368
0*a^9*b*c^6*f*k^2*m + 1980*a^5*b^10*c*f*k*m^2 - 384*a^5*b^10*c*g*j*m^2 - 18
0*a^4*b^11*c*f*k^2*m + 6*a^2*b^13*c*f^2*k*m - 10298880*a^9*b*c^6*d*k*m^2 +
2580480*a^9*b*c^6*e*j*m^2 + 5310*a^4*b^11*c*d*k*m^2 - 1674*a*b^13*c^2*d^2*k
*m - 540*a^3*b^12*c*d*k^2*m - 10616832*a^7*b*c^8*e^2*j*l - 3538944*a^8*b*c^
7*e*j^2*l + 2727936*a^8*b*c^7*d*j^2*m - 2496000*a^9*b*c^6*f*h*m^2 - 1543680
*a^8*b*c^7*f*h^2*m + 565248*a^8*b*c^7*f*j^2*k - 270*a^4*b^11*c*f*h*m^2 - 59
512320*a^6*b*c^9*d^2*f*m + 5087232*a^7*b*c^8*e^2*h*m + 1105920*a^8*b*c^7*e*
j*k^2 - 3456*a*b^12*c^3*d^2*j*l - 1635840*a^7*b*c^8*f^2*h*k - 1009152*a^8*b
*c^7*f*h*k^2 + 10260*a*b^12*c^3*d^2*h*m - 684*a^3*b^12*c*d*h*m^2 - 24675840
*a^6*b*c^9*d^2*h*k - 15552000*a^8*b*c^7*d*f*m^2 + 24551424*a^6*b*c^9*d*e^2*
m - 3939840*a^7*b*c^8*d*h^2*k + 1105920*a^7*b*c^8*e*h^2*j - 25074*a*b^11*c^
4*d^2*f*m + 10530*a*b^11*c^4*d^2*h*k + 10368*a*b^11*c^4*d^2*g*l + 420*a*b^1
2*c^3*d*f^2*m - 378*a^2*b^13*c*d*f*m^2 - 10616832*a^6*b*c^9*e^2*g*j + 50872
32*a^6*b*c^9*e^2*f*k - 3538944*a^7*b*c^8*e*g*j^2 + 1843200*a^7*b*c^8*d*h*j^
2 - 7994880*a^6*b*c^9*d*f^2*k - 4990464*a^7*b*c^8*d*f*k^2 + 2580480*a^6*b*c
^9*e*f^2*j + 65664*a*b^10*c^5*d^2*g*j - 27972*a*b^10*c^5*d^2*f*k - 20736*a*
b^10*c^5*d^2*e*l + 1260*a*b^11*c^4*d*f^2*k + 54*a*b^13*c^2*d*f*k^2 + 232243
20*a^5*b*c^10*d^2*e*j - 37062144*a^5*b*c^10*d^2*f*h + 384*a*b^12*c^3*d*f*j^
2 - 131328*a*b^9*c^6*d^2*e*j - 5985792*a^6*b*c^9*d*f*h^2 + 206010*a*b^9*c^6
*d^2*f*h - 6300*a*b^10*c^5*d*f^2*h + 1350*a*b^11*c^4*d*f*h^2 + 16588800*a^5
*b*c^10*d*e^2*h + 3456*a*b^10*c^5*d*f*g^2 + 435456*a*b^8*c^7*d^2*e*g + 1382
4*a*b^8*c^7*d*e^2*f - 1474560*a^9*c^7*e*j*k*m + 460800*a^9*c^7*f*h*k*m + 32
25600*a^8*c^8*d*f*k*m - 2457600*a^8*c^8*e*f*j*m - 884736*a^8*c^8*e*h*j*k -
6193152*a^7*c^9*d*e*j*k + 1935360*a^7*c^9*d*f*h*k - 1474560*a^7*c^9*e*f*h*j
 - 10321920*a^6*c^10*d*e*f*j - 1105920*a^9*b^4*c^3*k*l^2*m - 552960*a^10*b^
2*c^4*k*l^2*m - 34560*a^8*b^6*c^2*k*l^2*m - 1290240*a^10*b^2*c^4*j*l*m^2 -
860160*a^9*b^4*c^3*j*l*m^2 - 80640*a^8*b^6*c^2*j*l*m^2 - 737280*a^9*b^2*c^5
*j^2*k*m - 568320*a^8*b^4*c^4*j^2*k*m - 136704*a^7*b^6*c^3*j^2*k*m - 2304*a
^6*b^8*c^2*j^2*k*m + 1271808*a^9*b^3*c^4*h*l^2*m - 552960*a^9*b^2*c^5*j*k^2
*l - 552960*a^8*b^4*c^4*j*k^2*l + 414720*a^8*b^5*c^3*h*l^2*m - 145152*a^7*b
^6*c^3*j*k^2*l - 17280*a^7*b^7*c^2*h*l^2*m - 3456*a^6*b^8*c^2*j*k^2*l - 364
0320*a^9*b^3*c^4*h*k*m^2 - 2626560*a^8*b^3*c^5*h^2*k*m + 2211840*a^9*b^2*c^
5*h*k^2*m + 2056320*a^8*b^4*c^4*h*k^2*m + 1935360*a^9*b^3*c^4*g*l*m^2 - 114
3360*a^8*b^5*c^3*h*k*m^2 - 1097280*a^7*b^5*c^4*h^2*k*m + 364608*a^7*b^6*c^3
*h*k^2*m + 322560*a^8*b^5*c^3*g*l*m^2 - 56160*a^6*b^7*c^3*h^2*k*m - 40320*a
^7*b^7*c^2*g*l*m^2 + 27936*a^7*b^7*c^2*h*k*m^2 - 3780*a^6*b^8*c^2*h*k^2*m +
 2970*a^5*b^9*c^2*h^2*k*m - 1419264*a^8*b^4*c^4*f*l^2*m - 1105920*a^7*b^4*c
^5*g^2*k*m - 921600*a^9*b^2*c^5*f*l^2*m - 829440*a^8*b^4*c^4*h*k*l^2 + 7495

68*a^8*b^3*c^5*h*j^2*m - 552960*a^8*b^2*c^6*g^2*k*m - 331776*a^9*b^2*c^5*h*k*l^2 + 317952*a^7*b^5*c^4*h*j^2*m - 103680*a^7*b^6*c^3*h*k*l^2 + 80640*a^7*b^6*c^3*f*l^2*m + 38400*a^6*b^7*c^3*h*j^2*m - 34560*a^6*b^6*c^4*g^2*k*m + 3456*a^5*b^8*c^3*g^2*k*m - 1920*a^5*b^9*c^2*h*j^2*m - 5142528*a^7*b^3*c^6*f^2*k*m + 5068800*a^9*b^2*c^5*f*k*m^2 - 3870720*a^9*b^2*c^5*e*l*m^2 - 3755520*a^8*b^3*c^5*f*k^2*m + 3000960*a^8*b^4*c^4*f*k*m^2 - 1290240*a^9*b^2*c^5*g*j*m^2 - 1085760*a^7*b^5*c^4*f*k^2*m - 959040*a^6*b^5*c^5*f^2*k*m - 860160*a^8*b^4*c^4*g*j*m^2 + 829440*a^8*b^3*c^5*g*k^2*l - 645120*a^8*b^4*c^4*e*l*m^2 - 552960*a^8*b^2*c^6*h^2*j*l - 552960*a^7*b^4*c^5*h^2*j*l + 414720*a^7*b^5*c^4*g*k^2*l - 145152*a^6*b^6*c^4*h^2*j*l + 103200*a^5*b^7*c^4*f^2*k*m - 80640*a^7*b^6*c^3*g*j*m^2 + 80640*a^7*b^6*c^3*e*l*m^2 + 41280*a^7*b^6*c^3*f*k*m^2 - 37188*a^6*b^8*c^2*f*k*m^2 + 13536*a^6*b^7*c^3*f*k^2*m + 12672*a^6*b^8*c^2*g*j*m^2 + 10368*a^6*b^7*c^3*g*k^2*l + 5490*a^5*b^9*c^2*f*k^2*m - 3456*a^5*b^8*c^3*h^2*j*l - 2304*a^6*b^8*c^2*e*l*m^2 + 810*a^4*b^9*c^3*f^2*k*m - 270*a^3*b^11*c^2*f^2*k*m + 6137856*a^8*b^3*c^5*d*l^2*m - 4423680*a^7*b^2*c^7*e^2*k*m - 2654208*a^8*b^3*c^5*g*j*l^2 - 2654208*a^7*b^3*c^6*g^2*j*l + 1769472*a^8*b^2*c^6*g*j^2*l + 1769472*a^7*b^4*c^5*g*j^2*l - 1354752*a^7*b^5*c^4*d*l^2*m - 1327104*a^7*b^5*c^4*g*j*l^2 - 1327104*a^6*b^5*c^5*g^2*j*l + 1271808*a^8*b^3*c^5*f*k*l^2 - 1040384*a^8*b^2*c^6*f*j^2*m - 697344*a^7*b^4*c^5*f*j^2*m - 516096*a^8*b^2*c^6*h*j^2*k - 451584*a^7*b^4*c^5*h*j^2*k + 442368*a^6*b^6*c^4*g*j^2*l + 414720*a^7*b^5*c^4*f*k*l^2 - 138240*a^6*b^6*c^4*h*j^2*k - 138240*a^6*b^4*c^6*e^2*k*m - 121856*a^6*b^6*c^4*f*j^2*m + 120960*a^6*b^7*c^3*d*l^2*m - 17280*a^6*b^7*c^3*f*k*l^2 + 13824*a^5*b^6*c^5*e^2*k*m - 11520*a^5*b^8*c^3*h*j^2*k + 8960*a^5*b^8*c^3*f*j^2*m + 10851840*a^8*b^2*c^6*d*k^2*m - 10464768*a^6*b^3*c^7*d^2*k*m - 10275840*a^8*b^3*c^5*d*k*m^2 + 7121088*a^5*b^5*c^6*d^2*k*m + 3127680*a^7*b^4*c^5*d*k^2*m + 1720320*a^8*b^3*c^5*e*j*m^2 - 1658880*a^8*b^2*c^6*e*k^2*l - 1290240*a^7*b^2*c^7*f^2*j*l + 1271808*a^7*b^3*c^6*g^2*h*m - 1222560*a^4*b^7*c^5*d^2*k*m + 999360*a^7*b^5*c^4*d*k*m^2 - 860160*a^6*b^4*c^6*f^2*j*l - 829440*a^7*b^4*c^5*e*k^2*l - 705024*a^6*b^6*c^4*d*k^2*m - 552960*a^8*b^2*c^6*g*j*k^2 - 552960*a^7*b^4*c^5*g*j*k^2 + 414720*a^6*b^5*c^5*g^2*h*m + 319392*a^6*b^7*c^3*d*k*m^2 + 161280*a^7*b^5*c^4*e*j*m^2 - 145152*a^6*b^6*c^4*g*j*k^2 - 85734*a^5*b^9*c^2*d*k*m^2 - 80640*a^5*b^6*c^5*f^2*j*l - 25344*a^6*b^7*c^3*e*j*m^2 + 23490*a^3*b^9*c^4*d^2*k*m - 20736*a^6*b^6*c^4*e*k^2*l - 17280*a^5*b^7*c^4*g^2*h*m + 14148*a^5*b^8*c^3*d*k^2*m + 13716*a^2*b^11*c^3*d^2*k*m + 12690*a^4*b^10*c^2*d*k^2*m + 12672*a^4*b^8*c^4*f^2*j*l - 3456*a^5*b^8*c^3*g*j*k^2 + 768*a^5*b^9*c^2*e*j*m^2 - 384*a^3*b^10*c^3*f^2*j*l + 5308416*a^8*b^2*c^6*e*j*l^2 - 5308416*a^6*b^3*c^7*e^2*j*l - 5142528*a^8*b^3*c^5*f*h*m^2 + 5068800*a^7*b^2*c^7*f^2*h*m - 3755520*a^7*b^3*c^6*f*h^2*m - 3538944*a^7*b^3*c^6*e*j^2*l + 3000960*a^6*b^4*c^6*f^2*h*m + 2654208*a^7*b^4*c^5*e*j*l^2 - 2322432*a^8*b^2*c^6*d*k*l^2 + 2125824*a^7*b^3*c^6*d*j^2*m - 1990656*a^7*b^4*c^5*d*k*l^2 - 1085760*a^6*b^5*c^5*f*h^2*m - 959040*a^7*b^5*c^4*f*h*m^2 - 884736*a^6*b^5*c^5*e*j^2*l + 829440*a^7*b^3*c^6*g*h^2*l + 749568*a^7*b^3*c^6*f*j^2*k + 518400*a^6*b^6*c^4*d*k*l^2 + 414720*a^6*b^5*c^5*g*h^2*l + 317952*a^6*b^5*c^5*f*j^2*k + 133632*a^6*b^5*c^5*d*j^2*m + 103200*a^6*b^7*c^3*f*h*m^2 - 96768*a^5*b^7*c^4

*d*j^2*m - 51840*a^5*b^8*c^3*d*k*l^2 + 41280*a^5*b^6*c^5*f^2*h*m + 38400*a^
5*b^7*c^4*f*j^2*k - 37188*a^4*b^8*c^4*f^2*h*m + 13536*a^5*b^7*c^4*f*h^2*m +
13440*a^4*b^9*c^3*d*j^2*m + 10368*a^5*b^7*c^4*g*h^2*l + 5490*a^4*b^9*c^3*f
*h^2*m + 1980*a^3*b^10*c^3*f^2*h*m - 1920*a^4*b^9*c^3*f*j^2*k + 810*a^5*b^9
*c^2*f*h*m^2 - 180*a^3*b^11*c^2*f*h^2*m - 30*a^2*b^12*c^2*f^2*h*m + 3006720
0*a^6*b^2*c^8*d^2*h*m - 11612160*a^6*b^2*c^8*d^2*j*l + 1658880*a^6*b^3*c^7*
e^2*h*m + 1596672*a^4*b^6*c^6*d^2*j*l - 1419264*a^6*b^4*c^6*f*g^2*m - 11059
20*a^7*b^4*c^5*f*h*l^2 + 1105920*a^7*b^3*c^6*e*j*k^2 - 921600*a^7*b^2*c^7*f
*g^2*m - 829440*a^6*b^4*c^6*g^2*h*k - 552960*a^8*b^2*c^6*f*h*l^2 - 508032*a
^3*b^8*c^5*d^2*j*l - 331776*a^7*b^2*c^7*g^2*h*k + 290304*a^6*b^5*c^5*e*j*k^
2 - 103680*a^5*b^6*c^5*g^2*h*k + 80640*a^5*b^6*c^5*f*g^2*m - 69120*a^5*b^5*
c^6*e^2*h*m + 65664*a^2*b^10*c^4*d^2*j*l - 34560*a^6*b^6*c^4*f*h*l^2 + 6912
*a^5*b^7*c^4*e*j*k^2 + 3456*a^5*b^8*c^3*f*h*l^2 + 11930112*a^8*b^2*c^6*d*h*
m^2 + 8432640*a^7*b^2*c^7*d*h^2*m + 4450176*a^7*b^4*c^5*d*h*m^2 + 4337280*a
^6*b^4*c^6*d*h^2*m - 3870720*a^8*b^2*c^6*e*g*m^2 - 3640320*a^6*b^3*c^7*f^2*
h*k - 2885760*a^5*b^4*c^7*d^2*h*m - 2844288*a^4*b^6*c^6*d^2*h*m - 2626560*a
^7*b^3*c^6*f*h*k^2 + 2211840*a^7*b^2*c^7*f*h^2*k + 2056320*a^6*b^4*c^6*f*h^
2*k + 1935360*a^6*b^3*c^7*f^2*g*l - 1916928*a^7*b^2*c^7*d*j^2*k - 1687680*a
^6*b^6*c^4*d*h*m^2 - 1658880*a^7*b^2*c^7*e*h^2*l - 1143360*a^5*b^5*c^6*f^2*
h*k - 1097280*a^6*b^5*c^5*f*h*k^2 + 1019412*a^3*b^8*c^5*d^2*h*m - 1007424*a
^5*b^6*c^5*d*h^2*m - 912384*a^6*b^4*c^6*d*j^2*k - 829440*a^6*b^4*c^6*e*h^2*
l - 645120*a^7*b^4*c^5*e*g*m^2 - 552960*a^7*b^2*c^7*g*h^2*j - 552960*a^6*b^
4*c^6*g*h^2*j + 364608*a^5*b^6*c^5*f*h^2*k + 322560*a^5*b^5*c^6*f^2*g*l + 1
97460*a^5*b^8*c^3*d*h*m^2 - 145152*a^5*b^6*c^5*g*h^2*j - 143802*a^2*b^10*c^
4*d^2*h*m + 80640*a^6*b^6*c^4*e*g*m^2 - 56160*a^5*b^7*c^4*f*h*k^2 + 51948*a
^4*b^8*c^4*d*h^2*m - 40320*a^4*b^7*c^5*f^2*g*l + 34560*a^4*b^8*c^4*d*j^2*k
+ 27936*a^4*b^7*c^5*f^2*h*k - 20736*a^5*b^6*c^5*e*h^2*l - 13824*a^5*b^6*c^5
*d*j^2*k + 10800*a^3*b^10*c^3*d*h^2*m - 5760*a^3*b^10*c^3*d*j^2*k - 3780*a^
4*b^8*c^4*f*h^2*k + 3690*a^3*b^9*c^4*f^2*h*k - 3456*a^4*b^8*c^4*g*h^2*j + 2
970*a^4*b^9*c^3*f*h*k^2 - 2304*a^5*b^8*c^3*e*g*m^2 + 1152*a^3*b^9*c^4*f^2*g
*l - 540*a^3*b^10*c^3*f*h^2*k - 540*a^2*b^12*c^2*d*h^2*m - 90*a^4*b^10*c^2*
d*h*m^2 - 90*a^2*b^11*c^3*f^2*h*k + 54*a^3*b^11*c^2*f*h*k^2 + 15925248*a^6*
b^2*c^8*e^2*g*l - 7962624*a^7*b^3*c^6*e*g*l^2 - 7962624*a^6*b^3*c^7*e*g^2*l
+ 23385600*a^6*b^2*c^8*d*f^2*m + 6137856*a^6*b^3*c^7*d*g^2*m - 5677056*a^6
*b^2*c^8*e^2*f*m + 4147200*a^7*b^3*c^6*d*h*l^2 - 3317760*a^6*b^2*c^8*e^2*h*
k - 1354752*a^5*b^5*c^6*d*g^2*m + 1271808*a^6*b^3*c^7*f*g^2*k - 737280*a^7*
b^2*c^7*f*h*j^2 + 17418240*a^5*b^3*c^8*d^2*g*l - 568320*a^6*b^4*c^6*f*h*j^2
- 414720*a^6*b^5*c^5*d*h*l^2 + 414720*a^5*b^5*c^6*f*g^2*k - 414720*a^5*b^4
*c^7*e^2*h*k + 322560*a^5*b^4*c^7*e^2*f*m - 136704*a^5*b^6*c^5*f*h*j^2 + 12
0960*a^4*b^7*c^5*d*g^2*m - 31104*a^5*b^7*c^4*d*h*l^2 - 17280*a^4*b^7*c^5*f*
g^2*k + 10368*a^4*b^9*c^3*d*h*l^2 - 2304*a^4*b^8*c^4*f*h*j^2 + 384*a^3*b^10
*c^3*f*h*j^2 + 50042880*a^5*b^2*c^9*d^2*f*k - 13271040*a^5*b^3*c^8*d^2*h*k
- 13149696*a^7*b^3*c^6*d*f*m^2 + 10906560*a^4*b^5*c^7*d^2*f*m - 8709120*a^4
*b^5*c^7*d^2*g*l - 7418880*a^5*b^3*c^8*d^2*f*m + 7133184*a^7*b^2*c^7*d*h*k^
2 - 6428160*a^6*b^3*c^7*d*h^2*k + 5593536*a^4*b^5*c^7*d^2*h*k - 3870720*a^6

*b^2*c^8*e*f^2*l + 3369600*a^6*b^4*c^6*d*h*k^2 + 3148992*a^6*b^5*c^5*d*f*m^2 - 2985696*a^3*b^7*c^6*d^2*f*m + 1959552*a^3*b^7*c^6*d^2*g*l - 1658880*a^7*b^2*c^7*e*g*k^2 - 1505280*a^4*b^6*c^6*d*f^2*m - 1290240*a^6*b^2*c^8*f^2*g*j - 34836480*a^5*b^2*c^9*d^2*e*l + 1105920*a^6*b^3*c^7*e*h^2*j - 860160*a^5*b^4*c^7*f^2*g*j - 829440*a^6*b^4*c^6*e*g*k^2 - 692064*a^3*b^7*c^6*d^2*h*k - 689472*a^5*b^5*c^6*d*h^2*k - 645120*a^5*b^4*c^7*e*f^2*l - 388800*a^5*b^6*c^5*d*h*k^2 + 378954*a^2*b^9*c^5*d^2*f*m + 362880*a^5*b^4*c^7*d*f^2*m + 296964*a^3*b^8*c^5*d*f^2*m + 290304*a^5*b^5*c^6*e*h^2*j + 277344*a^4*b^7*c^5*d*h^2*k - 217728*a^2*b^9*c^5*d^2*g*l - 80640*a^4*b^6*c^6*f^2*g*j + 80640*a^4*b^6*c^6*e*f^2*l - 77070*a^4*b^9*c^3*d*f*m^2 - 30240*a^5*b^7*c^4*d*f*m^2 - 28350*a^3*b^9*c^4*d*h^2*k - 26406*a^2*b^9*c^5*d^2*h*k - 21060*a^4*b^8*c^4*d*h*k^2 - 20736*a^5*b^6*c^5*e*g*k^2 - 19278*a^2*b^10*c^4*d*f^2*m + 12672*a^3*b^8*c^5*f^2*g*j + 10044*a^3*b^10*c^3*d*h*k^2 + 8820*a^3*b^11*c^2*d*f*m^2 + 6912*a^4*b^7*c^5*e*h^2*j - 2304*a^3*b^8*c^5*e*f^2*l - 1620*a^2*b^11*c^3*d*h^2*k - 384*a^2*b^10*c^4*f^2*g*j + 162*a^2*b^12*c^2*d*h*k^2 - 5419008*a^5*b^3*c^8*d*e^2*m + 5308416*a^6*b^2*c^8*e*g^2*j - 5308416*a^5*b^3*c^8*e^2*g*j - 3870720*a^7*b^2*c^7*d*f*l^2 - 3538944*a^6*b^3*c^7*e*g*j^2 + 2654208*a^5*b^4*c^7*e*g^2*j - 2322432*a^6*b^2*c^8*d*g^2*k - 1990656*a^5*b^4*c^7*d*g^2*k - 1935360*a^6*b^4*c^6*d*f*l^2 + 1658880*a^6*b^3*c^7*d*h*j^2 + 1658880*a^5*b^3*c^8*e^2*f*k - 884736*a^5*b^5*c^6*e*g*j^2 + 725760*a^5*b^6*c^5*d*f*l^2 + 17418240*a^4*b^4*c^8*d^2*e*l + 518400*a^4*b^6*c^6*d*g^2*k + 483840*a^4*b^5*c^7*d*e^2*m + 262656*a^5*b^5*c^6*d*h*j^2 - 96768*a^4*b^8*c^4*d*f*l^2 - 69120*a^4*b^5*c^7*e^2*f*k - 55296*a^4*b^7*c^5*d*h*j^2 - 51840*a^3*b^8*c^5*d*g^2*k + 3456*a^3*b^10*c^3*d*f*l^2 + 1152*a^3*b^9*c^4*d*h*j^2 + 1152*a^2*b^11*c^3*d*h*j^2 - 15431040*a^4*b^4*c^8*d^2*f*k - 13248000*a^5*b^3*c^8*d*f^2*k - 11612160*a^5*b^2*c^9*d^2*g*j - 10063872*a^6*b^3*c^7*d*f*k^2 - 3919104*a^3*b^6*c^7*d^2*e*l + 2554560*a^4*b^5*c^7*d*f^2*k + 1720320*a^5*b^3*c^8*e*f^2*j + 1596672*a^3*b^6*c^7*d^2*g*j + 1518912*a^3*b^6*c^7*d^2*f*k - 1105920*a^5*b^4*c^7*f*g^2*h + 838080*a^5*b^5*c^6*d*f*k^2 - 552960*a^6*b^2*c^8*f*g^2*h - 508032*a^2*b^8*c^6*d^2*g*j + 435456*a^2*b^8*c^6*d^2*e*l + 161280*a^4*b^5*c^7*e*f^2*j + 116640*a^4*b^7*c^5*d*f*k^2 + 106812*a^2*b^8*c^6*d^2*f*k - 98208*a^3*b^7*c^6*d*f^2*k - 34560*a^4*b^6*c^6*f*g^2*h - 27270*a^3*b^9*c^4*d*f*k^2 - 26334*a^2*b^9*c^5*d*f^2*k - 25344*a^3*b^7*c^6*e*f^2*j + 3456*a^3*b^8*c^5*f*g^2*h + 768*a^2*b^9*c^5*e*f^2*j - 702*a^2*b^11*c^3*d*f*k^2 - 7962624*a^5*b^2*c^9*d*e^2*k - 2580480*a^6*b^2*c^8*d*f*j^2 + 2073600*a^4*b^4*c^8*d*e^2*k - 1658880*a^6*b^2*c^8*e*g*h^2 - 967680*a^5*b^4*c^7*d*f*j^2 - 829440*a^5*b^4*c^7*e*g*h^2 - 207360*a^3*b^6*c^7*d*e^2*k + 64512*a^4*b^6*c^6*d*f*j^2 + 39168*a^3*b^8*c^5*d*f*j^2 - 20736*a^4*b^6*c^6*e*g*h^2 - 9216*a^2*b^10*c^4*d*f*j^2 - 4423680*a^5*b^2*c^9*e^2*f*h + 4147200*a^5*b^3*c^8*d*g^2*h - 3193344*a^3*b^5*c^8*d^2*e*j + 1016064*a^2*b^7*c^7*d^2*e*j - 414720*a^4*b^5*c^7*d*g^2*h - 138240*a^4*b^4*c^8*e^2*f*h - 31104*a^3*b^7*c^6*d*g^2*h + 13824*a^3*b^6*c^7*e^2*f*h + 10368*a^2*b^9*c^5*d*g^2*h + 15630336*a^5*b^2*c^9*d*f^2*h - 14459904*a^4*b^3*c^9*d^2*f*h + 9630144*a^3*b^5*c^8*d^2*f*h - 8764416*a^5*b^3*c^8*d*f*h^2 - 3870720*a^5*b^2*c^9*e*f^2*g + 2867328*a^4*b^4*c^8*d*f^2*h - 2095200*a^2*b^7*c^7*d^2*f*h - 1414080*a^3*b^6*c^7*d*f^2*h - 34836480*a^4

*b^2*c^10*d^2*e*g - 645120*a^4*b^4*c^8*e*f^2*g + 306720*a^3*b^7*c^6*d*f*h^2
 + 197820*a^2*b^8*c^6*d*f^2*h + 146880*a^4*b^5*c^7*d*f*h^2 + 80640*a^3*b^6*
c^7*e*f^2*g - 55350*a^2*b^9*c^5*d*f*h^2 - 2304*a^2*b^8*c^6*e*f^2*g - 387072
0*a^5*b^2*c^9*d*f*g^2 - 1935360*a^4*b^4*c^8*d*f*g^2 - 1658880*a^4*b^3*c^9*d
*e^2*h + 725760*a^3*b^6*c^7*d*f*g^2 + 17418240*a^3*b^4*c^9*d^2*e*g - 124416
*a^3*b^5*c^8*d*e^2*h - 96768*a^2*b^8*c^6*d*f*g^2 + 41472*a^2*b^7*c^7*d*e^2*
h - 3919104*a^2*b^6*c^8*d^2*e*g - 7741440*a^4*b^2*c^10*d*e^2*f + 2903040*a^
3*b^4*c^9*d*e^2*f - 387072*a^2*b^6*c^8*d*e^2*f - 20160*a^8*b^7*c*l^2*m^2 -
1648128*a^10*b^3*c^3*k*m^3 - 898560*a^9*b^3*c^4*k^3*m - 354240*a^9*b^5*c^2*
k*m^3 - 354240*a^8*b^5*c^3*k^3*m - 21600*a^7*b^7*c^2*k^3*m - 13950*a^7*b^8*
c*k^2*m^2 + 430080*a^10*b*c^5*j^2*m^2 - 1984*a^6*b^9*c*j^2*m^2 - 884736*a^9
*b^3*c^4*j*l^3 - 589824*a^8*b^3*c^5*j^3*l - 442368*a^8*b^5*c^3*j*l^3 - 2949
12*a^7*b^5*c^4*j^3*l - 49152*a^6*b^7*c^3*j^3*l + 1359360*a^10*b^2*c^4*h*m^3
 + 1173120*a^9*b^4*c^3*h*m^3 + 743040*a^7*b^4*c^5*h^3*m + 622080*a^8*b^2*c^
6*h^3*m + 184320*a^9*b*c^6*j^2*k^2 + 107136*a^6*b^6*c^4*h^3*m - 32640*a^8*b
^6*c^2*h*m^3 + 540*a^5*b^8*c^3*h^3*m - 270*a^4*b^10*c^2*h^3*m - 180*a^5*b^1
0*c*h^2*m^2 - 2293760*a^9*b^3*c^4*f*m^3 - 2293760*a^6*b^3*c^7*f^3*m + 13271
04*a^8*b^4*c^4*g*l^3 + 1327104*a^6*b^4*c^6*g^3*l - 622080*a^8*b^3*c^5*h*k^3
 - 622080*a^7*b^3*c^6*h^3*k - 326592*a^7*b^5*c^4*h*k^3 - 326592*a^6*b^5*c^5
*h^3*k - 199360*a^8*b^5*c^3*f*m^3 - 199360*a^5*b^5*c^6*f^3*m + 61920*a^7*b^
7*c^2*f*m^3 + 61920*a^4*b^7*c^5*f^3*m - 38880*a^6*b^7*c^3*h*k^3 - 38880*a^5
*b^7*c^4*h^3*k - 3682*a^3*b^9*c^4*f^3*m - 810*a^5*b^9*c^2*h*k^3 - 810*a^4*b
^9*c^3*h^3*k - 70*a^3*b^12*c*f^2*m^2 + 70*a^2*b^11*c^3*f^3*m + 3870720*a^8*
b*c^7*e^2*m^2 + 184320*a^8*b*c^7*h^2*j^2 - 14152320*a^4*b^4*c^8*d^3*m + 106
44480*a^5*b^2*c^9*d^3*m + 5483520*a^9*b^2*c^5*d*m^3 + 4269888*a^3*b^6*c^7*d
^3*m - 2654208*a^8*b^3*c^5*e*l^3 + 1359360*a^6*b^2*c^8*f^3*k + 1330560*a^8*
b^4*c^4*d*m^3 + 1173120*a^5*b^4*c^7*f^3*k - 884736*a^6*b^3*c^7*g^3*j - 8265
60*a^7*b^6*c^3*d*m^3 + 743040*a^7*b^4*c^5*f*k^3 + 622080*a^8*b^2*c^6*f*k^3
 - 607068*a^2*b^8*c^6*d^3*m - 589824*a^7*b^3*c^6*g*j^3 - 442368*a^5*b^5*c^6*
g^3*j - 294912*a^6*b^5*c^5*g*j^3 + 145188*a^6*b^8*c^2*d*m^3 + 107136*a^6*b^
6*c^4*f*k^3 - 49152*a^5*b^7*c^4*g*j^3 - 32640*a^4*b^6*c^6*f^3*k - 5796*a^3*
b^8*c^5*f^3*k + 540*a^5*b^8*c^3*f*k^3 - 270*a^4*b^10*c^2*f*k^3 + 210*a^2*b^
10*c^4*f^3*k + 19077120*a^4*b^3*c^9*d^3*k + 1658880*a^7*b*c^8*e^2*k^2 + 430
080*a^7*b*c^8*f^2*j^2 + 3538944*a^5*b^2*c^9*e^3*j - 2488320*a^7*b^3*c^6*d*k
^3 - 2379456*a^3*b^5*c^8*d^3*k + 1179648*a^7*b^2*c^7*e*j^3 + 589824*a^6*b^4
*c^6*e*j^3 + 98304*a^5*b^6*c^5*e*j^3 - 95904*a^2*b^7*c^7*d^3*k - 57024*a^6*
b^5*c^5*d*k^3 + 49248*a^5*b^7*c^4*d*k^3 - 4050*a^4*b^9*c^3*d*k^3 - 810*a^3*
b^11*c^2*d*k^3 - 486*a*b^12*c^3*d^2*k^2 + 3870720*a^6*b*c^9*d^2*j^2 - 16481
28*a^5*b^3*c^8*f^3*h - 898560*a^6*b^3*c^7*f*h^3 - 354240*a^5*b^5*c^6*f*h^3
 - 354240*a^4*b^5*c^7*f^3*h + 43680*a^3*b^7*c^6*f^3*h - 21600*a^4*b^7*c^5*f*
h^3 - 9792*a*b^11*c^4*d^2*j^2 + 1350*a^3*b^9*c^4*f*h^3 - 1050*a^2*b^9*c^5*f
^3*h + 1658880*a^6*b*c^9*e^2*h^2 + 16547328*a^4*b^2*c^10*d^3*h - 12306816*a
^3*b^4*c^9*d^3*h + 37310976*a^3*b^3*c^10*d^3*f + 3037824*a^2*b^6*c^8*d^3*h
 - 2654208*a^5*b^3*c^8*e*g^3 + 1949184*a^6*b^2*c^8*d*h^3 + 1296000*a^5*b^4*c
^7*d*h^3 - 155520*a^4*b^6*c^6*d*h^3 - 40500*a*b^10*c^5*d^2*h^2 - 8100*a^3*b

^8*c^5*d*h^3 + 4050*a^2*b^10*c^4*d*h^3 + 3870720*a^5*b*c^10*e^2*f^2 + 34836
480*a^4*b*c^11*d^2*e^2 - 108864*a*b^9*c^6*d^2*g^2 - 8068032*a^2*b^5*c^9*d^3
*f - 5623296*a^4*b^3*c^9*d*f^3 + 1737792*a^3*b^5*c^8*d*f^3 - 260190*a*b^8*c
^7*d^2*f^2 - 211680*a^2*b^7*c^7*d*f^3 - 435456*a*b^7*c^8*d^2*e^2 - 245760*a
^10*c^6*j^2*k*m - 384*a^6*b^10*j*l*m^2 + 138240*a^10*c^6*h*k^2*m - 90*a^5*b
^11*h*k*m^2 + 384000*a^10*c^6*f*k*m^2 - 2211840*a^8*c^8*e^2*k*m - 409600*a^
9*c^7*f*j^2*m - 147456*a^9*c^7*h*j^2*k - 30*a^4*b^12*f*k*m^2 + 967680*a^9*c
^7*d*k^2*m + 384000*a^8*c^8*f^2*h*m - 90*a^3*b^13*d*k*m^2 + 20321280*a^7*c^
9*d^2*h*m - 883200*a^11*b*c^4*k*m^3 - 317952*a^10*b*c^5*k^3*m + 43680*a^8*b
^7*c*k*m^3 + 1350*a^6*b^9*c*k^3*m - 270*b^14*c^2*d^2*h*m + 6*a^3*b^13*f*h*m
^2 + 4838400*a^9*c^7*d*h*m^2 + 2903040*a^8*c^8*d*h^2*m - 1032192*a^8*c^8*d*
j^2*k + 138240*a^8*c^8*f*h^2*k - 3686400*a^7*c^9*e^2*f*m - 1327104*a^7*c^9*
e^2*h*k - 393216*a^9*b*c^6*j^3*l - 245760*a^8*c^8*f*h*j^2 - 810*b^13*c^3*d^
2*h*k + 630*b^13*c^3*d^2*f*m + 18*a^2*b^14*d*h*m^2 + 2688000*a^7*c^9*d*f^2*
m + 580608*a^8*c^8*d*h*k^2 - 5796*a^7*b^8*c*h*m^3 - 3456*b^12*c^4*d^2*g*j +
1890*b^12*c^4*d^2*f*k + 6773760*a^6*c^10*d^2*f*k - 1344000*a^10*b*c^5*f*m^
3 - 1344000*a^7*b*c^8*f^3*m - 207360*a^9*b*c^6*h*k^3 - 207360*a^8*b*c^7*h^3
*k - 3682*a^6*b^9*c*f*m^3 - 9289728*a^6*c^10*d*e^2*k - 1720320*a^7*c^9*d*f*
j^2 - 50803200*a^5*b*c^10*d^3*k + 6912*b^11*c^5*d^2*e*j - 10616832*a^6*b*c^
9*e^3*l - 2211840*a^6*c^10*e^2*f*h - 393216*a^8*b*c^7*g*j^3 + 43416*a*b^10*
c^5*d^3*m - 9576*a^5*b^10*c*d*m^3 - 9450*b^11*c^5*d^2*f*h - 504*a*b^14*c*d^
2*m^2 + 1612800*a^6*c^10*d*f^2*h - 1036800*a^8*b*c^7*d*k^3 + 45198*a*b^9*c^
6*d^3*k - 20736*b^10*c^6*d^2*e*g - 75188736*a^4*b*c^11*d^3*f - 883200*a^6*b
*c^9*f^3*h - 317952*a^7*b*c^8*f*h^3 - 15482880*a^5*c^11*d*e^2*f - 10616832*
a^5*b*c^10*e^3*g - 345060*a*b^8*c^7*d^3*h - 4262400*a^5*b*c^10*d*f^3 + 8527
68*a*b^7*c^8*d^3*f + 7350*a*b^9*c^6*d*f^3 + 967680*a^10*b^3*c^3*l^2*m^2 + 1
61280*a^9*b^5*c^2*l^2*m^2 + 1684224*a^10*b^2*c^4*k^2*m^2 + 1264320*a^9*b^4*
c^3*k^2*m^2 + 126720*a^8*b^6*c^2*k^2*m^2 + 501760*a^9*b^3*c^4*j^2*m^2 + 414
720*a^9*b^3*c^4*k^2*l^2 + 207360*a^8*b^5*c^3*k^2*l^2 + 170240*a^8*b^5*c^3*j
^2*m^2 + 9216*a^7*b^7*c^2*j^2*m^2 + 5184*a^7*b^7*c^2*k^2*l^2 + 884736*a^9*b
^2*c^5*j^2*l^2 + 884736*a^8*b^4*c^4*j^2*l^2 + 221184*a^7*b^6*c^3*j^2*l^2 +
1419840*a^8*b^4*c^4*h^2*m^2 + 1387008*a^9*b^2*c^5*h^2*m^2 + 276480*a^8*b^3*
c^5*j^2*k^2 + 140544*a^7*b^5*c^4*j^2*k^2 + 84960*a^7*b^6*c^3*h^2*m^2 + 2534
4*a^6*b^7*c^3*j^2*k^2 - 8010*a^6*b^8*c^2*h^2*m^2 + 576*a^5*b^9*c^2*j^2*k^2
+ 967680*a^8*b^3*c^5*g^2*m^2 + 414720*a^8*b^3*c^5*h^2*l^2 + 207360*a^7*b^5*
c^4*h^2*l^2 + 161280*a^7*b^5*c^4*g^2*m^2 - 20160*a^6*b^7*c^3*g^2*m^2 + 5184
*a^6*b^7*c^3*h^2*l^2 + 576*a^5*b^9*c^2*g^2*m^2 + 3808000*a^8*b^2*c^6*f^2*m^
2 + 1990656*a^7*b^4*c^5*g^2*l^2 + 1643712*a^7*b^4*c^5*f^2*m^2 + 803520*a^7*
b^4*c^5*h^2*k^2 + 725760*a^8*b^2*c^6*h^2*k^2 + 207360*a^6*b^6*c^4*h^2*k^2 -
125440*a^6*b^6*c^4*f^2*m^2 - 13790*a^5*b^8*c^3*f^2*m^2 + 10530*a^5*b^8*c^3
*h^2*k^2 + 1785*a^4*b^10*c^2*f^2*m^2 + 81*a^4*b^10*c^2*h^2*k^2 + 18427392*a
^7*b^2*c^7*d^2*m^2 + 967680*a^7*b^3*c^6*f^2*l^2 + 645120*a^7*b^3*c^6*e^2*m^
2 + 414720*a^7*b^3*c^6*g^2*k^2 + 276480*a^7*b^3*c^6*h^2*j^2 + 207360*a^6*b^
5*c^5*g^2*k^2 + 161280*a^6*b^5*c^5*f^2*l^2 + 140544*a^6*b^5*c^5*h^2*j^2 - 8
0640*a^6*b^5*c^5*e^2*m^2 + 25344*a^5*b^7*c^4*h^2*j^2 - 20160*a^5*b^7*c^4*f^

2*l^2 + 5184*a^5*b^7*c^4*g^2*k^2 + 2304*a^5*b^7*c^4*e^2*m^2 + 576*a^4*b^9*c
^3*h^2*j^2 + 576*a^4*b^9*c^3*f^2*l^2 + 7962624*a^7*b^2*c^7*e^2*l^2 - 414892
8*a^6*b^4*c^6*d^2*m^2 + 1419840*a^6*b^4*c^6*f^2*k^2 + 1387008*a^7*b^2*c^7*f
^2*k^2 - 1183392*a^5*b^6*c^5*d^2*m^2 + 884736*a^7*b^2*c^7*g^2*j^2 + 884736*
a^6*b^4*c^6*g^2*j^2 + 645750*a^4*b^8*c^4*d^2*m^2 + 221184*a^5*b^6*c^5*g^2*j
^2 - 115920*a^3*b^10*c^3*d^2*m^2 + 84960*a^5*b^6*c^5*f^2*k^2 + 10836*a^2*b^
12*c^2*d^2*m^2 - 8010*a^4*b^8*c^4*f^2*k^2 - 180*a^3*b^10*c^3*f^2*k^2 + 9*a^
2*b^12*c^2*f^2*k^2 + 8709120*a^6*b^3*c^7*d^2*l^2 - 4354560*a^5*b^5*c^6*d^2*
l^2 + 979776*a^4*b^7*c^5*d^2*l^2 + 829440*a^6*b^3*c^7*e^2*k^2 + 17480448*a^
6*b^2*c^8*d^2*k^2 + 501760*a^6*b^3*c^7*f^2*j^2 + 170240*a^5*b^5*c^6*f^2*j^2
 - 108864*a^3*b^9*c^4*d^2*l^2 + 20736*a^5*b^5*c^6*e^2*k^2 + 9216*a^4*b^7*c^
5*f^2*j^2 + 5184*a^2*b^11*c^3*d^2*l^2 - 1984*a^3*b^9*c^4*f^2*j^2 + 64*a^2*b
^11*c^3*f^2*j^2 + 3538944*a^6*b^2*c^8*e^2*j^2 - 3302208*a^5*b^4*c^7*d^2*k^2
 + 884736*a^5*b^4*c^7*e^2*j^2 + 414720*a^6*b^3*c^7*g^2*h^2 + 207360*a^5*b^5
*c^6*g^2*h^2 - 103680*a^4*b^6*c^6*d^2*k^2 + 101250*a^3*b^8*c^5*d^2*k^2 - 57
51*a^2*b^10*c^4*d^2*k^2 + 5184*a^4*b^7*c^5*g^2*h^2 + 1935360*a^5*b^3*c^8*d^
2*j^2 + 1684224*a^6*b^2*c^8*f^2*h^2 + 1264320*a^5*b^4*c^7*f^2*h^2 - 532224*
a^4*b^5*c^7*d^2*j^2 + 126720*a^4*b^6*c^6*f^2*h^2 - 96768*a^3*b^7*c^6*d^2*j^
2 + 62784*a^2*b^9*c^5*d^2*j^2 - 13950*a^3*b^8*c^5*f^2*h^2 + 225*a^2*b^10*c^
4*f^2*h^2 + 967680*a^5*b^3*c^8*f^2*g^2 + 829440*a^5*b^3*c^8*e^2*h^2 + 16128
0*a^4*b^5*c^7*f^2*g^2 + 20736*a^4*b^5*c^7*e^2*h^2 - 20160*a^3*b^7*c^6*f^2*g
^2 + 576*a^2*b^9*c^5*f^2*g^2 + 11487744*a^5*b^2*c^9*d^2*h^2 + 7962624*a^5*b
^2*c^9*e^2*g^2 + 35525376*a^4*b^2*c^10*d^2*f^2 - 1412640*a^3*b^6*c^7*d^2*h^
2 + 461376*a^4*b^4*c^8*d^2*h^2 + 375030*a^2*b^8*c^6*d^2*h^2 + 8709120*a^4*b
^3*c^9*d^2*g^2 - 4354560*a^3*b^5*c^8*d^2*g^2 + 979776*a^2*b^7*c^7*d^2*g^2 +
 645120*a^4*b^3*c^9*e^2*f^2 - 80640*a^3*b^5*c^8*e^2*f^2 + 2304*a^2*b^7*c^7*
e^2*f^2 - 15269184*a^3*b^4*c^9*d^2*f^2 + 2870784*a^2*b^6*c^8*d^2*f^2 - 1741
8240*a^3*b^3*c^10*d^2*e^2 + 3919104*a^2*b^5*c^9*d^2*e^2 + 54*b^15*c*d^2*k*m
 + 6*a*b^15*d*f*m^2 + 115200*a^11*c^5*k^2*m^2 + 576*a^7*b^9*l^2*m^2 + 225*a
^6*b^10*k^2*m^2 + 64*a^5*b^11*j^2*m^2 + 345600*a^10*c^6*h^2*m^2 + 9*a^4*b^1
2*h^2*m^2 + 320000*a^9*c^7*f^2*m^2 + 41472*a^9*c^7*h^2*k^2 + 16934400*a^8*c
^8*d^2*m^2 + 345600*a^8*c^8*f^2*k^2 + 81*b^14*c^2*d^2*k^2 + 3538944*a^7*c^9
*e^2*j^2 + 2032128*a^7*c^9*d^2*k^2 + 492800*a^11*b^2*c^3*m^4 + 351456*a^10*
b^4*c^2*m^4 + 576*b^13*c^3*d^2*j^2 + 331776*a^9*b^4*c^3*l^4 + 115200*a^7*c^
9*f^2*h^2 + 142560*a^8*b^4*c^4*k^4 + 103680*a^9*b^2*c^5*k^4 + 32400*a^7*b^6
*c^3*k^4 + 2025*b^12*c^4*d^2*h^2 + 2025*a^6*b^8*c^2*k^4 + 6096384*a^6*c^10*
d^2*h^2 + 131072*a^8*b^2*c^6*j^4 + 98304*a^7*b^4*c^5*j^4 + 32768*a^6*b^6*c^
4*j^4 + 5184*b^11*c^5*d^2*g^2 + 4096*a^5*b^8*c^3*j^4 + 11025*b^10*c^6*d^2*f
^2 + 5644800*a^5*c^11*d^2*f^2 + 142560*a^6*b^4*c^6*h^4 + 103680*a^7*b^2*c^7
*h^4 + 32400*a^5*b^6*c^5*h^4 + 20736*b^9*c^7*d^2*e^2 + 2025*a^4*b^8*c^4*h^4
 + 331776*a^5*b^4*c^7*g^4 + 492800*a^5*b^2*c^9*f^4 + 351456*a^4*b^4*c^8*f^4
 - 43120*a^3*b^6*c^7*f^4 + 1225*a^2*b^8*c^6*f^4 - 27433728*a^3*b^2*c^11*d^4
 + 6446304*a^2*b^4*c^10*d^4 - 1050*a^7*b^9*k*m^3 + 384000*a^11*c^5*h*m^3 +
138240*a^9*c^7*h^3*m + 210*a^6*b^10*h*m^3 + 47416320*a^6*c^10*d^3*m - 1134*
b^12*c^4*d^3*m + 70*a^5*b^11*f*m^3 + 2688000*a^10*c^6*d*m^3 + 384000*a^7*c^

9*f^3*k + 138240*a^9*c^7*f*k^3 - 3402*b^11*c^5*d^3*k + 210*a^4*b^12*d*m^3 + 7077888*a^6*c^10*e^3*j + 786432*a^8*c^8*e*j^3 - 43120*a^9*b^6*c*m^4 + 28449792*a^5*c^11*d^3*h + 17010*b^10*c^6*d^3*h + 580608*a^7*c^9*d*h^3 - 39690*b^9*c^7*d^3*f - 734832*a*b^6*c^9*d^4 + 9*b^16*d^2*m^2 + 160000*a^12*c^4*m^4 + 1225*a^8*b^8*m^4 + 20736*a^10*c^6*k^4 + 65536*a^9*c^7*j^4 + 20736*a^8*c^8*h^4 + 49787136*a^4*c^12*d^4 + 160000*a^6*c^10*f^4 + 5308416*a^5*c^11*e^4 + 35721*b^8*c^8*d^4 + a^2*b^14*f^2*m^2, z, k1)*(root(56371445760*a^11*b^8*c^9*z^4 - 503316480*a^8*b^14*c^6*z^4 + 47185920*a^7*b^16*c^5*z^4 - 2621440*a^6*b^18*c^4*z^4 + 65536*a^5*b^20*c^3*z^4 - 171798691840*a^14*b^2*c^12*z^4 + 193273528320*a^13*b^4*c^11*z^4 - 128849018880*a^12*b^6*c^10*z^4 - 16911433728*a^10*b^10*c^8*z^4 + 3523215360*a^9*b^12*c^7*z^4 + 68719476736*a^15*c^13*z^4 + 1536*a^5*b^16*c*k*m*z^2 + 1536*a*b^18*c^3*d*f*z^2 - 2571632640*a^9*b^5*c^8*d*m*z^2 + 2548039680*a^9*b^3*c^10*d*h*z^2 + 1509949440*a^10*b^3*c^9*e*l*z^2 + 1509949440*a^9*b^3*c^10*e*g*z^2 - 1401421824*a^8*b^5*c^9*d*h*z^2 - 1321205760*a^9*b^2*c^11*d*f*z^2 - 2793406464*a^11*b*c^10*d*m*z^2 + 890634240*a^8*b^7*c^7*d*m*z^2 - 754974720*a^10*b^4*c^8*g*l*z^2 - 754974720*a^9*b^5*c^8*e*l*z^2 + 719585280*a^8*b^6*c^8*d*k*z^2 - 707788800*a^9*b^4*c^9*d*k*z^2 - 754974720*a^8*b^5*c^9*e*g*z^2 + 603979776*a^11*b^2*c^9*g*l*z^2 - 581959680*a^10*b^4*c^8*f*m*z^2 + 732168192*a^7*b^6*c^9*d*f*z^2 + 534773760*a^11*b^3*c^8*h*m*z^2 - 456130560*a^11*b^4*c^7*k*m*z^2 - 603979776*a^10*b^2*c^10*e*j*z^2 + 534773760*a^10*b^3*c^9*f*k*z^2 + 384040960*a^9*b^6*c^7*f*m*z^2 + 377487360*a^9*b^6*c^7*g*l*z^2 - 456130560*a^9*b^4*c^9*f*h*z^2 + 301989888*a^11*b^3*c^8*j*l*z^2 - 415236096*a^10*b^2*c^10*d*k*z^2 + 254017536*a^10*b^6*c^6*k*m*z^2 - 330301440*a^10*b^4*c^8*h*k*z^2 + 390463488*a^7*b^7*c^8*d*h*z^2 + 188743680*a^12*b^2*c^8*k*m*z^2 + 301989888*a^10*b^3*c^9*g*j*z^2 - 297861120*a^7*b^8*c^7*d*k*z^2 - 366280704*a^6*b^8*c^8*d*f*z^2 + 188743680*a^11*b^2*c^9*h*k*z^2 - 330301440*a^8*b^4*c^10*d*f*z^2 + 254017536*a^8*b^6*c^8*f*h*z^2 - 1887436800*a^10*b*c^11*d*h*z^2 + 188743680*a^8*b^7*c^7*e*l*z^2 + 153354240*a^9*b^6*c^7*h*k*z^2 - 185303040*a^7*b^9*c^6*d*m*z^2 - 117964800*a^10*b^5*c^7*h*m*z^2 - 61931520*a^9*b^8*c^5*k*m*z^2 + 121634816*a^11*b^2*c^9*f*m*z^2 - 115671040*a^8*b^8*c^6*f*m*z^2 - 62914560*a^9*b^7*c^6*j*l*z^2 + 188743680*a^10*b^2*c^10*f*h*z^2 - 94371840*a^8*b^8*c^6*g*l*z^2 + 6144000*a^8*b^10*c^4*k*m*z^2 - 117964800*a^9*b^5*c^8*f*k*z^2 + 61440*a^7*b^12*c^3*k*m*z^2 - 46080*a^6*b^14*c^2*k*m*z^2 + 23592960*a^8*b^9*c^5*j*l*z^2 + 188743680*a^7*b^7*c^8*e*g*z^2 - 37355520*a^9*b^7*c^6*h*m*z^2 + 125829120*a^8*b^6*c^8*e*j*z^2 + 23101440*a^8*b^9*c^5*h*m*z^2 - 3538944*a^7*b^11*c^4*j*l*z^2 + 196608*a^6*b^13*c^3*j*l*z^2 - 4349952*a^7*b^11*c^4*h*m*z^2 + 337920*a^6*b^13*c^3*h*m*z^2 - 7680*a^5*b^15*c^2*h*m*z^2 - 62914560*a^8*b^7*c^7*g*j*z^2 - 26542080*a^8*b^8*c^6*h*k*z^2 + 17940480*a^7*b^10*c^5*f*m*z^2 + 11796480*a^7*b^10*c^5*g*l*z^2 - 37355520*a^8*b^7*c^7*f*k*z^2 - 1347584*a^6*b^12*c^4*f*m*z^2 + 68272128*a^6*b^10*c^6*d*k*z^2 - 589824*a^6*b^12*c^4*g*l*z^2 + 552960*a^6*b^12*c^4*h*k*z^2 - 147456*a^7*b^10*c^5*h*k*z^2 - 46080*a^5*b^14*c^3*h*k*z^2 + 35840*a^5*b^14*c^3*f*m*z^2 + 23592960*a^7*b^9*c^6*g*j*z^2 - 23592960*a^7*b^9*c^6*e*l*z^2 + 23371776*a^6*b^11*c^5*d*m*z^2 + 23101440*a^7*b^9*c^6*f*k*z^2 - 47185920*a^7*b^8*c^7*e*j*z^2 - 61931520*a^7*b^8*c^7*f*h*z^2 - 4349952

*a^6*b^11*c^5*f*k*z^2 - 3538944*a^6*b^11*c^5*g*j*z^2 - 1677312*a^5*b^13*c^4
*d*m*z^2 + 1179648*a^6*b^11*c^5*e*l*z^2 + 337920*a^5*b^13*c^4*f*k*z^2 + 196
608*a^5*b^13*c^4*g*j*z^2 + 53760*a^4*b^15*c^3*d*m*z^2 - 7680*a^4*b^15*c^3*f
*k*z^2 + 96583680*a^5*b^10*c^7*d*f*z^2 - 9179136*a^5*b^12*c^5*d*k*z^2 + 707
7888*a^6*b^10*c^6*e*j*z^2 - 51609600*a^6*b^9*c^7*d*h*z^2 + 691200*a^4*b^14*
c^4*d*k*z^2 - 393216*a^5*b^12*c^5*e*j*z^2 - 23040*a^3*b^16*c^3*d*k*z^2 + 61
44000*a^6*b^10*c^6*f*h*z^2 + 61440*a^5*b^12*c^5*f*h*z^2 - 46080*a^4*b^14*c^
4*f*h*z^2 + 1536*a^3*b^16*c^3*f*h*z^2 - 23592960*a^6*b^9*c^7*e*g*z^2 + 1179
648*a^5*b^11*c^6*e*g*z^2 + 829440*a^4*b^13*c^5*d*h*z^2 + 368640*a^5*b^11*c^
6*d*h*z^2 - 105984*a^3*b^15*c^4*d*h*z^2 + 4608*a^2*b^17*c^3*d*h*z^2 - 15175
680*a^4*b^12*c^6*d*f*z^2 + 1428480*a^3*b^14*c^5*d*f*z^2 - 73728*a^2*b^16*c^
4*d*f*z^2 + 4108320768*a^10*b^3*c^9*d*m*z^2 - 1207959552*a^11*b*c^10*e*l*z^
2 - 1207959552*a^10*b*c^11*e*g*z^2 - 578813952*a^12*b*c^9*h*m*z^2 - 5788139
52*a^11*b*c^10*f*k*z^2 - 402653184*a^12*b*c^9*j*l*z^2 - 402653184*a^11*b*c^
10*g*j*z^2 - 440401920*a^10*b*c^11*f^2*z^2 - 188743680*a^12*b*c^9*k^2*z^2 -
 188743680*a^11*b*c^10*h^2*z^2 + 1761607680*a^10*c^12*d*f*z^2 - 14080*a^6*b
^15*c*m^2*z^2 - 94464*a*b^17*c^4*d^2*z^2 + 6936330240*a^8*b^3*c^11*d^2*z^2
+ 2464874496*a^6*b^7*c^9*d^2*z^2 - 3963617280*a^9*b*c^12*d^2*z^2 + 10569646
08*a^11*c^11*d*k*z^2 + 805306368*a^11*c^11*e*j*z^2 + 419430400*a^12*c^10*f*
m*z^2 + 251658240*a^13*c^9*k*m*z^2 - 1509949440*a^9*b^2*c^11*e^2*z^2 + 2516
58240*a^11*c^11*f*h*z^2 + 150994944*a^12*c^10*h*k*z^2 - 5400428544*a^7*b^5*
c^10*d^2*z^2 + 754974720*a^8*b^4*c^10*e^2*z^2 - 730054656*a^5*b^9*c^8*d^2*z
^2 + 477102080*a^12*b^3*c^7*m^2*z^2 - 377487360*a^11*b^4*c^7*l^2*z^2 + 4771
02080*a^9*b^3*c^10*f^2*z^2 + 301989888*a^12*b^2*c^8*l^2*z^2 - 377487360*a^9
*b^4*c^9*g^2*z^2 + 301989888*a^10*b^2*c^10*g^2*z^2 - 174325760*a^11*b^5*c^6
*m^2*z^2 + 188743680*a^10*b^6*c^6*l^2*z^2 + 141557760*a^11*b^3*c^8*k^2*z^2
+ 188743680*a^8*b^6*c^8*g^2*z^2 + 141557760*a^10*b^3*c^9*h^2*z^2 - 17432576
0*a^8*b^5*c^9*f^2*z^2 - 188743680*a^7*b^6*c^9*e^2*z^2 - 47185920*a^9*b^8*c^
5*l^2*z^2 + 11206656*a^10*b^7*c^5*m^2*z^2 + 8929280*a^9*b^9*c^4*m^2*z^2 - 2
600960*a^8*b^11*c^3*m^2*z^2 + 291840*a^7*b^13*c^2*m^2*z^2 - 50331648*a^10*b
^4*c^8*j^2*z^2 + 146165760*a^4*b^11*c^7*d^2*z^2 - 26542080*a^9*b^7*c^6*k^2*
z^2 + 5898240*a^8*b^10*c^4*l^2*z^2 - 294912*a^7*b^12*c^3*l^2*z^2 - 33554432
*a^11*b^2*c^9*j^2*z^2 + 9584640*a^8*b^9*c^5*k^2*z^2 + 20971520*a^9*b^6*c^7*
j^2*z^2 - 2359296*a^10*b^5*c^7*k^2*z^2 - 1290240*a^7*b^11*c^4*k^2*z^2 + 460
80*a^6*b^13*c^3*k^2*z^2 + 2304*a^5*b^15*c^2*k^2*z^2 - 2752512*a^7*b^10*c^5*
j^2*z^2 + 2621440*a^8*b^8*c^6*j^2*z^2 + 524288*a^6*b^12*c^4*j^2*z^2 - 32768
*a^5*b^14*c^3*j^2*z^2 - 47185920*a^7*b^8*c^7*g^2*z^2 - 26542080*a^8*b^7*c^7
*h^2*z^2 + 9584640*a^7*b^9*c^6*h^2*z^2 - 2359296*a^9*b^5*c^8*h^2*z^2 - 1290
240*a^6*b^11*c^5*h^2*z^2 + 46080*a^5*b^13*c^4*h^2*z^2 + 2304*a^4*b^15*c^3*h
^2*z^2 + 5898240*a^6*b^10*c^6*g^2*z^2 - 294912*a^5*b^12*c^5*g^2*z^2 + 11206
656*a^7*b^7*c^8*f^2*z^2 + 8929280*a^6*b^9*c^7*f^2*z^2 + 23592960*a^6*b^8*c^
8*e^2*z^2 - 2600960*a^5*b^11*c^6*f^2*z^2 + 291840*a^4*b^13*c^5*f^2*z^2 - 14
080*a^3*b^15*c^4*f^2*z^2 + 256*a^2*b^17*c^3*f^2*z^2 - 19860480*a^3*b^13*c^6
*d^2*z^2 - 1179648*a^5*b^10*c^7*e^2*z^2 + 1771776*a^2*b^15*c^5*d^2*z^2 - 44
0401920*a^13*b*c^8*m^2*z^2 + 1207959552*a^10*c^12*e^2*z^2 + 134217728*a^12*

c^10*j^2*z^2 + 256*a^5*b^17*m^2*z^2 + 2304*b^19*c^3*d^2*z^2 - 23592960*a^10 *b*c^8*f*k*l*z + 99090432*a^9*b*c^9*d*h*l*z + 9437184*a^10*b*c^8*e*k*m*z + 23592960*a^10*b*c^8*g*h*m*z + 141557760*a^8*b*c^10*d*e*k*z + 47185920*a^9*b *c^9*d*j*k*z - 23592960*a^9*b*c^9*f*g*k*z + 169869312*a^7*b*c^11*d*e*f*z + 99090432*a^8*b*c^10*d*g*h*z - 3145728*a^9*b*c^9*f*h*j*z + 56623104*a^8*b*c^ 10*d*f*j*z + 1536*a*b^15*c^3*d*f*j*z - 9437184*a^8*b*c^10*e*f*h*z - 4608*a* b^14*c^4*d*f*g*z + 9216*a*b^13*c^5*d*e*f*z + 412876800*a^8*b^2*c^9*d*e*m*z - 206438400*a^9*b^3*c^7*d*l*m*z + 5898240*a^10*b^4*c^5*k*l*m*z - 206438400* a^8*b^3*c^8*d*g*m*z - 4718592*a^11*b^2*c^6*k*l*m*z - 2949120*a^9*b^6*c^4*k* l*m*z + 737280*a^8*b^8*c^3*k*l*m*z - 92160*a^7*b^10*c^2*k*l*m*z + 103219200 *a^8*b^5*c^6*d*l*m*z - 29491200*a^10*b^3*c^6*h*l*m*z - 206438400*a^7*b^4*c^ 8*d*e*m*z - 2359296*a^10*b^3*c^6*j*k*m*z + 491520*a^8*b^7*c^4*j*k*m*z - 184 320*a^7*b^9*c^3*j*k*m*z + 27648*a^6*b^11*c^2*j*k*m*z + 14745600*a^9*b^5*c^5 *h*l*m*z - 3686400*a^8*b^7*c^4*h*l*m*z + 460800*a^7*b^9*c^3*h*l*m*z - 23040 *a^6*b^11*c^2*h*l*m*z + 88473600*a^8*b^4*c^7*d*k*l*z + 82575360*a^9*b^2*c^8 *d*j*m*z + 11796480*a^10*b^2*c^7*h*j*m*z + 5898240*a^9*b^4*c^6*g*k*m*z - 47 18592*a^10*b^2*c^7*g*k*m*z - 70778880*a^9*b^2*c^8*d*k*l*z - 2949120*a^8*b^6 *c^5*g*k*m*z - 2457600*a^8*b^6*c^5*h*j*m*z + 921600*a^7*b^8*c^4*h*j*m*z + 7 37280*a^7*b^8*c^4*g*k*m*z - 138240*a^6*b^10*c^3*h*j*m*z - 92160*a^6*b^10*c^ 3*g*k*m*z + 7680*a^5*b^12*c^2*h*j*m*z + 4608*a^5*b^12*c^2*g*k*m*z + 2949120 0*a^9*b^3*c^7*f*k*l*z - 176947200*a^7*b^3*c^9*d*e*k*z - 109707264*a^8*b^3*c ^8*d*h*l*z - 25804800*a^7*b^7*c^5*d*l*m*z + 103219200*a^7*b^5*c^7*d*g*m*z + 219414528*a^7*b^2*c^10*d*e*h*z - 14745600*a^8*b^5*c^6*f*k*l*z - 29491200*a ^9*b^3*c^7*g*h*m*z - 11796480*a^9*b^3*c^7*e*k*m*z - 44236800*a^7*b^6*c^6*d* k*l*z + 58982400*a^9*b^2*c^8*e*h*m*z + 5898240*a^8*b^5*c^6*e*k*m*z + 368640 0*a^7*b^7*c^5*f*k*l*z + 3225600*a^6*b^9*c^4*d*l*m*z - 1474560*a^7*b^7*c^5*e *k*m*z - 460800*a^6*b^9*c^4*f*k*l*z + 184320*a^6*b^9*c^4*e*k*m*z - 161280*a ^5*b^11*c^3*d*l*m*z + 23040*a^5*b^11*c^3*f*k*l*z - 9216*a^5*b^11*c^3*e*k*m* z + 14745600*a^8*b^5*c^6*g*h*m*z + 110886912*a^7*b^4*c^8*d*f*l*z - 3686400* a^7*b^7*c^5*g*h*m*z - 221773824*a^6*b^3*c^10*d*e*f*z + 460800*a^6*b^9*c^4*g *h*m*z - 17203200*a^7*b^6*c^6*d*j*m*z - 23040*a^5*b^11*c^3*g*h*m*z - 294912 00*a^8*b^4*c^7*e*h*m*z - 11796480*a^9*b^2*c^8*f*j*k*z + 11059200*a^6*b^8*c^ 5*d*k*l*z + 6451200*a^6*b^8*c^5*d*j*m*z + 88473600*a^7*b^4*c^8*d*g*k*z + 24 57600*a^7*b^6*c^6*f*j*k*z - 35389440*a^8*b^3*c^8*d*j*k*z - 1382400*a^5*b^10 *c^4*d*k*l*z - 84934656*a^8*b^2*c^9*d*f*l*z - 967680*a^5*b^10*c^4*d*j*m*z - 921600*a^6*b^8*c^5*f*j*k*z + 138240*a^5*b^10*c^4*f*j*k*z + 69120*a^4*b^12* c^3*d*k*l*z + 53760*a^4*b^12*c^3*d*j*m*z - 7680*a^4*b^12*c^3*f*j*k*z + 4423 6800*a^7*b^5*c^7*d*h*l*z + 7372800*a^7*b^6*c^6*e*h*m*z - 5898240*a^8*b^4*c^ 7*f*h*l*z + 4718592*a^9*b^2*c^8*f*h*l*z - 70778880*a^8*b^2*c^9*d*g*k*z + 29 49120*a^7*b^6*c^6*f*h*l*z - 921600*a^6*b^8*c^5*e*h*m*z - 737280*a^6*b^8*c^5 *f*h*l*z + 92160*a^5*b^10*c^4*f*h*l*z + 46080*a^5*b^10*c^4*e*h*m*z - 4608*a ^4*b^12*c^3*f*h*l*z + 29491200*a^8*b^3*c^8*f*g*k*z - 109707264*a^7*b^3*c^9* d*g*h*z - 25804800*a^6*b^7*c^6*d*g*m*z - 58982400*a^8*b^2*c^9*e*f*k*z - 589 82400*a^6*b^6*c^7*d*f*l*z + 7372800*a^6*b^7*c^6*d*j*k*z + 88473600*a^6*b^5* c^8*d*e*k*z - 2764800*a^5*b^9*c^5*d*j*k*z + 51609600*a^6*b^6*c^7*d*e*m*z +

414720*a^4*b^11*c^4*d*j*k*z - 23040*a^3*b^13*c^3*d*j*k*z - 14745600*a^7*b^5
*c^7*f*g*k*z - 44236800*a^6*b^6*c^7*d*g*k*z - 6635520*a^6*b^7*c^6*d*h*l*z +
40108032*a^8*b^2*c^9*d*h*j*z + 3686400*a^6*b^7*c^6*f*g*k*z + 3225600*a^5*b
^9*c^5*d*g*m*z + 2359296*a^8*b^3*c^8*f*h*j*z - 491520*a^6*b^7*c^6*f*h*j*z -
460800*a^5*b^9*c^5*f*g*k*z - 276480*a^5*b^9*c^5*d*h*l*z + 184320*a^5*b^9*c
^5*f*h*j*z + 179712*a^4*b^11*c^4*d*h*l*z - 161280*a^4*b^11*c^4*d*g*m*z - 27
648*a^4*b^11*c^4*f*h*j*z + 23040*a^4*b^11*c^4*f*g*k*z - 13824*a^3*b^13*c^3*
d*h*l*z + 1536*a^3*b^13*c^3*f*h*j*z + 29491200*a^7*b^4*c^8*e*f*k*z + 110886
912*a^6*b^4*c^9*d*f*g*z + 16220160*a^5*b^8*c^6*d*f*l*z - 45613056*a^7*b^3*c
^9*d*f*j*z + 11059200*a^5*b^8*c^6*d*g*k*z - 10321920*a^6*b^6*c^7*d*h*j*z -
7372800*a^6*b^6*c^7*e*f*k*z + 7077888*a^7*b^4*c^8*d*h*j*z - 6451200*a^5*b^8
*c^6*d*e*m*z - 88473600*a^6*b^4*c^9*d*e*h*z + 2396160*a^5*b^8*c^6*d*h*j*z -
2396160*a^4*b^10*c^5*d*f*l*z - 1382400*a^4*b^10*c^5*d*g*k*z - 84934656*a^7
*b^2*c^10*d*f*g*z + 921600*a^5*b^8*c^6*e*f*k*z + 117964800*a^5*b^5*c^9*d*e*
f*z + 322560*a^4*b^10*c^5*d*e*m*z + 175104*a^3*b^12*c^4*d*f*l*z + 69120*a^3
*b^12*c^4*d*g*k*z - 50688*a^3*b^12*c^4*d*h*j*z - 46080*a^4*b^10*c^5*e*f*k*z
- 27648*a^4*b^10*c^5*d*h*j*z + 4608*a^2*b^14*c^3*d*h*j*z - 4608*a^2*b^14*c
^3*d*f*l*z + 44236800*a^6*b^5*c^8*d*g*h*z - 5898240*a^7*b^4*c^8*f*g*h*z - 2
2118400*a^5*b^7*c^7*d*e*k*z + 4718592*a^8*b^2*c^9*f*g*h*z + 2949120*a^6*b^6
*c^7*f*g*h*z - 737280*a^5*b^8*c^6*f*g*h*z + 92160*a^4*b^10*c^5*f*g*h*z - 46
08*a^3*b^12*c^4*f*g*h*z + 8847360*a^5*b^7*c^7*d*f*j*z - 58982400*a^5*b^6*c^
8*d*f*g*z - 3809280*a^4*b^9*c^6*d*f*j*z + 2764800*a^4*b^9*c^6*d*e*k*z + 235
9296*a^6*b^5*c^8*d*f*j*z + 681984*a^3*b^11*c^5*d*f*j*z - 138240*a^3*b^11*c^
5*d*e*k*z - 55296*a^2*b^13*c^4*d*f*j*z + 11796480*a^7*b^3*c^9*e*f*h*z - 663
5520*a^5*b^7*c^7*d*g*h*z - 5898240*a^6*b^5*c^8*e*f*h*z + 1474560*a^5*b^7*c^
7*e*f*h*z - 276480*a^4*b^9*c^6*d*g*h*z - 184320*a^4*b^9*c^6*e*f*h*z + 17971
2*a^3*b^11*c^5*d*g*h*z - 13824*a^2*b^13*c^4*d*g*h*z + 9216*a^3*b^11*c^5*e*f
*h*z + 16220160*a^4*b^8*c^7*d*f*g*z + 13271040*a^5*b^6*c^8*d*e*h*z - 239616
0*a^3*b^10*c^6*d*f*g*z + 552960*a^4*b^8*c^7*d*e*h*z - 359424*a^3*b^10*c^6*d
*e*h*z + 175104*a^2*b^12*c^5*d*f*g*z + 27648*a^2*b^12*c^5*d*e*h*z - 3244032
0*a^4*b^7*c^8*d*e*f*z + 4792320*a^3*b^9*c^7*d*e*f*z - 350208*a^2*b^11*c^6*d
*e*f*z + 165150720*a^10*b*c^8*d*l*m*z + 4608*a^6*b^12*c*k*l*m*z + 23592960*
a^11*b*c^7*h*l*m*z + 3145728*a^11*b*c^7*j*k*m*z - 1536*a^5*b^13*c*j*k*m*z +
165150720*a^9*b*c^9*d*g*m*z + 346816512*a^7*b*c^11*d^2*g*z + 19660800*a^12
*b*c^6*l*m^2*z - 34560*a^7*b^11*c*l*m^2*z - 7077888*a^11*b*c^7*k^2*l*z + 11
008*a^6*b^12*c*j*m^2*z + 19660800*a^11*b*c^7*g*m^2*z + 7077888*a^10*b*c^8*h
^2*l*z + 768*a^5*b^13*c*g*m^2*z - 19660800*a^9*b*c^9*f^2*l*z - 7077888*a^10
*b*c^8*g*k^2*z - 6912*a*b^15*c^3*d^2*l*z + 7077888*a^9*b*c^9*g*h^2*z - 1966
0800*a^8*b*c^10*f^2*g*z - 66816*a*b^14*c^4*d^2*j*z + 214272*a*b^13*c^5*d^2*
g*z - 428544*a*b^12*c^6*d^2*e*z - 330301440*a^9*c^10*d*e*m*z - 110100480*a^
10*c^9*d*j*m*z - 15728640*a^11*c^8*h*j*m*z - 47185920*a^10*c^9*e*h*m*z - 19
8180864*a^8*c^11*d*e*h*z + 15728640*a^10*c^9*f*j*k*z - 66060288*a^9*c^10*d*
h*j*z + 47185920*a^9*c^10*e*f*k*z + 1022754816*a^6*b^2*c^11*d^2*e*z - 64231
8336*a^5*b^4*c^10*d^2*e*z - 511377408*a^7*b^3*c^9*d^2*l*z - 511377408*a^6*b
^3*c^10*d^2*g*z + 321159168*a^6*b^5*c^8*d^2*l*z + 321159168*a^5*b^5*c^9*d^2

*g*z + 225312768*a^7*b^2*c^10*d^2*j*z - 25362432*a^11*b^3*c^5*l*m^2*z + 132
71040*a^10*b^5*c^4*l*m^2*z - 3563520*a^9*b^7*c^3*l*m^2*z + 506880*a^8*b^9*c
^2*l*m^2*z + 10354688*a^11*b^2*c^6*j*m^2*z + 8847360*a^10*b^3*c^6*k^2*l*z -
 4423680*a^9*b^5*c^5*k^2*l*z - 2048000*a^9*b^6*c^4*j*m^2*z + 1105920*a^8*b^
7*c^4*k^2*l*z + 849920*a^8*b^8*c^3*j*m^2*z - 393216*a^10*b^4*c^5*j*m^2*z -
145920*a^7*b^10*c^2*j*m^2*z - 138240*a^7*b^9*c^3*k^2*l*z + 6912*a^6*b^11*c^
2*k^2*l*z - 111697920*a^5*b^7*c^7*d^2*l*z + 223395840*a^4*b^6*c^9*d^2*e*z -
 25362432*a^10*b^3*c^6*g*m^2*z - 3538944*a^10*b^2*c^7*j*k^2*z + 737280*a^8*
b^6*c^5*j*k^2*z + 50724864*a^10*b^2*c^7*e*m^2*z - 276480*a^7*b^8*c^4*j*k^2*
z + 41472*a^6*b^10*c^3*j*k^2*z - 2304*a^5*b^12*c^2*j*k^2*z + 13271040*a^9*b
^5*c^5*g*m^2*z - 8847360*a^9*b^3*c^7*h^2*l*z + 4423680*a^8*b^5*c^6*h^2*l*z
- 3563520*a^8*b^7*c^4*g*m^2*z - 1105920*a^7*b^7*c^5*h^2*l*z + 506880*a^7*b^
9*c^3*g*m^2*z + 138240*a^6*b^9*c^4*h^2*l*z - 34560*a^6*b^11*c^2*g*m^2*z - 6
912*a^5*b^11*c^3*h^2*l*z - 26542080*a^9*b^4*c^6*e*m^2*z + 25362432*a^8*b^3*
c^8*f^2*l*z - 13271040*a^7*b^5*c^7*f^2*l*z + 8847360*a^9*b^3*c^7*g*k^2*z +
7127040*a^8*b^6*c^5*e*m^2*z - 4423680*a^8*b^5*c^6*g*k^2*z + 3563520*a^6*b^7
*c^6*f^2*l*z + 3538944*a^9*b^2*c^8*h^2*j*z + 1105920*a^7*b^7*c^5*g*k^2*z -
1013760*a^7*b^8*c^4*e*m^2*z - 737280*a^7*b^6*c^6*h^2*j*z - 506880*a^5*b^9*c
^5*f^2*l*z + 276480*a^6*b^8*c^5*h^2*j*z - 138240*a^6*b^9*c^4*g*k^2*z + 6912
0*a^6*b^10*c^3*e*m^2*z - 41472*a^5*b^10*c^4*h^2*j*z + 34560*a^4*b^11*c^4*f^
2*l*z + 6912*a^5*b^11*c^3*g*k^2*z + 2304*a^4*b^12*c^3*h^2*j*z - 1536*a^5*b^
12*c^2*e*m^2*z - 768*a^3*b^13*c^3*f^2*l*z - 111697920*a^4*b^7*c^8*d^2*g*z +
 23362560*a^4*b^9*c^6*d^2*l*z - 17694720*a^9*b^2*c^8*e*k^2*z - 10354688*a^8
*b^2*c^9*f^2*j*z - 43646976*a^6*b^4*c^9*d^2*j*z + 8847360*a^8*b^4*c^7*e*k^2
*z - 2965248*a^3*b^11*c^5*d^2*l*z - 2211840*a^7*b^6*c^6*e*k^2*z + 2048000*a
^6*b^6*c^7*f^2*j*z - 849920*a^5*b^8*c^6*f^2*j*z + 393216*a^7*b^4*c^8*f^2*j*
z + 276480*a^6*b^8*c^5*e*k^2*z + 214272*a^2*b^13*c^4*d^2*l*z + 145920*a^4*b
^10*c^5*f^2*j*z - 13824*a^5*b^10*c^4*e*k^2*z - 11008*a^3*b^12*c^4*f^2*j*z +
 256*a^2*b^14*c^3*f^2*j*z - 32587776*a^5*b^6*c^8*d^2*j*z - 8847360*a^8*b^3*
c^8*g*h^2*z + 21657600*a^4*b^8*c^7*d^2*j*z + 4423680*a^7*b^5*c^7*g*h^2*z -
1105920*a^6*b^7*c^6*g*h^2*z + 138240*a^5*b^9*c^5*g*h^2*z - 6912*a^4*b^11*c^
4*g*h^2*z + 25362432*a^7*b^3*c^9*f^2*g*z - 5810688*a^3*b^10*c^6*d^2*j*z + 1
7694720*a^8*b^2*c^9*e*h^2*z + 845568*a^2*b^12*c^5*d^2*j*z - 50724864*a^7*b^
2*c^10*e*f^2*z - 13271040*a^6*b^5*c^8*f^2*g*z - 8847360*a^7*b^4*c^8*e*h^2*z
 + 3563520*a^5*b^7*c^7*f^2*g*z + 2211840*a^6*b^6*c^7*e*h^2*z - 506880*a^4*b
^9*c^6*f^2*g*z - 276480*a^5*b^8*c^6*e*h^2*z + 34560*a^3*b^11*c^5*f^2*g*z +
13824*a^4*b^10*c^5*e*h^2*z - 768*a^2*b^13*c^4*f^2*g*z + 26542080*a^6*b^4*c^
9*e*f^2*z + 23362560*a^3*b^9*c^7*d^2*g*z - 46725120*a^3*b^8*c^8*d^2*e*z - 7
127040*a^5*b^6*c^8*e*f^2*z - 2965248*a^2*b^11*c^6*d^2*g*z + 1013760*a^4*b^8
*c^7*e*f^2*z - 69120*a^3*b^10*c^6*e*f^2*z + 1536*a^2*b^12*c^5*e*f^2*z + 593
0496*a^2*b^10*c^7*d^2*e*z + 346816512*a^8*b*c^10*d^2*l*z - 693633024*a^7*c^
12*d^2*e*z - 231211008*a^8*c^11*d^2*j*z + 768*a^6*b^13*l*m^2*z - 13107200*a
^12*c^7*j*m^2*z - 256*a^5*b^14*j*m^2*z + 4718592*a^11*c^8*j*k^2*z - 3932160
0*a^11*c^8*e*m^2*z - 4718592*a^10*c^9*h^2*j*z + 14155776*a^10*c^9*e*k^2*z +
 13107200*a^9*c^10*f^2*j*z + 2304*b^16*c^3*d^2*j*z - 14155776*a^9*c^10*e*h^

2*z + 39321600*a^8*c^11*e*f^2*z - 6912*b^15*c^4*d^2*g*z + 13824*b^14*c^5*d^
2*e*z + 737280*a^10*b*c^5*j*k*l*m - 2304*a^6*b^9*c*j*k*l*m + 2211840*a^9*b*
c^6*e*k*l*m + 1228800*a^9*b*c^6*f*j*l*m + 737280*a^9*b*c^6*g*j*k*m + 442368
*a^9*b*c^6*h*j*k*l + 36*a^3*b^12*c*f*h*k*m + 3096576*a^8*b*c^7*d*j*k*l - 12
745728*a^8*b*c^7*d*h*k*m + 3686400*a^8*b*c^7*e*f*l*m + 3391488*a^8*b*c^7*e*
h*j*m + 2211840*a^8*b*c^7*e*g*k*m + 1327104*a^8*b*c^7*e*h*k*l + 1228800*a^8
*b*c^7*f*g*j*m + 737280*a^8*b*c^7*f*h*j*l + 442368*a^8*b*c^7*g*h*j*k + 108*
a^2*b^13*c*d*h*k*m + 16367616*a^7*b*c^8*d*e*j*m + 9289728*a^7*b*c^8*d*e*k*l
 + 5160960*a^7*b*c^8*d*f*j*l + 3391488*a^7*b*c^8*e*f*j*k + 3096576*a^7*b*c^
8*d*g*j*k - 19307520*a^7*b*c^8*d*f*h*m + 3686400*a^7*b*c^8*e*f*g*m + 221184
0*a^7*b*c^8*e*f*h*l + 1327104*a^7*b*c^8*e*g*h*k + 737280*a^7*b*c^8*f*g*h*j
- 180*a*b^13*c^2*d*f*h*m - 540*a*b^12*c^3*d*f*h*k + 15482880*a^6*b*c^9*d*e*
f*l + 11059200*a^6*b*c^9*d*e*h*j + 9289728*a^6*b*c^9*d*e*g*k + 5160960*a^6*
b*c^9*d*f*g*j - 2304*a*b^11*c^4*d*f*g*j + 2211840*a^6*b*c^9*e*f*g*h + 4608*
a*b^10*c^5*d*e*f*j + 15482880*a^5*b*c^10*d*e*f*g - 13824*a*b^9*c^6*d*e*f*g
+ 36*a*b^14*c*d*f*k*m + 1843200*a^9*b^3*c^4*j*k*l*m + 783360*a^8*b^5*c^3*j*
k*l*m + 18432*a^7*b^7*c^2*j*k*l*m - 2211840*a^8*b^4*c^4*g*k*l*m - 1695744*a
^9*b^2*c^5*h*j*l*m - 1400832*a^8*b^4*c^4*h*j*l*m - 1105920*a^9*b^2*c^5*g*k*
l*m - 253440*a^7*b^6*c^3*h*j*l*m - 69120*a^7*b^6*c^3*g*k*l*m + 11520*a^6*b^
8*c^2*h*j*l*m + 6912*a^6*b^8*c^2*g*k*l*m + 4423680*a^8*b^3*c^5*e*k*l*m + 25
06752*a^8*b^3*c^5*f*j*l*m + 1843200*a^8*b^3*c^5*g*j*k*m + 1327104*a^8*b^3*c
^5*h*j*k*l + 838656*a^7*b^5*c^4*f*j*l*m + 783360*a^7*b^5*c^4*g*j*k*m + 6912
00*a^7*b^5*c^4*h*j*k*l + 138240*a^7*b^5*c^4*e*k*l*m + 69120*a^6*b^7*c^3*h*j
*k*l - 53760*a^6*b^7*c^3*f*j*l*m + 18432*a^6*b^7*c^3*g*j*k*m - 13824*a^6*b^
7*c^3*e*k*l*m - 2304*a^5*b^9*c^2*g*j*k*m + 2543616*a^8*b^3*c^5*g*h*l*m + 82
9440*a^7*b^5*c^4*g*h*l*m - 34560*a^6*b^7*c^3*g*h*l*m - 8183808*a^8*b^2*c^6*
d*j*l*m - 3686400*a^8*b^2*c^6*e*j*k*m - 2285568*a^7*b^4*c^5*d*j*l*m - 16957
44*a^8*b^2*c^6*f*j*k*l - 1566720*a^7*b^4*c^5*e*j*k*m - 1400832*a^7*b^4*c^5*
f*j*k*l + 741888*a^6*b^6*c^4*d*j*l*m - 253440*a^6*b^6*c^4*f*j*k*l - 80640*a
^5*b^8*c^3*d*j*l*m - 36864*a^6*b^6*c^4*e*j*k*m + 11520*a^5*b^8*c^3*f*j*k*l
+ 4608*a^5*b^8*c^3*e*j*k*m + 6700032*a^8*b^2*c^6*f*h*k*m + 5103360*a^7*b^4*
c^5*f*h*k*m - 5087232*a^8*b^2*c^6*e*h*l*m - 2838528*a^7*b^4*c^5*f*g*l*m - 1
843200*a^8*b^2*c^6*f*g*l*m - 1695744*a^8*b^2*c^6*g*h*j*m - 1658880*a^7*b^4*
c^5*g*h*k*l - 1658880*a^7*b^4*c^5*e*h*l*m - 1400832*a^7*b^4*c^5*g*h*j*m - 6
63552*a^8*b^2*c^6*g*h*k*l + 483840*a^6*b^6*c^4*f*h*k*m - 253440*a^6*b^6*c^4
*g*h*j*m - 207360*a^6*b^6*c^4*g*h*k*l + 161280*a^6*b^6*c^4*f*g*l*m + 69120*
a^6*b^6*c^4*e*h*l*m - 50040*a^5*b^8*c^3*f*h*k*m + 11520*a^5*b^8*c^3*g*h*j*m
 + 180*a^4*b^10*c^2*f*h*k*m + 4202496*a^7*b^3*c^6*d*j*k*l + 635904*a^6*b^5*
c^5*d*j*k*l - 276480*a^5*b^7*c^4*d*j*k*l + 34560*a^4*b^9*c^3*d*j*k*l - 1667
1744*a^7*b^3*c^6*d*h*k*m + 12275712*a^7*b^3*c^6*d*g*l*m + 5677056*a^7*b^3*c
^6*e*f*l*m + 4423680*a^7*b^3*c^6*e*g*k*m + 3317760*a^7*b^3*c^6*e*h*k*l + 28
01664*a^7*b^3*c^6*e*h*j*m - 2709504*a^6*b^5*c^5*d*g*l*m + 2543616*a^7*b^3*c
^6*f*g*k*l + 2506752*a^7*b^3*c^6*f*g*j*m + 1843200*a^7*b^3*c^6*f*h*j*l + 13
27104*a^7*b^3*c^6*g*h*j*k + 838656*a^6*b^5*c^5*f*g*j*m + 829440*a^6*b^5*c^5
*f*g*k*l + 783360*a^6*b^5*c^5*f*h*j*l + 691200*a^6*b^5*c^5*g*h*j*k + 665280

*a^5*b^7*c^4*d*h*k*m + 506880*a^6*b^5*c^5*e*h*j*m + 414720*a^6*b^5*c^5*e*h*
k*l - 322560*a^6*b^5*c^5*e*f*l*m + 241920*a^5*b^7*c^4*d*g*l*m + 138240*a^6*
b^5*c^5*e*g*k*m - 108540*a^4*b^9*c^3*d*h*k*m + 69120*a^5*b^7*c^4*g*h*j*k -
53760*a^5*b^7*c^4*f*g*j*m - 51840*a^6*b^5*c^5*d*h*k*m - 34560*a^5*b^7*c^4*f
*g*k*l - 23040*a^5*b^7*c^4*e*h*j*m + 18432*a^5*b^7*c^4*f*h*j*l - 13824*a^5*
b^7*c^4*e*g*k*m - 2304*a^4*b^9*c^3*f*h*j*l + 1296*a^3*b^11*c^2*d*h*k*m + 31
924224*a^7*b^2*c^7*d*f*k*m - 24551424*a^7*b^2*c^7*d*e*l*m + 10616832*a^7*b^
2*c^7*e*g*j*l - 8183808*a^7*b^2*c^7*d*g*j*m - 5529600*a^7*b^2*c^7*d*h*j*l +
5419008*a^6*b^4*c^6*d*e*l*m + 5308416*a^6*b^4*c^6*e*g*j*l - 5087232*a^7*b^
2*c^7*e*f*k*l - 5013504*a^7*b^2*c^7*e*f*j*m + 4868352*a^6*b^4*c^6*d*f*k*m -
4644864*a^7*b^2*c^7*d*g*k*l - 3981312*a^6*b^4*c^6*d*g*k*l - 2654208*a^7*b^
2*c^7*e*h*j*k - 2367360*a^5*b^6*c^5*d*f*k*m - 2285568*a^6*b^4*c^6*d*g*j*m -
2211840*a^6*b^4*c^6*d*h*j*l - 1695744*a^7*b^2*c^7*f*g*j*k - 1677312*a^6*b^
4*c^6*e*f*j*m - 1658880*a^6*b^4*c^6*e*f*k*l - 1400832*a^6*b^4*c^6*f*g*j*k -
1382400*a^6*b^4*c^6*e*h*j*k + 1036800*a^5*b^6*c^5*d*g*k*l + 741888*a^5*b^6
*c^5*d*g*j*m - 483840*a^5*b^6*c^5*d*e*l*m + 317952*a^5*b^6*c^5*d*h*j*l + 26
8920*a^4*b^8*c^4*d*f*k*m - 253440*a^5*b^6*c^5*f*g*j*k - 138240*a^5*b^6*c^5*
e*h*j*k + 107520*a^5*b^6*c^5*e*f*j*m - 103680*a^4*b^8*c^4*d*g*k*l - 80640*a
^4*b^8*c^4*d*g*j*m + 69120*a^5*b^6*c^5*e*f*k*l + 11520*a^4*b^8*c^4*f*g*j*k
+ 6912*a^4*b^8*c^4*d*h*j*l - 6912*a^3*b^10*c^3*d*h*j*l + 6120*a^3*b^10*c^3*
d*f*k*m - 1368*a^2*b^12*c^2*d*f*k*m - 5087232*a^7*b^2*c^7*e*g*h*m - 2211840
*a^6*b^4*c^6*f*g*h*l - 1658880*a^6*b^4*c^6*e*g*h*m - 1105920*a^7*b^2*c^7*f*
g*h*l - 69120*a^5*b^6*c^5*f*g*h*l + 69120*a^5*b^6*c^5*e*g*h*m + 6912*a^4*b^
8*c^4*f*g*h*l + 7962624*a^6*b^3*c^7*d*e*k*l - 22164480*a^6*b^3*c^7*d*f*h*m
+ 5160960*a^6*b^3*c^7*d*f*j*l + 4571136*a^6*b^3*c^7*d*e*j*m + 4202496*a^6*b
^3*c^7*d*g*j*k + 2801664*a^6*b^3*c^7*e*f*j*k - 2073600*a^5*b^5*c^6*d*e*k*l
- 1483776*a^5*b^5*c^6*d*e*j*m + 635904*a^5*b^5*c^6*d*g*j*k + 506880*a^5*b^5
*c^6*e*f*j*k - 354816*a^4*b^7*c^5*d*f*j*l + 322560*a^5*b^5*c^6*d*f*j*l - 27
6480*a^4*b^7*c^5*d*g*j*k + 207360*a^4*b^7*c^5*d*e*k*l + 161280*a^4*b^7*c^5*
d*e*j*m + 59904*a^3*b^9*c^4*d*f*j*l + 34560*a^3*b^9*c^4*d*g*j*k - 23040*a^4
*b^7*c^5*e*f*j*k - 2304*a^2*b^11*c^3*d*f*j*l + 8294400*a^6*b^3*c^7*d*g*h*l
+ 5677056*a^6*b^3*c^7*e*f*g*m + 4423680*a^6*b^3*c^7*e*f*h*l + 3317760*a^6*b
^3*c^7*e*g*h*k + 2805120*a^5*b^5*c^6*d*f*h*m + 1843200*a^6*b^3*c^7*f*g*h*j
- 829440*a^5*b^5*c^6*d*g*h*l + 783360*a^5*b^5*c^6*f*g*h*j + 437184*a^4*b^7*
c^5*d*f*h*m + 414720*a^5*b^5*c^6*e*g*h*k - 322560*a^5*b^5*c^6*e*f*g*m - 146
268*a^3*b^9*c^4*d*f*h*m + 138240*a^5*b^5*c^6*e*f*h*l - 62208*a^4*b^7*c^5*d*
g*h*l + 20736*a^3*b^9*c^4*d*g*h*l + 18432*a^4*b^7*c^5*f*g*h*j - 13824*a^4*b
^7*c^5*e*f*h*l + 9360*a^2*b^11*c^3*d*f*h*m - 2304*a^3*b^9*c^4*f*g*h*j - 840
4992*a^6*b^2*c^8*d*e*j*k - 24551424*a^6*b^2*c^8*d*e*g*m + 21150720*a^6*b^2*
c^8*d*f*h*k - 1271808*a^5*b^4*c^7*d*e*j*k + 552960*a^4*b^6*c^6*d*e*j*k - 69
120*a^3*b^8*c^5*d*e*j*k - 16588800*a^6*b^2*c^8*d*e*h*l - 7741440*a^6*b^2*c^
8*d*f*g*l + 6946560*a^5*b^4*c^7*d*f*h*k - 5529600*a^6*b^2*c^8*d*g*h*j + 541
9008*a^5*b^4*c^7*d*e*g*m - 5087232*a^6*b^2*c^8*e*f*g*k - 3870720*a^5*b^4*c^
7*d*f*g*l - 3686400*a^6*b^2*c^8*e*f*h*j - 2211840*a^5*b^4*c^7*d*g*h*j - 175
5648*a^4*b^6*c^6*d*f*h*k - 1658880*a^5*b^4*c^7*e*f*g*k + 1658880*a^5*b^4*c^

7*d*e*h*l - 1566720*a^5*b^4*c^7*e*f*h*j + 1451520*a^4*b^6*c^6*d*f*g*l - 483
840*a^4*b^6*c^6*d*e*g*m + 317952*a^4*b^6*c^6*d*g*h*j - 193536*a^3*b^8*c^5*d
*f*g*l + 124416*a^4*b^6*c^6*d*e*h*l + 114696*a^3*b^8*c^5*d*f*h*k + 69120*a^
4*b^6*c^6*e*f*g*k - 41472*a^3*b^8*c^5*d*e*h*l - 36864*a^4*b^6*c^6*e*f*h*j +
 14580*a^2*b^10*c^4*d*f*h*k + 6912*a^3*b^8*c^5*d*g*h*j - 6912*a^2*b^10*c^4*
d*g*h*j + 6912*a^2*b^10*c^4*d*f*g*l + 4608*a^3*b^8*c^5*e*f*h*j + 7962624*a^
5*b^3*c^8*d*e*g*k + 7741440*a^5*b^3*c^8*d*e*f*l + 5160960*a^5*b^3*c^8*d*f*g
*j + 4423680*a^5*b^3*c^8*d*e*h*j - 2903040*a^4*b^5*c^7*d*e*f*l - 2073600*a^
4*b^5*c^7*d*e*g*k - 635904*a^4*b^5*c^7*d*e*h*j + 387072*a^3*b^7*c^6*d*e*f*l
 - 354816*a^3*b^7*c^6*d*f*g*j + 322560*a^4*b^5*c^7*d*f*g*j + 207360*a^3*b^7
*c^6*d*e*g*k + 59904*a^2*b^9*c^5*d*f*g*j - 13824*a^3*b^7*c^6*d*e*h*j + 1382
4*a^2*b^9*c^5*d*e*h*j - 13824*a^2*b^9*c^5*d*e*f*l + 4423680*a^5*b^3*c^8*e*f
*g*h + 138240*a^4*b^5*c^7*e*f*g*h - 13824*a^3*b^7*c^6*e*f*g*h - 10321920*a^
5*b^2*c^9*d*e*f*j + 709632*a^3*b^6*c^7*d*e*f*j - 645120*a^4*b^4*c^8*d*e*f*j
 - 119808*a^2*b^8*c^6*d*e*f*j - 16588800*a^5*b^2*c^9*d*e*g*h + 1658880*a^4*
b^4*c^8*d*e*g*h + 124416*a^3*b^6*c^7*d*e*g*h - 41472*a^2*b^8*c^6*d*e*g*h +
7741440*a^4*b^3*c^9*d*e*f*g - 2903040*a^3*b^5*c^8*d*e*f*g + 387072*a^2*b^7*
c^7*d*e*f*g + 3456*a^7*b^8*c*k*l^2*m + 12672*a^7*b^8*c*j*l*m^2 + 384*a^5*b^
10*c*j^2*k*m - 1635840*a^10*b*c^5*h*k*m^2 - 1009152*a^9*b*c^6*h^2*k*m + 369
0*a^6*b^9*c*h*k*m^2 + 1152*a^6*b^9*c*g*l*m^2 - 540*a^5*b^10*c*h*k^2*m + 54*
a^4*b^11*c*h^2*k*m + 565248*a^9*b*c^6*h*j^2*m - 39771648*a^7*b*c^8*d^2*k*m
- 2496000*a^8*b*c^7*f^2*k*m - 1543680*a^9*b*c^6*f*k^2*m + 1980*a^5*b^10*c*f
*k*m^2 - 384*a^5*b^10*c*g*j*m^2 - 180*a^4*b^11*c*f*k^2*m + 6*a^2*b^13*c*f^2
*k*m - 10298880*a^9*b*c^6*d*k*m^2 + 2580480*a^9*b*c^6*e*j*m^2 + 5310*a^4*b^
11*c*d*k*m^2 - 1674*a*b^13*c^2*d^2*k*m - 540*a^3*b^12*c*d*k^2*m - 10616832*
a^7*b*c^8*e^2*j*l - 3538944*a^8*b*c^7*e*j^2*l + 2727936*a^8*b*c^7*d*j^2*m -
 2496000*a^9*b*c^6*f*h*m^2 - 1543680*a^8*b*c^7*f*h^2*m + 565248*a^8*b*c^7*f
*j^2*k - 270*a^4*b^11*c*f*h*m^2 - 59512320*a^6*b*c^9*d^2*f*m + 5087232*a^7*
b*c^8*e^2*h*m + 1105920*a^8*b*c^7*e*j*k^2 - 3456*a*b^12*c^3*d^2*j*l - 16358
40*a^7*b*c^8*f^2*h*k - 1009152*a^8*b*c^7*f*h*k^2 + 10260*a*b^12*c^3*d^2*h*m
 - 684*a^3*b^12*c*d*h*m^2 - 24675840*a^6*b*c^9*d^2*h*k - 15552000*a^8*b*c^7
*d*f*m^2 + 24551424*a^6*b*c^9*d*e^2*m - 3939840*a^7*b*c^8*d*h^2*k + 1105920
*a^7*b*c^8*e*h^2*j - 25074*a*b^11*c^4*d^2*f*m + 10530*a*b^11*c^4*d^2*h*k +
10368*a*b^11*c^4*d^2*g*l + 420*a*b^12*c^3*d*f^2*m - 378*a^2*b^13*c*d*f*m^2
- 10616832*a^6*b*c^9*e^2*g*j + 5087232*a^6*b*c^9*e^2*f*k - 3538944*a^7*b*c^
8*e*g*j^2 + 1843200*a^7*b*c^8*d*h*j^2 - 7994880*a^6*b*c^9*d*f^2*k - 4990464
*a^7*b*c^8*d*f*k^2 + 2580480*a^6*b*c^9*e*f^2*j + 65664*a*b^10*c^5*d^2*g*j -
 27972*a*b^10*c^5*d^2*f*k - 20736*a*b^10*c^5*d^2*e*l + 1260*a*b^11*c^4*d*f^
2*k + 54*a*b^13*c^2*d*f*k^2 + 23224320*a^5*b*c^10*d^2*e*j - 37062144*a^5*b*
c^10*d^2*f*h + 384*a*b^12*c^3*d*f*j^2 - 131328*a*b^9*c^6*d^2*e*j - 5985792*
a^6*b*c^9*d*f*h^2 + 206010*a*b^9*c^6*d^2*f*h - 6300*a*b^10*c^5*d*f^2*h + 13
50*a*b^11*c^4*d*f*h^2 + 16588800*a^5*b*c^10*d*e^2*h + 3456*a*b^10*c^5*d*f*g
^2 + 435456*a*b^8*c^7*d^2*e*g + 13824*a*b^8*c^7*d*e^2*f - 1474560*a^9*c^7*e
*j*k*m + 460800*a^9*c^7*f*h*k*m + 3225600*a^8*c^8*d*f*k*m - 2457600*a^8*c^8
*e*f*j*m - 884736*a^8*c^8*e*h*j*k - 6193152*a^7*c^9*d*e*j*k + 1935360*a^7*c

^9*d*f*h*k - 1474560*a^7*c^9*e*f*h*j - 10321920*a^6*c^10*d*e*f*j - 1105920*
a^9*b^4*c^3*k*l^2*m - 552960*a^10*b^2*c^4*k*l^2*m - 34560*a^8*b^6*c^2*k*l^2
*m - 1290240*a^10*b^2*c^4*j*l*m^2 - 860160*a^9*b^4*c^3*j*l*m^2 - 80640*a^8*
b^6*c^2*j*l*m^2 - 737280*a^9*b^2*c^5*j^2*k*m - 568320*a^8*b^4*c^4*j^2*k*m -
136704*a^7*b^6*c^3*j^2*k*m - 2304*a^6*b^8*c^2*j^2*k*m + 1271808*a^9*b^3*c^
4*h*l^2*m - 552960*a^9*b^2*c^5*j*k^2*l - 552960*a^8*b^4*c^4*j*k^2*l + 41472
0*a^8*b^5*c^3*h*l^2*m - 145152*a^7*b^6*c^3*j*k^2*l - 17280*a^7*b^7*c^2*h*l^
2*m - 3456*a^6*b^8*c^2*j*k^2*l - 3640320*a^9*b^3*c^4*h*k*m^2 - 2626560*a^8*
b^3*c^5*h^2*k*m + 2211840*a^9*b^2*c^5*h*k^2*m + 2056320*a^8*b^4*c^4*h*k^2*m
+ 1935360*a^9*b^3*c^4*g*l*m^2 - 1143360*a^8*b^5*c^3*h*k*m^2 - 1097280*a^7*
b^5*c^4*h^2*k*m + 364608*a^7*b^6*c^3*h*k^2*m + 322560*a^8*b^5*c^3*g*l*m^2 -
56160*a^6*b^7*c^3*h^2*k*m - 40320*a^7*b^7*c^2*g*l*m^2 + 27936*a^7*b^7*c^2*
h*k*m^2 - 3780*a^6*b^8*c^2*h*k^2*m + 2970*a^5*b^9*c^2*h^2*k*m - 1419264*a^8
*b^4*c^4*f*l^2*m - 1105920*a^7*b^4*c^5*g^2*k*m - 921600*a^9*b^2*c^5*f*l^2*m
- 829440*a^8*b^4*c^4*h*k*l^2 + 749568*a^8*b^3*c^5*h*j^2*m - 552960*a^8*b^2
*c^6*g^2*k*m - 331776*a^9*b^2*c^5*h*k*l^2 + 317952*a^7*b^5*c^4*h*j^2*m - 10
3680*a^7*b^6*c^3*h*k*l^2 + 80640*a^7*b^6*c^3*f*l^2*m + 38400*a^6*b^7*c^3*h*
j^2*m - 34560*a^6*b^6*c^4*g^2*k*m + 3456*a^5*b^8*c^3*g^2*k*m - 1920*a^5*b^9
*c^2*h*j^2*m - 5142528*a^7*b^3*c^6*f^2*k*m + 5068800*a^9*b^2*c^5*f*k*m^2 -
3870720*a^9*b^2*c^5*e*l*m^2 - 3755520*a^8*b^3*c^5*f*k^2*m + 3000960*a^8*b^4
*c^4*f*k*m^2 - 1290240*a^9*b^2*c^5*g*j*m^2 - 1085760*a^7*b^5*c^4*f*k^2*m -
959040*a^6*b^5*c^5*f^2*k*m - 860160*a^8*b^4*c^4*g*j*m^2 + 829440*a^8*b^3*c^
5*g*k^2*l - 645120*a^8*b^4*c^4*e*l*m^2 - 552960*a^8*b^2*c^6*h^2*j*l - 55296
0*a^7*b^4*c^5*h^2*j*l + 414720*a^7*b^5*c^4*g*k^2*l - 145152*a^6*b^6*c^4*h^2
*j*l + 103200*a^5*b^7*c^4*f^2*k*m - 80640*a^7*b^6*c^3*g*j*m^2 + 80640*a^7*b
^6*c^3*e*l*m^2 + 41280*a^7*b^6*c^3*f*k*m^2 - 37188*a^6*b^8*c^2*f*k*m^2 + 13
536*a^6*b^7*c^3*f*k^2*m + 12672*a^6*b^8*c^2*g*j*m^2 + 10368*a^6*b^7*c^3*g*k
^2*l + 5490*a^5*b^9*c^2*f*k^2*m - 3456*a^5*b^8*c^3*h^2*j*l - 2304*a^6*b^8*c
^2*e*l*m^2 + 810*a^4*b^9*c^3*f^2*k*m - 270*a^3*b^11*c^2*f^2*k*m + 6137856*a
^8*b^3*c^5*d*l^2*m - 4423680*a^7*b^2*c^7*e^2*k*m - 2654208*a^8*b^3*c^5*g*j*
l^2 - 2654208*a^7*b^3*c^6*g^2*j*l + 1769472*a^8*b^2*c^6*g*j^2*l + 1769472*a
^7*b^4*c^5*g*j^2*l - 1354752*a^7*b^5*c^4*d*l^2*m - 1327104*a^7*b^5*c^4*g*j*
l^2 - 1327104*a^6*b^5*c^5*g^2*j*l + 1271808*a^8*b^3*c^5*f*k*l^2 - 1040384*a
^8*b^2*c^6*f*j^2*m - 697344*a^7*b^4*c^5*f*j^2*m - 516096*a^8*b^2*c^6*h*j^2*
k - 451584*a^7*b^4*c^5*h*j^2*k + 442368*a^6*b^6*c^4*g*j^2*l + 414720*a^7*b^
5*c^4*f*k*l^2 - 138240*a^6*b^6*c^4*h*j^2*k - 138240*a^6*b^4*c^6*e^2*k*m - 1
21856*a^6*b^6*c^4*f*j^2*m + 120960*a^6*b^7*c^3*d*l^2*m - 17280*a^6*b^7*c^3*
f*k*l^2 + 13824*a^5*b^6*c^5*e^2*k*m - 11520*a^5*b^8*c^3*h*j^2*k + 8960*a^5*
b^8*c^3*f*j^2*m + 10851840*a^8*b^2*c^6*d*k^2*m - 10464768*a^6*b^3*c^7*d^2*k
*m - 10275840*a^8*b^3*c^5*d*k*m^2 + 7121088*a^5*b^5*c^6*d^2*k*m + 3127680*a
^7*b^4*c^5*d*k^2*m + 1720320*a^8*b^3*c^5*e*j*m^2 - 1658880*a^8*b^2*c^6*e*k^
2*l - 1290240*a^7*b^2*c^7*f^2*j*l + 1271808*a^7*b^3*c^6*g^2*h*m - 1222560*a
^4*b^7*c^5*d^2*k*m + 999360*a^7*b^5*c^4*d*k*m^2 - 860160*a^6*b^4*c^6*f^2*j*
l - 829440*a^7*b^4*c^5*e*k^2*l - 705024*a^6*b^6*c^4*d*k^2*m - 552960*a^8*b^
2*c^6*g*j*k^2 - 552960*a^7*b^4*c^5*g*j*k^2 + 414720*a^6*b^5*c^5*g^2*h*m + 3

19392*a^6*b^7*c^3*d*k*m^2 + 161280*a^7*b^5*c^4*e*j*m^2 - 145152*a^6*b^6*c^4
*g*j*k^2 - 85734*a^5*b^9*c^2*d*k*m^2 - 80640*a^5*b^6*c^5*f^2*j*l - 25344*a^
6*b^7*c^3*e*j*m^2 + 23490*a^3*b^9*c^4*d^2*k*m - 20736*a^6*b^6*c^4*e*k^2*l -
 17280*a^5*b^7*c^4*g^2*h*m + 14148*a^5*b^8*c^3*d*k^2*m + 13716*a^2*b^11*c^3
*d^2*k*m + 12690*a^4*b^10*c^2*d*k^2*m + 12672*a^4*b^8*c^4*f^2*j*l - 3456*a^
5*b^8*c^3*g*j*k^2 + 768*a^5*b^9*c^2*e*j*m^2 - 384*a^3*b^10*c^3*f^2*j*l + 53
08416*a^8*b^2*c^6*e*j*l^2 - 5308416*a^6*b^3*c^7*e^2*j*l - 5142528*a^8*b^3*c
^5*f*h*m^2 + 5068800*a^7*b^2*c^7*f^2*h*m - 3755520*a^7*b^3*c^6*f*h^2*m - 35
38944*a^7*b^3*c^6*e*j^2*l + 3000960*a^6*b^4*c^6*f^2*h*m + 2654208*a^7*b^4*c
^5*e*j*l^2 - 2322432*a^8*b^2*c^6*d*k*l^2 + 2125824*a^7*b^3*c^6*d*j^2*m - 19
90656*a^7*b^4*c^5*d*k*l^2 - 1085760*a^6*b^5*c^5*f*h^2*m - 959040*a^7*b^5*c^
4*f*h*m^2 - 884736*a^6*b^5*c^5*e*j^2*l + 829440*a^7*b^3*c^6*g*h^2*l + 74956
8*a^7*b^3*c^6*f*j^2*k + 518400*a^6*b^6*c^4*d*k*l^2 + 414720*a^6*b^5*c^5*g*h
^2*l + 317952*a^6*b^5*c^5*f*j^2*k + 133632*a^6*b^5*c^5*d*j^2*m + 103200*a^6
*b^7*c^3*f*h*m^2 - 96768*a^5*b^7*c^4*d*j^2*m - 51840*a^5*b^8*c^3*d*k*l^2 +
41280*a^5*b^6*c^5*f^2*h*m + 38400*a^5*b^7*c^4*f*j^2*k - 37188*a^4*b^8*c^4*f
^2*h*m + 13536*a^5*b^7*c^4*f*h^2*m + 13440*a^4*b^9*c^3*d*j^2*m + 10368*a^5*
b^7*c^4*g*h^2*l + 5490*a^4*b^9*c^3*f*h^2*m + 1980*a^3*b^10*c^3*f^2*h*m - 19
20*a^4*b^9*c^3*f*j^2*k + 810*a^5*b^9*c^2*f*h*m^2 - 180*a^3*b^11*c^2*f*h^2*m
 - 30*a^2*b^12*c^2*f^2*h*m + 30067200*a^6*b^2*c^8*d^2*h*m - 11612160*a^6*b^
2*c^8*d^2*j*l + 1658880*a^6*b^3*c^7*e^2*h*m + 1596672*a^4*b^6*c^6*d^2*j*l -
 1419264*a^6*b^4*c^6*f*g^2*m - 1105920*a^7*b^4*c^5*f*h*l^2 + 1105920*a^7*b^
3*c^6*e*j*k^2 - 921600*a^7*b^2*c^7*f*g^2*m - 829440*a^6*b^4*c^6*g^2*h*k - 5
52960*a^8*b^2*c^6*f*h*l^2 - 508032*a^3*b^8*c^5*d^2*j*l - 331776*a^7*b^2*c^7
*g^2*h*k + 290304*a^6*b^5*c^5*e*j*k^2 - 103680*a^5*b^6*c^5*g^2*h*k + 80640*
a^5*b^6*c^5*f*g^2*m - 69120*a^5*b^5*c^6*e^2*h*m + 65664*a^2*b^10*c^4*d^2*j*
l - 34560*a^6*b^6*c^4*f*h*l^2 + 6912*a^5*b^7*c^4*e*j*k^2 + 3456*a^5*b^8*c^3
*f*h*l^2 + 11930112*a^8*b^2*c^6*d*h*m^2 + 8432640*a^7*b^2*c^7*d*h^2*m + 445
0176*a^7*b^4*c^5*d*h*m^2 + 4337280*a^6*b^4*c^6*d*h^2*m - 3870720*a^8*b^2*c^
6*e*g*m^2 - 3640320*a^6*b^3*c^7*f^2*h*k - 2885760*a^5*b^4*c^7*d^2*h*m - 284
4288*a^4*b^6*c^6*d^2*h*m - 2626560*a^7*b^3*c^6*f*h*k^2 + 2211840*a^7*b^2*c^
7*f*h^2*k + 2056320*a^6*b^4*c^6*f*h^2*k + 1935360*a^6*b^3*c^7*f^2*g*l - 191
6928*a^7*b^2*c^7*d*j^2*k - 1687680*a^6*b^6*c^4*d*h*m^2 - 1658880*a^7*b^2*c^
7*e*h^2*l - 1143360*a^5*b^5*c^6*f^2*h*k - 1097280*a^6*b^5*c^5*f*h*k^2 + 101
9412*a^3*b^8*c^5*d^2*h*m - 1007424*a^5*b^6*c^5*d*h^2*m - 912384*a^6*b^4*c^6
*d*j^2*k - 829440*a^6*b^4*c^6*e*h^2*l - 645120*a^7*b^4*c^5*e*g*m^2 - 552960
*a^7*b^2*c^7*g*h^2*j - 552960*a^6*b^4*c^6*g*h^2*j + 364608*a^5*b^6*c^5*f*h^
2*k + 322560*a^5*b^5*c^6*f^2*g*l + 197460*a^5*b^8*c^3*d*h*m^2 - 145152*a^5*
b^6*c^5*g*h^2*j - 143802*a^2*b^10*c^4*d^2*h*m + 80640*a^6*b^6*c^4*e*g*m^2 -
 56160*a^5*b^7*c^4*f*h*k^2 + 51948*a^4*b^8*c^4*d*h^2*m - 40320*a^4*b^7*c^5*
f^2*g*l + 34560*a^4*b^8*c^4*d*j^2*k + 27936*a^4*b^7*c^5*f^2*h*k - 20736*a^5
*b^6*c^5*e*h^2*l - 13824*a^5*b^6*c^5*d*j^2*k + 10800*a^3*b^10*c^3*d*h^2*m -
 5760*a^3*b^10*c^3*d*j^2*k - 3780*a^4*b^8*c^4*f*h^2*k + 3690*a^3*b^9*c^4*f^
2*h*k - 3456*a^4*b^8*c^4*g*h^2*j + 2970*a^4*b^9*c^3*f*h*k^2 - 2304*a^5*b^8*
c^3*e*g*m^2 + 1152*a^3*b^9*c^4*f^2*g*l - 540*a^3*b^10*c^3*f*h^2*k - 540*a^2

*b^12*c^2*d*h^2*m - 90*a^4*b^10*c^2*d*h*m^2 - 90*a^2*b^11*c^3*f^2*h*k + 54*
a^3*b^11*c^2*f*h*k^2 + 15925248*a^6*b^2*c^8*e^2*g*l - 7962624*a^7*b^3*c^6*e
*g*l^2 - 7962624*a^6*b^3*c^7*e*g^2*l + 23385600*a^6*b^2*c^8*d*f^2*m + 61378
56*a^6*b^3*c^7*d*g^2*m - 5677056*a^6*b^2*c^8*e^2*f*m + 4147200*a^7*b^3*c^6*
d*h*l^2 - 3317760*a^6*b^2*c^8*e^2*h*k - 1354752*a^5*b^5*c^6*d*g^2*m + 12718
08*a^6*b^3*c^7*f*g^2*k - 737280*a^7*b^2*c^7*f*h*j^2 + 17418240*a^5*b^3*c^8*
d^2*g*l - 568320*a^6*b^4*c^6*f*h*j^2 - 414720*a^6*b^5*c^5*d*h*l^2 + 414720*
a^5*b^5*c^6*f*g^2*k - 414720*a^5*b^4*c^7*e^2*h*k + 322560*a^5*b^4*c^7*e^2*f
*m - 136704*a^5*b^6*c^5*f*h*j^2 + 120960*a^4*b^7*c^5*d*g^2*m - 31104*a^5*b^
7*c^4*d*h*l^2 - 17280*a^4*b^7*c^5*f*g^2*k + 10368*a^4*b^9*c^3*d*h*l^2 - 230
4*a^4*b^8*c^4*f*h*j^2 + 384*a^3*b^10*c^3*f*h*j^2 + 50042880*a^5*b^2*c^9*d^2
*f*k - 13271040*a^5*b^3*c^8*d^2*h*k - 13149696*a^7*b^3*c^6*d*f*m^2 + 109065
60*a^4*b^5*c^7*d^2*f*m - 8709120*a^4*b^5*c^7*d^2*g*l - 7418880*a^5*b^3*c^8*
d^2*f*m + 7133184*a^7*b^2*c^7*d*h*k^2 - 6428160*a^6*b^3*c^7*d*h^2*k + 55935
36*a^4*b^5*c^7*d^2*h*k - 3870720*a^6*b^2*c^8*e*f^2*l + 3369600*a^6*b^4*c^6*
d*h*k^2 + 3148992*a^6*b^5*c^5*d*f*m^2 - 2985696*a^3*b^7*c^6*d^2*f*m + 19595
52*a^3*b^7*c^6*d^2*g*l - 1658880*a^7*b^2*c^7*e*g*k^2 - 1505280*a^4*b^6*c^6*
d*f^2*m - 1290240*a^6*b^2*c^8*f^2*g*j - 34836480*a^5*b^2*c^9*d^2*e*l + 1105
920*a^6*b^3*c^7*e*h^2*j - 860160*a^5*b^4*c^7*f^2*g*j - 829440*a^6*b^4*c^6*e
*g*k^2 - 692064*a^3*b^7*c^6*d^2*h*k - 689472*a^5*b^5*c^6*d*h^2*k - 645120*a
^5*b^4*c^7*e*f^2*l - 388800*a^5*b^6*c^5*d*h*k^2 + 378954*a^2*b^9*c^5*d^2*f*
m + 362880*a^5*b^4*c^7*d*f^2*m + 296964*a^3*b^8*c^5*d*f^2*m + 290304*a^5*b^
5*c^6*e*h^2*j + 277344*a^4*b^7*c^5*d*h^2*k - 217728*a^2*b^9*c^5*d^2*g*l - 8
0640*a^4*b^6*c^6*f^2*g*j + 80640*a^4*b^6*c^6*e*f^2*l - 77070*a^4*b^9*c^3*d*
f*m^2 - 30240*a^5*b^7*c^4*d*f*m^2 - 28350*a^3*b^9*c^4*d*h^2*k - 26406*a^2*b
^9*c^5*d^2*h*k - 21060*a^4*b^8*c^4*d*h*k^2 - 20736*a^5*b^6*c^5*e*g*k^2 - 19
278*a^2*b^10*c^4*d*f^2*m + 12672*a^3*b^8*c^5*f^2*g*j + 10044*a^3*b^10*c^3*d
*h*k^2 + 8820*a^3*b^11*c^2*d*f*m^2 + 6912*a^4*b^7*c^5*e*h^2*j - 2304*a^3*b^
8*c^5*e*f^2*l - 1620*a^2*b^11*c^3*d*h^2*k - 384*a^2*b^10*c^4*f^2*g*j + 162*
a^2*b^12*c^2*d*h*k^2 - 5419008*a^5*b^3*c^8*d*e^2*m + 5308416*a^6*b^2*c^8*e*
g^2*j - 5308416*a^5*b^3*c^8*e^2*g*j - 3870720*a^7*b^2*c^7*d*f*l^2 - 3538944
*a^6*b^3*c^7*e*g*j^2 + 2654208*a^5*b^4*c^7*e*g^2*j - 2322432*a^6*b^2*c^8*d*
g^2*k - 1990656*a^5*b^4*c^7*d*g^2*k - 1935360*a^6*b^4*c^6*d*f*l^2 + 1658880
*a^6*b^3*c^7*d*h*j^2 + 1658880*a^5*b^3*c^8*e^2*f*k - 884736*a^5*b^5*c^6*e*g
*j^2 + 725760*a^5*b^6*c^5*d*f*l^2 + 17418240*a^4*b^4*c^8*d^2*e*l + 518400*a
^4*b^6*c^6*d*g^2*k + 483840*a^4*b^5*c^7*d*e^2*m + 262656*a^5*b^5*c^6*d*h*j^
2 - 96768*a^4*b^8*c^4*d*f*l^2 - 69120*a^4*b^5*c^7*e^2*f*k - 55296*a^4*b^7*c
^5*d*h*j^2 - 51840*a^3*b^8*c^5*d*g^2*k + 3456*a^3*b^10*c^3*d*f*l^2 + 1152*a
^3*b^9*c^4*d*h*j^2 + 1152*a^2*b^11*c^3*d*h*j^2 - 15431040*a^4*b^4*c^8*d^2*f
*k - 13248000*a^5*b^3*c^8*d*f^2*k - 11612160*a^5*b^2*c^9*d^2*g*j - 10063872
*a^6*b^3*c^7*d*f*k^2 - 3919104*a^3*b^6*c^7*d^2*e*l + 2554560*a^4*b^5*c^7*d*
f^2*k + 1720320*a^5*b^3*c^8*e*f^2*j + 1596672*a^3*b^6*c^7*d^2*g*j + 1518912
*a^3*b^6*c^7*d^2*f*k - 1105920*a^5*b^4*c^7*f*g^2*h + 838080*a^5*b^5*c^6*d*f
*k^2 - 552960*a^6*b^2*c^8*f*g^2*h - 508032*a^2*b^8*c^6*d^2*g*j + 435456*a^2
*b^8*c^6*d^2*e*l + 161280*a^4*b^5*c^7*e*f^2*j + 116640*a^4*b^7*c^5*d*f*k^2

+ 106812*a^2*b^8*c^6*d^2*f*k - 98208*a^3*b^7*c^6*d*f^2*k - 34560*a^4*b^6*c^6*f*g^2*h - 27270*a^3*b^9*c^4*d*f*k^2 - 26334*a^2*b^9*c^5*d*f^2*k - 25344*a^3*b^7*c^6*e*f^2*j + 3456*a^3*b^8*c^5*f*g^2*h + 768*a^2*b^9*c^5*e*f^2*j - 702*a^2*b^11*c^3*d*f*k^2 - 7962624*a^5*b^2*c^9*d*e^2*k - 2580480*a^6*b^2*c^8*d*f*j^2 + 2073600*a^4*b^4*c^8*d*e^2*k - 1658880*a^6*b^2*c^8*e*g*h^2 - 967680*a^5*b^4*c^7*d*f*j^2 - 829440*a^5*b^4*c^7*e*g*h^2 - 207360*a^3*b^6*c^7*d*e^2*k + 64512*a^4*b^6*c^6*d*f*j^2 + 39168*a^3*b^8*c^5*d*f*j^2 - 20736*a^4*b^6*c^6*e*g*h^2 - 9216*a^2*b^10*c^4*d*f*j^2 - 4423680*a^5*b^2*c^9*e^2*f*h + 4147200*a^5*b^3*c^8*d*g^2*h - 3193344*a^3*b^5*c^8*d^2*e*j + 1016064*a^2*b^7*c^7*d^2*e*j - 414720*a^4*b^5*c^7*d*g^2*h - 138240*a^4*b^4*c^8*e^2*f*h - 31104*a^3*b^7*c^6*d*g^2*h + 13824*a^3*b^6*c^7*e^2*f*h + 10368*a^2*b^9*c^5*d*g^2*h + 15630336*a^5*b^2*c^9*d*f^2*h - 14459904*a^4*b^3*c^9*d^2*f*h + 9630144*a^3*b^5*c^8*d^2*f*h - 8764416*a^5*b^3*c^8*d*f*h^2 - 3870720*a^5*b^2*c^9*e*f^2*g + 2867328*a^4*b^4*c^8*d*f^2*h - 2095200*a^2*b^7*c^7*d^2*f*h - 1414080*a^3*b^6*c^7*d*f^2*h - 34836480*a^4*b^2*c^10*d^2*e*g - 645120*a^4*b^4*c^8*e*f^2*g + 306720*a^3*b^7*c^6*d*f*h^2 + 197820*a^2*b^8*c^6*d*f^2*h + 146880*a^4*b^5*c^7*d*f*h^2 + 80640*a^3*b^6*c^7*e*f^2*g - 55350*a^2*b^9*c^5*d*f*h^2 - 2304*a^2*b^8*c^6*e*f^2*g - 3870720*a^5*b^2*c^9*d*f*g^2 - 1935360*a^4*b^4*c^8*d*f*g^2 - 1658880*a^4*b^3*c^9*d*e^2*h + 725760*a^3*b^6*c^7*d*f*g^2 + 17418240*a^3*b^4*c^9*d^2*e*g - 124416*a^3*b^5*c^8*d*e^2*h - 96768*a^2*b^8*c^6*d*f*g^2 + 41472*a^2*b^7*c^7*d*e^2*h - 3919104*a^2*b^6*c^8*d^2*e*g - 7741440*a^4*b^2*c^10*d*e^2*f + 2903040*a^3*b^4*c^9*d*e^2*f - 387072*a^2*b^6*c^8*d*e^2*f - 20160*a^8*b^7*c*l^2*m^2 - 1648128*a^10*b^3*c^3*k*m^3 - 898560*a^9*b^3*c^4*k^3*m - 354240*a^9*b^5*c^2*k*m^3 - 354240*a^8*b^5*c^3*k^3*m - 21600*a^7*b^7*c^2*k^3*m - 13950*a^7*b^8*c*k^2*m^2 + 430080*a^10*b*c^5*j^2*m^2 - 1984*a^6*b^9*c*j^2*m^2 - 884736*a^9*b^3*c^4*j*l^3 - 589824*a^8*b^3*c^5*j^3*l - 442368*a^8*b^5*c^3*j*l^3 - 294912*a^7*b^5*c^4*j^3*l - 49152*a^6*b^7*c^3*j^3*l + 1359360*a^10*b^2*c^4*h*m^3 + 1173120*a^9*b^4*c^3*h*m^3 + 743040*a^7*b^4*c^5*h^3*m + 622080*a^8*b^2*c^6*h^3*m + 184320*a^9*b*c^6*j^2*k^2 + 107136*a^6*b^6*c^4*h^3*m - 32640*a^8*b^6*c^2*h*m^3 + 540*a^5*b^8*c^3*h^3*m - 270*a^4*b^10*c^2*h^3*m - 180*a^5*b^10*c*h^2*m^2 - 2293760*a^9*b^3*c^4*f*m^3 - 2293760*a^6*b^3*c^7*f^3*m + 1327104*a^8*b^4*c^4*g*l^3 + 1327104*a^6*b^4*c^6*g^3*l - 622080*a^8*b^3*c^5*h*k^3 - 622080*a^7*b^3*c^6*h^3*k - 326592*a^7*b^5*c^4*h*k^3 - 326592*a^6*b^5*c^5*h^3*k - 199360*a^8*b^5*c^3*f*m^3 - 199360*a^5*b^5*c^6*f^3*m + 61920*a^7*b^7*c^2*f*m^3 + 61920*a^4*b^7*c^5*f^3*m - 38880*a^6*b^7*c^3*h*k^3 - 38880*a^5*b^7*c^4*h^3*k - 3682*a^3*b^9*c^4*f^3*m - 810*a^5*b^9*c^2*h*k^3 - 810*a^4*b^9*c^3*h^3*k - 70*a^3*b^12*c*f^2*m^2 + 70*a^2*b^11*c^3*f^3*m + 3870720*a^8*b*c^7*e^2*m^2 + 184320*a^8*b*c^7*h^2*j^2 - 14152320*a^4*b^4*c^8*d^3*m + 10644480*a^5*b^2*c^9*d^3*m + 5483520*a^9*b^2*c^5*d*m^3 + 4269888*a^3*b^6*c^7*d^3*m - 2654208*a^8*b^3*c^5*e*l^3 + 1359360*a^6*b^2*c^8*f^3*k + 1330560*a^8*b^4*c^4*d*m^3 + 1173120*a^5*b^4*c^7*f^3*k - 884736*a^6*b^3*c^7*g^3*j - 826560*a^7*b^6*c^3*d*m^3 + 743040*a^7*b^4*c^5*f*k^3 + 622080*a^8*b^2*c^6*f*k^3 - 607068*a^2*b^8*c^6*d^3*m - 589824*a^7*b^3*c^6*g*j^3 - 442368*a^5*b^5*c^6*g^3*j - 294912*a^6*b^5*c^5*g*j^3 + 145188*a^6*b^8*c^2*d*m^3 + 107136*a^6*b^6*c^4*f*k^3 - 49152*a^5*b^7*c^4*g*j^3 -

32640*a^4*b^6*c^6*f^3*k - 5796*a^3*b^8*c^5*f^3*k + 540*a^5*b^8*c^3*f*k^3 -
270*a^4*b^10*c^2*f*k^3 + 210*a^2*b^10*c^4*f^3*k + 19077120*a^4*b^3*c^9*d^3
*k + 1658880*a^7*b*c^8*e^2*k^2 + 430080*a^7*b*c^8*f^2*j^2 + 3538944*a^5*b^2
*c^9*e^3*j - 2488320*a^7*b^3*c^6*d*k^3 - 2379456*a^3*b^5*c^8*d^3*k + 117964
8*a^7*b^2*c^7*e*j^3 + 589824*a^6*b^4*c^6*e*j^3 + 98304*a^5*b^6*c^5*e*j^3 -
95904*a^2*b^7*c^7*d^3*k - 57024*a^6*b^5*c^5*d*k^3 + 49248*a^5*b^7*c^4*d*k^3
- 4050*a^4*b^9*c^3*d*k^3 - 810*a^3*b^11*c^2*d*k^3 - 486*a*b^12*c^3*d^2*k^2
+ 3870720*a^6*b*c^9*d^2*j^2 - 1648128*a^5*b^3*c^8*f^3*h - 898560*a^6*b^3*c
^7*f*h^3 - 354240*a^5*b^5*c^6*f*h^3 - 354240*a^4*b^5*c^7*f^3*h + 43680*a^3*
b^7*c^6*f^3*h - 21600*a^4*b^7*c^5*f*h^3 - 9792*a*b^11*c^4*d^2*j^2 + 1350*a^
3*b^9*c^4*f*h^3 - 1050*a^2*b^9*c^5*f^3*h + 1658880*a^6*b*c^9*e^2*h^2 + 1654
7328*a^4*b^2*c^10*d^3*h - 12306816*a^3*b^4*c^9*d^3*h + 37310976*a^3*b^3*c^1
0*d^3*f + 3037824*a^2*b^6*c^8*d^3*h - 2654208*a^5*b^3*c^8*e*g^3 + 1949184*a
^6*b^2*c^8*d*h^3 + 1296000*a^5*b^4*c^7*d*h^3 - 155520*a^4*b^6*c^6*d*h^3 - 4
0500*a*b^10*c^5*d^2*h^2 - 8100*a^3*b^8*c^5*d*h^3 + 4050*a^2*b^10*c^4*d*h^3
+ 3870720*a^5*b*c^10*e^2*f^2 + 34836480*a^4*b*c^11*d^2*e^2 - 108864*a*b^9*c
^6*d^2*g^2 - 8068032*a^2*b^5*c^9*d^3*f - 5623296*a^4*b^3*c^9*d*f^3 + 173779
2*a^3*b^5*c^8*d*f^3 - 260190*a*b^8*c^7*d^2*f^2 - 211680*a^2*b^7*c^7*d*f^3 -
435456*a*b^7*c^8*d^2*e^2 - 245760*a^10*c^6*j^2*k*m - 384*a^6*b^10*j*l*m^2
+ 138240*a^10*c^6*h*k^2*m - 90*a^5*b^11*h*k*m^2 + 384000*a^10*c^6*f*k*m^2 -
2211840*a^8*c^8*e^2*k*m - 409600*a^9*c^7*f*j^2*m - 147456*a^9*c^7*h*j^2*k
- 30*a^4*b^12*f*k*m^2 + 967680*a^9*c^7*d*k^2*m + 384000*a^8*c^8*f^2*h*m - 9
0*a^3*b^13*d*k*m^2 + 20321280*a^7*c^9*d^2*h*m - 883200*a^11*b*c^4*k*m^3 - 3
17952*a^10*b*c^5*k^3*m + 43680*a^8*b^7*c*k*m^3 + 1350*a^6*b^9*c*k^3*m - 270
*b^14*c^2*d^2*h*m + 6*a^3*b^13*f*h*m^2 + 4838400*a^9*c^7*d*h*m^2 + 2903040*
a^8*c^8*d*h^2*m - 1032192*a^8*c^8*d*j^2*k + 138240*a^8*c^8*f*h^2*k - 368640
0*a^7*c^9*e^2*f*m - 1327104*a^7*c^9*e^2*h*k - 393216*a^9*b*c^6*j^3*l - 2457
60*a^8*c^8*f*h*j^2 - 810*b^13*c^3*d^2*h*k + 630*b^13*c^3*d^2*f*m + 18*a^2*b
^14*d*h*m^2 + 2688000*a^7*c^9*d*f^2*m + 580608*a^8*c^8*d*h*k^2 - 5796*a^7*b
^8*c*h*m^3 - 3456*b^12*c^4*d^2*g*j + 1890*b^12*c^4*d^2*f*k + 6773760*a^6*c^
10*d^2*f*k - 1344000*a^10*b*c^5*f*m^3 - 1344000*a^7*b*c^8*f^3*m - 207360*a^
9*b*c^6*h*k^3 - 207360*a^8*b*c^7*h^3*k - 3682*a^6*b^9*c*f*m^3 - 9289728*a^6
*c^10*d*e^2*k - 1720320*a^7*c^9*d*f*j^2 - 50803200*a^5*b*c^10*d^3*k + 6912*
b^11*c^5*d^2*e*j - 10616832*a^6*b*c^9*e^3*l - 2211840*a^6*c^10*e^2*f*h - 39
3216*a^8*b*c^7*g*j^3 + 43416*a*b^10*c^5*d^3*m - 9576*a^5*b^10*c*d*m^3 - 945
0*b^11*c^5*d^2*f*h - 504*a*b^14*c*d^2*m^2 + 1612800*a^6*c^10*d*f^2*h - 1036
800*a^8*b*c^7*d*k^3 + 45198*a*b^9*c^6*d^3*k - 20736*b^10*c^6*d^2*e*g - 7518
8736*a^4*b*c^11*d^3*f - 883200*a^6*b*c^9*f^3*h - 317952*a^7*b*c^8*f*h^3 - 1
5482880*a^5*c^11*d*e^2*f - 10616832*a^5*b*c^10*e^3*g - 345060*a*b^8*c^7*d^3
*h - 4262400*a^5*b*c^10*d*f^3 + 852768*a*b^7*c^8*d^3*f + 7350*a*b^9*c^6*d*f
^3 + 967680*a^10*b^3*c^3*l^2*m^2 + 161280*a^9*b^5*c^2*l^2*m^2 + 1684224*a^1
0*b^2*c^4*k^2*m^2 + 1264320*a^9*b^4*c^3*k^2*m^2 + 126720*a^8*b^6*c^2*k^2*m^
2 + 501760*a^9*b^3*c^4*j^2*m^2 + 414720*a^9*b^3*c^4*k^2*l^2 + 207360*a^8*b^
5*c^3*k^2*l^2 + 170240*a^8*b^5*c^3*j^2*m^2 + 9216*a^7*b^7*c^2*j^2*m^2 + 518
4*a^7*b^7*c^2*k^2*l^2 + 884736*a^9*b^2*c^5*j^2*l^2 + 884736*a^8*b^4*c^4*j^2

*l^2 + 221184*a^7*b^6*c^3*j^2*l^2 + 1419840*a^8*b^4*c^4*h^2*m^2 + 1387008*a
^9*b^2*c^5*h^2*m^2 + 276480*a^8*b^3*c^5*j^2*k^2 + 140544*a^7*b^5*c^4*j^2*k^
2 + 84960*a^7*b^6*c^3*h^2*m^2 + 25344*a^6*b^7*c^3*j^2*k^2 - 8010*a^6*b^8*c^
2*h^2*m^2 + 576*a^5*b^9*c^2*j^2*k^2 + 967680*a^8*b^3*c^5*g^2*m^2 + 414720*a
^8*b^3*c^5*h^2*l^2 + 207360*a^7*b^5*c^4*h^2*l^2 + 161280*a^7*b^5*c^4*g^2*m^
2 - 20160*a^6*b^7*c^3*g^2*m^2 + 5184*a^6*b^7*c^3*h^2*l^2 + 576*a^5*b^9*c^2*
g^2*m^2 + 3808000*a^8*b^2*c^6*f^2*m^2 + 1990656*a^7*b^4*c^5*g^2*l^2 + 16437
12*a^7*b^4*c^5*f^2*m^2 + 803520*a^7*b^4*c^5*h^2*k^2 + 725760*a^8*b^2*c^6*h^
2*k^2 + 207360*a^6*b^6*c^4*h^2*k^2 - 125440*a^6*b^6*c^4*f^2*m^2 - 13790*a^5
*b^8*c^3*f^2*m^2 + 10530*a^5*b^8*c^3*h^2*k^2 + 1785*a^4*b^10*c^2*f^2*m^2 +
81*a^4*b^10*c^2*h^2*k^2 + 18427392*a^7*b^2*c^7*d^2*m^2 + 967680*a^7*b^3*c^6
*f^2*l^2 + 645120*a^7*b^3*c^6*e^2*m^2 + 414720*a^7*b^3*c^6*g^2*k^2 + 276480
*a^7*b^3*c^6*h^2*j^2 + 207360*a^6*b^5*c^5*g^2*k^2 + 161280*a^6*b^5*c^5*f^2*
l^2 + 140544*a^6*b^5*c^5*h^2*j^2 - 80640*a^6*b^5*c^5*e^2*m^2 + 25344*a^5*b^
7*c^4*h^2*j^2 - 20160*a^5*b^7*c^4*f^2*l^2 + 5184*a^5*b^7*c^4*g^2*k^2 + 2304
*a^5*b^7*c^4*e^2*m^2 + 576*a^4*b^9*c^3*h^2*j^2 + 576*a^4*b^9*c^3*f^2*l^2 +
7962624*a^7*b^2*c^7*e^2*l^2 - 4148928*a^6*b^4*c^6*d^2*m^2 + 1419840*a^6*b^4
*c^6*f^2*k^2 + 1387008*a^7*b^2*c^7*f^2*k^2 - 1183392*a^5*b^6*c^5*d^2*m^2 +
884736*a^7*b^2*c^7*g^2*j^2 + 884736*a^6*b^4*c^6*g^2*j^2 + 645750*a^4*b^8*c^
4*d^2*m^2 + 221184*a^5*b^6*c^5*g^2*j^2 - 115920*a^3*b^10*c^3*d^2*m^2 + 8496
0*a^5*b^6*c^5*f^2*k^2 + 10836*a^2*b^12*c^2*d^2*m^2 - 8010*a^4*b^8*c^4*f^2*k
^2 - 180*a^3*b^10*c^3*f^2*k^2 + 9*a^2*b^12*c^2*f^2*k^2 + 8709120*a^6*b^3*c^
7*d^2*l^2 - 4354560*a^5*b^5*c^6*d^2*l^2 + 979776*a^4*b^7*c^5*d^2*l^2 + 8294
40*a^6*b^3*c^7*e^2*k^2 + 17480448*a^6*b^2*c^8*d^2*k^2 + 501760*a^6*b^3*c^7*
f^2*j^2 + 170240*a^5*b^5*c^6*f^2*j^2 - 108864*a^3*b^9*c^4*d^2*l^2 + 20736*a
^5*b^5*c^6*e^2*k^2 + 9216*a^4*b^7*c^5*f^2*j^2 + 5184*a^2*b^11*c^3*d^2*l^2 -
1984*a^3*b^9*c^4*f^2*j^2 + 64*a^2*b^11*c^3*f^2*j^2 + 3538944*a^6*b^2*c^8*e
^2*j^2 - 3302208*a^5*b^4*c^7*d^2*k^2 + 884736*a^5*b^4*c^7*e^2*j^2 + 414720*
a^6*b^3*c^7*g^2*h^2 + 207360*a^5*b^5*c^6*g^2*h^2 - 103680*a^4*b^6*c^6*d^2*k
^2 + 101250*a^3*b^8*c^5*d^2*k^2 - 5751*a^2*b^10*c^4*d^2*k^2 + 5184*a^4*b^7*
c^5*g^2*h^2 + 1935360*a^5*b^3*c^8*d^2*j^2 + 1684224*a^6*b^2*c^8*f^2*h^2 + 1
264320*a^5*b^4*c^7*f^2*h^2 - 532224*a^4*b^5*c^7*d^2*j^2 + 126720*a^4*b^6*c^
6*f^2*h^2 - 96768*a^3*b^7*c^6*d^2*j^2 + 62784*a^2*b^9*c^5*d^2*j^2 - 13950*a
^3*b^8*c^5*f^2*h^2 + 225*a^2*b^10*c^4*f^2*h^2 + 967680*a^5*b^3*c^8*f^2*g^2
+ 829440*a^5*b^3*c^8*e^2*h^2 + 161280*a^4*b^5*c^7*f^2*g^2 + 20736*a^4*b^5*c
^7*e^2*h^2 - 20160*a^3*b^7*c^6*f^2*g^2 + 576*a^2*b^9*c^5*f^2*g^2 + 11487744
*a^5*b^2*c^9*d^2*h^2 + 7962624*a^5*b^2*c^9*e^2*g^2 + 35525376*a^4*b^2*c^10*
d^2*f^2 - 1412640*a^3*b^6*c^7*d^2*h^2 + 461376*a^4*b^4*c^8*d^2*h^2 + 375030
*a^2*b^8*c^6*d^2*h^2 + 8709120*a^4*b^3*c^9*d^2*g^2 - 4354560*a^3*b^5*c^8*d^
2*g^2 + 979776*a^2*b^7*c^7*d^2*g^2 + 645120*a^4*b^3*c^9*e^2*f^2 - 80640*a^3
*b^5*c^8*e^2*f^2 + 2304*a^2*b^7*c^7*e^2*f^2 - 15269184*a^3*b^4*c^9*d^2*f^2
+ 2870784*a^2*b^6*c^8*d^2*f^2 - 17418240*a^3*b^3*c^10*d^2*e^2 + 3919104*a^2
*b^5*c^9*d^2*e^2 + 54*b^15*c*d^2*k*m + 6*a*b^15*d*f*m^2 + 115200*a^11*c^5*k
^2*m^2 + 576*a^7*b^9*l^2*m^2 + 225*a^6*b^10*k^2*m^2 + 64*a^5*b^11*j^2*m^2 +
345600*a^10*c^6*h^2*m^2 + 9*a^4*b^12*h^2*m^2 + 320000*a^9*c^7*f^2*m^2 + 41

472*a^9*c^7*h^2*k^2 + 16934400*a^8*c^8*d^2*m^2 + 345600*a^8*c^8*f^2*k^2 + 8
1*b^14*c^2*d^2*k^2 + 3538944*a^7*c^9*e^2*j^2 + 2032128*a^7*c^9*d^2*k^2 + 49
2800*a^11*b^2*c^3*m^4 + 351456*a^10*b^4*c^2*m^4 + 576*b^13*c^3*d^2*j^2 + 33
1776*a^9*b^4*c^3*l^4 + 115200*a^7*c^9*f^2*h^2 + 142560*a^8*b^4*c^4*k^4 + 10
3680*a^9*b^2*c^5*k^4 + 32400*a^7*b^6*c^3*k^4 + 2025*b^12*c^4*d^2*h^2 + 2025
*a^6*b^8*c^2*k^4 + 6096384*a^6*c^10*d^2*h^2 + 131072*a^8*b^2*c^6*j^4 + 9830
4*a^7*b^4*c^5*j^4 + 32768*a^6*b^6*c^4*j^4 + 5184*b^11*c^5*d^2*g^2 + 4096*a^
5*b^8*c^3*j^4 + 11025*b^10*c^6*d^2*f^2 + 5644800*a^5*c^11*d^2*f^2 + 142560*
a^6*b^4*c^6*h^4 + 103680*a^7*b^2*c^7*h^4 + 32400*a^5*b^6*c^5*h^4 + 20736*b^
9*c^7*d^2*e^2 + 2025*a^4*b^8*c^4*h^4 + 331776*a^5*b^4*c^7*g^4 + 492800*a^5*
b^2*c^9*f^4 + 351456*a^4*b^4*c^8*f^4 - 43120*a^3*b^6*c^7*f^4 + 1225*a^2*b^8
*c^6*f^4 - 27433728*a^3*b^2*c^11*d^4 + 6446304*a^2*b^4*c^10*d^4 - 1050*a^7*
b^9*k*m^3 + 384000*a^11*c^5*h*m^3 + 138240*a^9*c^7*h^3*m + 210*a^6*b^10*h*m
^3 + 47416320*a^6*c^10*d^3*m - 1134*b^12*c^4*d^3*m + 70*a^5*b^11*f*m^3 + 26
88000*a^10*c^6*d*m^3 + 384000*a^7*c^9*f^3*k + 138240*a^9*c^7*f*k^3 - 3402*b
^11*c^5*d^3*k + 210*a^4*b^12*d*m^3 + 7077888*a^6*c^10*e^3*j + 786432*a^8*c^
8*e*j^3 - 43120*a^9*b^6*c*m^4 + 28449792*a^5*c^11*d^3*h + 17010*b^10*c^6*d^
3*h + 580608*a^7*c^9*d*h^3 - 39690*b^9*c^7*d^3*f - 734832*a*b^6*c^9*d^4 + 9
*b^16*d^2*m^2 + 160000*a^12*c^4*m^4 + 1225*a^8*b^8*m^4 + 20736*a^10*c^6*k^4
 + 65536*a^9*c^7*j^4 + 20736*a^8*c^8*h^4 + 49787136*a^4*c^12*d^4 + 160000*a
^6*c^10*f^4 + 5308416*a^5*c^11*e^4 + 35721*b^8*c^8*d^4 + a^2*b^14*f^2*m^2,
z, k1)*((768*a^2*b^14*c^3*d - 3145728*a^10*c^9*h - 5242880*a^11*c^8*m - 220
20096*a^9*c^10*d - 22272*a^3*b^12*c^4*d + 282624*a^4*b^10*c^5*d - 2027520*a
^5*b^8*c^6*d + 8847360*a^6*b^6*c^7*d - 23396352*a^7*b^4*c^8*d + 34603008*a^
8*b^2*c^9*d + 256*a^3*b^13*c^3*f - 9216*a^4*b^11*c^4*f + 122880*a^5*b^9*c^5
*f - 819200*a^6*b^7*c^6*f + 2949120*a^7*b^5*c^7*f - 5505024*a^8*b^3*c^8*f +
 768*a^4*b^12*c^3*h - 12288*a^5*b^10*c^4*h + 61440*a^6*b^8*c^5*h - 983040*a
^8*b^4*c^7*h + 3145728*a^9*b^2*c^8*h - 3072*a^5*b^11*c^3*k + 61440*a^6*b^9*
c^4*k - 491520*a^7*b^7*c^5*k + 1966080*a^8*b^5*c^6*k - 3932160*a^9*b^3*c^7*
k + 256*a^5*b^12*c^2*m - 61440*a^7*b^8*c^4*m + 655360*a^8*b^6*c^5*m - 29491
20*a^9*b^4*c^6*m + 6291456*a^10*b^2*c^7*m + 4194304*a^9*b*c^9*f + 3145728*a
^10*b*c^8*k)/(512*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 + 240*a^6*b
^8*c^3 - 1280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6)) + (x*(157
2864*a^9*c^10*e + 524288*a^10*c^9*j - 1536*a^4*b^10*c^5*e + 30720*a^5*b^8*c
^6*e - 245760*a^6*b^6*c^7*e + 983040*a^7*b^4*c^8*e - 1966080*a^8*b^2*c^9*e
+ 768*a^4*b^11*c^4*g - 15360*a^5*b^9*c^5*g + 122880*a^6*b^7*c^6*g - 491520*
a^7*b^5*c^7*g + 983040*a^8*b^3*c^8*g - 256*a^4*b^12*c^3*j + 4608*a^5*b^10*c
^4*j - 30720*a^6*b^8*c^5*j + 81920*a^7*b^6*c^6*j - 393216*a^9*b^2*c^8*j + 7
68*a^5*b^11*c^3*l - 15360*a^6*b^9*c^4*l + 122880*a^7*b^7*c^5*l - 491520*a^8
*b^5*c^6*l + 983040*a^9*b^3*c^7*l - 786432*a^9*b*c^9*g - 786432*a^10*b*c^8*
l))/(64*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 + 240*a^6*b^8*c^3 - 1
280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6)) + (root(56371445760
*a^11*b^8*c^9*z^4 - 503316480*a^8*b^14*c^6*z^4 + 47185920*a^7*b^16*c^5*z^4
- 2621440*a^6*b^18*c^4*z^4 + 65536*a^5*b^20*c^3*z^4 - 171798691840*a^14*b^2
*c^12*z^4 + 193273528320*a^13*b^4*c^11*z^4 - 128849018880*a^12*b^6*c^10*z^4

 - 16911433728*a^10*b^10*c^8*z^4 + 3523215360*a^9*b^12*c^7*z^4 + 6871947673
6*a^15*c^13*z^4 + 1536*a^5*b^16*c*k*m*z^2 + 1536*a*b^18*c^3*d*f*z^2 - 25716
32640*a^9*b^5*c^8*d*m*z^2 + 2548039680*a^9*b^3*c^10*d*h*z^2 + 1509949440*a^
10*b^3*c^9*e*l*z^2 + 1509949440*a^9*b^3*c^10*e*g*z^2 - 1401421824*a^8*b^5*c
^9*d*h*z^2 - 1321205760*a^9*b^2*c^11*d*f*z^2 - 2793406464*a^11*b*c^10*d*m*z
^2 + 890634240*a^8*b^7*c^7*d*m*z^2 - 754974720*a^10*b^4*c^8*g*l*z^2 - 75497
4720*a^9*b^5*c^8*e*l*z^2 + 719585280*a^8*b^6*c^8*d*k*z^2 - 707788800*a^9*b^
4*c^9*d*k*z^2 - 754974720*a^8*b^5*c^9*e*g*z^2 + 603979776*a^11*b^2*c^9*g*l*
z^2 - 581959680*a^10*b^4*c^8*f*m*z^2 + 732168192*a^7*b^6*c^9*d*f*z^2 + 5347
73760*a^11*b^3*c^8*h*m*z^2 - 456130560*a^11*b^4*c^7*k*m*z^2 - 603979776*a^1
0*b^2*c^10*e*j*z^2 + 534773760*a^10*b^3*c^9*f*k*z^2 + 384040960*a^9*b^6*c^7
*f*m*z^2 + 377487360*a^9*b^6*c^7*g*l*z^2 - 456130560*a^9*b^4*c^9*f*h*z^2 +
301989888*a^11*b^3*c^8*j*l*z^2 - 415236096*a^10*b^2*c^10*d*k*z^2 + 25401753
6*a^10*b^6*c^6*k*m*z^2 - 330301440*a^10*b^4*c^8*h*k*z^2 + 390463488*a^7*b^7
*c^8*d*h*z^2 + 188743680*a^12*b^2*c^8*k*m*z^2 + 301989888*a^10*b^3*c^9*g*j*
z^2 - 297861120*a^7*b^8*c^7*d*k*z^2 - 366280704*a^6*b^8*c^8*d*f*z^2 + 18874
3680*a^11*b^2*c^9*h*k*z^2 - 330301440*a^8*b^4*c^10*d*f*z^2 + 254017536*a^8*
b^6*c^8*f*h*z^2 - 1887436800*a^10*b*c^11*d*h*z^2 + 188743680*a^8*b^7*c^7*e*
l*z^2 + 153354240*a^9*b^6*c^7*h*k*z^2 - 185303040*a^7*b^9*c^6*d*m*z^2 - 117
964800*a^10*b^5*c^7*h*m*z^2 - 61931520*a^9*b^8*c^5*k*m*z^2 + 121634816*a^11
*b^2*c^9*f*m*z^2 - 115671040*a^8*b^8*c^6*f*m*z^2 - 62914560*a^9*b^7*c^6*j*l
*z^2 + 188743680*a^10*b^2*c^10*f*h*z^2 - 94371840*a^8*b^8*c^6*g*l*z^2 + 614
4000*a^8*b^10*c^4*k*m*z^2 - 117964800*a^9*b^5*c^8*f*k*z^2 + 61440*a^7*b^12*
c^3*k*m*z^2 - 46080*a^6*b^14*c^2*k*m*z^2 + 23592960*a^8*b^9*c^5*j*l*z^2 + 1
88743680*a^7*b^7*c^8*e*g*z^2 - 37355520*a^9*b^7*c^6*h*m*z^2 + 125829120*a^8
*b^6*c^8*e*j*z^2 + 23101440*a^8*b^9*c^5*h*m*z^2 - 3538944*a^7*b^11*c^4*j*l*
z^2 + 196608*a^6*b^13*c^3*j*l*z^2 - 4349952*a^7*b^11*c^4*h*m*z^2 + 337920*a
^6*b^13*c^3*h*m*z^2 - 7680*a^5*b^15*c^2*h*m*z^2 - 62914560*a^8*b^7*c^7*g*j*
z^2 - 26542080*a^8*b^8*c^6*h*k*z^2 + 17940480*a^7*b^10*c^5*f*m*z^2 + 117964
80*a^7*b^10*c^5*g*l*z^2 - 37355520*a^8*b^7*c^7*f*k*z^2 - 1347584*a^6*b^12*c
^4*f*m*z^2 + 68272128*a^6*b^10*c^6*d*k*z^2 - 589824*a^6*b^12*c^4*g*l*z^2 +
552960*a^6*b^12*c^4*h*k*z^2 - 147456*a^7*b^10*c^5*h*k*z^2 - 46080*a^5*b^14*
c^3*h*k*z^2 + 35840*a^5*b^14*c^3*f*m*z^2 + 23592960*a^7*b^9*c^6*g*j*z^2 - 2
3592960*a^7*b^9*c^6*e*l*z^2 + 23371776*a^6*b^11*c^5*d*m*z^2 + 23101440*a^7*
b^9*c^6*f*k*z^2 - 47185920*a^7*b^8*c^7*e*j*z^2 - 61931520*a^7*b^8*c^7*f*h*z
^2 - 4349952*a^6*b^11*c^5*f*k*z^2 - 3538944*a^6*b^11*c^5*g*j*z^2 - 1677312*
a^5*b^13*c^4*d*m*z^2 + 1179648*a^6*b^11*c^5*e*l*z^2 + 337920*a^5*b^13*c^4*f
*k*z^2 + 196608*a^5*b^13*c^4*g*j*z^2 + 53760*a^4*b^15*c^3*d*m*z^2 - 7680*a^
4*b^15*c^3*f*k*z^2 + 96583680*a^5*b^10*c^7*d*f*z^2 - 9179136*a^5*b^12*c^5*d
*k*z^2 + 7077888*a^6*b^10*c^6*e*j*z^2 - 51609600*a^6*b^9*c^7*d*h*z^2 + 6912
00*a^4*b^14*c^4*d*k*z^2 - 393216*a^5*b^12*c^5*e*j*z^2 - 23040*a^3*b^16*c^3*
d*k*z^2 + 6144000*a^6*b^10*c^6*f*h*z^2 + 61440*a^5*b^12*c^5*f*h*z^2 - 46080
*a^4*b^14*c^4*f*h*z^2 + 1536*a^3*b^16*c^3*f*h*z^2 - 23592960*a^6*b^9*c^7*e*
g*z^2 + 1179648*a^5*b^11*c^6*e*g*z^2 + 829440*a^4*b^13*c^5*d*h*z^2 + 368640
*a^5*b^11*c^6*d*h*z^2 - 105984*a^3*b^15*c^4*d*h*z^2 + 4608*a^2*b^17*c^3*d*h

*z^2 - 15175680*a^4*b^12*c^6*d*f*z^2 + 1428480*a^3*b^14*c^5*d*f*z^2 - 73728
*a^2*b^16*c^4*d*f*z^2 + 4108320768*a^10*b^3*c^9*d*m*z^2 - 1207959552*a^11*b
*c^10*e*l*z^2 - 1207959552*a^10*b*c^11*e*g*z^2 - 578813952*a^12*b*c^9*h*m*z
^2 - 578813952*a^11*b*c^10*f*k*z^2 - 402653184*a^12*b*c^9*j*l*z^2 - 4026531
84*a^11*b*c^10*g*j*z^2 - 440401920*a^10*b*c^11*f^2*z^2 - 188743680*a^12*b*c
^9*k^2*z^2 - 188743680*a^11*b*c^10*h^2*z^2 + 1761607680*a^10*c^12*d*f*z^2 -
14080*a^6*b^15*c*m^2*z^2 - 94464*a*b^17*c^4*d^2*z^2 + 6936330240*a^8*b^3*c
^11*d^2*z^2 + 2464874496*a^6*b^7*c^9*d^2*z^2 - 3963617280*a^9*b*c^12*d^2*z^
2 + 1056964608*a^11*c^11*d*k*z^2 + 805306368*a^11*c^11*e*j*z^2 + 419430400*
a^12*c^10*f*m*z^2 + 251658240*a^13*c^9*k*m*z^2 - 1509949440*a^9*b^2*c^11*e^
2*z^2 + 251658240*a^11*c^11*f*h*z^2 + 150994944*a^12*c^10*h*k*z^2 - 5400428
544*a^7*b^5*c^10*d^2*z^2 + 754974720*a^8*b^4*c^10*e^2*z^2 - 730054656*a^5*b
^9*c^8*d^2*z^2 + 477102080*a^12*b^3*c^7*m^2*z^2 - 377487360*a^11*b^4*c^7*l^
2*z^2 + 477102080*a^9*b^3*c^10*f^2*z^2 + 301989888*a^12*b^2*c^8*l^2*z^2 - 3
77487360*a^9*b^4*c^9*g^2*z^2 + 301989888*a^10*b^2*c^10*g^2*z^2 - 174325760*
a^11*b^5*c^6*m^2*z^2 + 188743680*a^10*b^6*c^6*l^2*z^2 + 141557760*a^11*b^3*
c^8*k^2*z^2 + 188743680*a^8*b^6*c^8*g^2*z^2 + 141557760*a^10*b^3*c^9*h^2*z^
2 - 174325760*a^8*b^5*c^9*f^2*z^2 - 188743680*a^7*b^6*c^9*e^2*z^2 - 4718592
0*a^9*b^8*c^5*l^2*z^2 + 11206656*a^10*b^7*c^5*m^2*z^2 + 8929280*a^9*b^9*c^4
*m^2*z^2 - 2600960*a^8*b^11*c^3*m^2*z^2 + 291840*a^7*b^13*c^2*m^2*z^2 - 503
31648*a^10*b^4*c^8*j^2*z^2 + 146165760*a^4*b^11*c^7*d^2*z^2 - 26542080*a^9*
b^7*c^6*k^2*z^2 + 5898240*a^8*b^10*c^4*l^2*z^2 - 294912*a^7*b^12*c^3*l^2*z^
2 - 33554432*a^11*b^2*c^9*j^2*z^2 + 9584640*a^8*b^9*c^5*k^2*z^2 + 20971520*
a^9*b^6*c^7*j^2*z^2 - 2359296*a^10*b^5*c^7*k^2*z^2 - 1290240*a^7*b^11*c^4*k
^2*z^2 + 46080*a^6*b^13*c^3*k^2*z^2 + 2304*a^5*b^15*c^2*k^2*z^2 - 2752512*a
^7*b^10*c^5*j^2*z^2 + 2621440*a^8*b^8*c^6*j^2*z^2 + 524288*a^6*b^12*c^4*j^2
*z^2 - 32768*a^5*b^14*c^3*j^2*z^2 - 47185920*a^7*b^8*c^7*g^2*z^2 - 26542080
*a^8*b^7*c^7*h^2*z^2 + 9584640*a^7*b^9*c^6*h^2*z^2 - 2359296*a^9*b^5*c^8*h^
2*z^2 - 1290240*a^6*b^11*c^5*h^2*z^2 + 46080*a^5*b^13*c^4*h^2*z^2 + 2304*a^
4*b^15*c^3*h^2*z^2 + 5898240*a^6*b^10*c^6*g^2*z^2 - 294912*a^5*b^12*c^5*g^2
*z^2 + 11206656*a^7*b^7*c^8*f^2*z^2 + 8929280*a^6*b^9*c^7*f^2*z^2 + 2359296
0*a^6*b^8*c^8*e^2*z^2 - 2600960*a^5*b^11*c^6*f^2*z^2 + 291840*a^4*b^13*c^5*
f^2*z^2 - 14080*a^3*b^15*c^4*f^2*z^2 + 256*a^2*b^17*c^3*f^2*z^2 - 19860480*
a^3*b^13*c^6*d^2*z^2 - 1179648*a^5*b^10*c^7*e^2*z^2 + 1771776*a^2*b^15*c^5*
d^2*z^2 - 440401920*a^13*b*c^8*m^2*z^2 + 1207959552*a^10*c^12*e^2*z^2 + 134
217728*a^12*c^10*j^2*z^2 + 256*a^5*b^17*m^2*z^2 + 2304*b^19*c^3*d^2*z^2 - 2
3592960*a^10*b*c^8*f*k*l*z + 99090432*a^9*b*c^9*d*h*l*z + 9437184*a^10*b*c^
8*e*k*m*z + 23592960*a^10*b*c^8*g*h*m*z + 141557760*a^8*b*c^10*d*e*k*z + 47
185920*a^9*b*c^9*d*j*k*z - 23592960*a^9*b*c^9*f*g*k*z + 169869312*a^7*b*c^1
1*d*e*f*z + 99090432*a^8*b*c^10*d*g*h*z - 3145728*a^9*b*c^9*f*h*j*z + 56623
104*a^8*b*c^10*d*f*j*z + 1536*a*b^15*c^3*d*f*j*z - 9437184*a^8*b*c^10*e*f*h
*z - 4608*a*b^14*c^4*d*f*g*z + 9216*a*b^13*c^5*d*e*f*z + 412876800*a^8*b^2*
c^9*d*e*m*z - 206438400*a^9*b^3*c^7*d*l*m*z + 5898240*a^10*b^4*c^5*k*l*m*z
- 206438400*a^8*b^3*c^8*d*g*m*z - 4718592*a^11*b^2*c^6*k*l*m*z - 2949120*a^
9*b^6*c^4*k*l*m*z + 737280*a^8*b^8*c^3*k*l*m*z - 92160*a^7*b^10*c^2*k*l*m*z

+ 103219200*a^8*b^5*c^6*d*l*m*z - 29491200*a^10*b^3*c^6*h*l*m*z - 20643840
0*a^7*b^4*c^8*d*e*m*z - 2359296*a^10*b^3*c^6*j*k*m*z + 491520*a^8*b^7*c^4*j
*k*m*z - 184320*a^7*b^9*c^3*j*k*m*z + 27648*a^6*b^11*c^2*j*k*m*z + 14745600
*a^9*b^5*c^5*h*l*m*z - 3686400*a^8*b^7*c^4*h*l*m*z + 460800*a^7*b^9*c^3*h*l
*m*z - 23040*a^6*b^11*c^2*h*l*m*z + 88473600*a^8*b^4*c^7*d*k*l*z + 82575360
*a^9*b^2*c^8*d*j*m*z + 11796480*a^10*b^2*c^7*h*j*m*z + 5898240*a^9*b^4*c^6*
g*k*m*z - 4718592*a^10*b^2*c^7*g*k*m*z - 70778880*a^9*b^2*c^8*d*k*l*z - 294
9120*a^8*b^6*c^5*g*k*m*z - 2457600*a^8*b^6*c^5*h*j*m*z + 921600*a^7*b^8*c^4
*h*j*m*z + 737280*a^7*b^8*c^4*g*k*m*z - 138240*a^6*b^10*c^3*h*j*m*z - 92160
*a^6*b^10*c^3*g*k*m*z + 7680*a^5*b^12*c^2*h*j*m*z + 4608*a^5*b^12*c^2*g*k*m
*z + 29491200*a^9*b^3*c^7*f*k*l*z - 176947200*a^7*b^3*c^9*d*e*k*z - 1097072
64*a^8*b^3*c^8*d*h*l*z - 25804800*a^7*b^7*c^5*d*l*m*z + 103219200*a^7*b^5*c
^7*d*g*m*z + 219414528*a^7*b^2*c^10*d*e*h*z - 14745600*a^8*b^5*c^6*f*k*l*z
- 29491200*a^9*b^3*c^7*g*h*m*z - 11796480*a^9*b^3*c^7*e*k*m*z - 44236800*a^
7*b^6*c^6*d*k*l*z + 58982400*a^9*b^2*c^8*e*h*m*z + 5898240*a^8*b^5*c^6*e*k*
m*z + 3686400*a^7*b^7*c^5*f*k*l*z + 3225600*a^6*b^9*c^4*d*l*m*z - 1474560*a
^7*b^7*c^5*e*k*m*z - 460800*a^6*b^9*c^4*f*k*l*z + 184320*a^6*b^9*c^4*e*k*m*
z - 161280*a^5*b^11*c^3*d*l*m*z + 23040*a^5*b^11*c^3*f*k*l*z - 9216*a^5*b^1
1*c^3*e*k*m*z + 14745600*a^8*b^5*c^6*g*h*m*z + 110886912*a^7*b^4*c^8*d*f*l*
z - 3686400*a^7*b^7*c^5*g*h*m*z - 221773824*a^6*b^3*c^10*d*e*f*z + 460800*a
^6*b^9*c^4*g*h*m*z - 17203200*a^7*b^6*c^6*d*j*m*z - 23040*a^5*b^11*c^3*g*h*
m*z - 29491200*a^8*b^4*c^7*e*h*m*z - 11796480*a^9*b^2*c^8*f*j*k*z + 1105920
0*a^6*b^8*c^5*d*k*l*z + 6451200*a^6*b^8*c^5*d*j*m*z + 88473600*a^7*b^4*c^8*
d*g*k*z + 2457600*a^7*b^6*c^6*f*j*k*z - 35389440*a^8*b^3*c^8*d*j*k*z - 1382
400*a^5*b^10*c^4*d*k*l*z - 84934656*a^8*b^2*c^9*d*f*l*z - 967680*a^5*b^10*c
^4*d*j*m*z - 921600*a^6*b^8*c^5*f*j*k*z + 138240*a^5*b^10*c^4*f*j*k*z + 691
20*a^4*b^12*c^3*d*k*l*z + 53760*a^4*b^12*c^3*d*j*m*z - 7680*a^4*b^12*c^3*f*
j*k*z + 44236800*a^7*b^5*c^7*d*h*l*z + 7372800*a^7*b^6*c^6*e*h*m*z - 589824
0*a^8*b^4*c^7*f*h*l*z + 4718592*a^9*b^2*c^8*f*h*l*z - 70778880*a^8*b^2*c^9*
d*g*k*z + 2949120*a^7*b^6*c^6*f*h*l*z - 921600*a^6*b^8*c^5*e*h*m*z - 737280
*a^6*b^8*c^5*f*h*l*z + 92160*a^5*b^10*c^4*f*h*l*z + 46080*a^5*b^10*c^4*e*h*
m*z - 4608*a^4*b^12*c^3*f*h*l*z + 29491200*a^8*b^3*c^8*f*g*k*z - 109707264*
a^7*b^3*c^9*d*g*h*z - 25804800*a^6*b^7*c^6*d*g*m*z - 58982400*a^8*b^2*c^9*e
*f*k*z - 58982400*a^6*b^6*c^7*d*f*l*z + 7372800*a^6*b^7*c^6*d*j*k*z + 88473
600*a^6*b^5*c^8*d*e*k*z - 2764800*a^5*b^9*c^5*d*j*k*z + 51609600*a^6*b^6*c^
7*d*e*m*z + 414720*a^4*b^11*c^4*d*j*k*z - 23040*a^3*b^13*c^3*d*j*k*z - 1474
5600*a^7*b^5*c^7*f*g*k*z - 44236800*a^6*b^6*c^7*d*g*k*z - 6635520*a^6*b^7*c
^6*d*h*l*z + 40108032*a^8*b^2*c^9*d*h*j*z + 3686400*a^6*b^7*c^6*f*g*k*z + 3
225600*a^5*b^9*c^5*d*g*m*z + 2359296*a^8*b^3*c^8*f*h*j*z - 491520*a^6*b^7*c
^6*f*h*j*z - 460800*a^5*b^9*c^5*f*g*k*z - 276480*a^5*b^9*c^5*d*h*l*z + 1843
20*a^5*b^9*c^5*f*h*j*z + 179712*a^4*b^11*c^4*d*h*l*z - 161280*a^4*b^11*c^4*
d*g*m*z - 27648*a^4*b^11*c^4*f*h*j*z + 23040*a^4*b^11*c^4*f*g*k*z - 13824*a
^3*b^13*c^3*d*h*l*z + 1536*a^3*b^13*c^3*f*h*j*z + 29491200*a^7*b^4*c^8*e*f*
k*z + 110886912*a^6*b^4*c^9*d*f*g*z + 16220160*a^5*b^8*c^6*d*f*l*z - 456130
56*a^7*b^3*c^9*d*f*j*z + 11059200*a^5*b^8*c^6*d*g*k*z - 10321920*a^6*b^6*c^

7*d*h*j*z - 7372800*a^6*b^6*c^7*e*f*k*z + 7077888*a^7*b^4*c^8*d*h*j*z - 645
1200*a^5*b^8*c^6*d*e*m*z - 88473600*a^6*b^4*c^9*d*e*h*z + 2396160*a^5*b^8*c
^6*d*h*j*z - 2396160*a^4*b^10*c^5*d*f*l*z - 1382400*a^4*b^10*c^5*d*g*k*z -
84934656*a^7*b^2*c^10*d*f*g*z + 921600*a^5*b^8*c^6*e*f*k*z + 117964800*a^5*
b^5*c^9*d*e*f*z + 322560*a^4*b^10*c^5*d*e*m*z + 175104*a^3*b^12*c^4*d*f*l*z
 + 69120*a^3*b^12*c^4*d*g*k*z - 50688*a^3*b^12*c^4*d*h*j*z - 46080*a^4*b^10
*c^5*e*f*k*z - 27648*a^4*b^10*c^5*d*h*j*z + 4608*a^2*b^14*c^3*d*h*j*z - 460
8*a^2*b^14*c^3*d*f*l*z + 44236800*a^6*b^5*c^8*d*g*h*z - 5898240*a^7*b^4*c^8
*f*g*h*z - 22118400*a^5*b^7*c^7*d*e*k*z + 4718592*a^8*b^2*c^9*f*g*h*z + 294
9120*a^6*b^6*c^7*f*g*h*z - 737280*a^5*b^8*c^6*f*g*h*z + 92160*a^4*b^10*c^5*
f*g*h*z - 4608*a^3*b^12*c^4*f*g*h*z + 8847360*a^5*b^7*c^7*d*f*j*z - 5898240
0*a^5*b^6*c^8*d*f*g*z - 3809280*a^4*b^9*c^6*d*f*j*z + 2764800*a^4*b^9*c^6*d
*e*k*z + 2359296*a^6*b^5*c^8*d*f*j*z + 681984*a^3*b^11*c^5*d*f*j*z - 138240
*a^3*b^11*c^5*d*e*k*z - 55296*a^2*b^13*c^4*d*f*j*z + 11796480*a^7*b^3*c^9*e
*f*h*z - 6635520*a^5*b^7*c^7*d*g*h*z - 5898240*a^6*b^5*c^8*e*f*h*z + 147456
0*a^5*b^7*c^7*e*f*h*z - 276480*a^4*b^9*c^6*d*g*h*z - 184320*a^4*b^9*c^6*e*f
*h*z + 179712*a^3*b^11*c^5*d*g*h*z - 13824*a^2*b^13*c^4*d*g*h*z + 9216*a^3*
b^11*c^5*e*f*h*z + 16220160*a^4*b^8*c^7*d*f*g*z + 13271040*a^5*b^6*c^8*d*e*
h*z - 2396160*a^3*b^10*c^6*d*f*g*z + 552960*a^4*b^8*c^7*d*e*h*z - 359424*a^
3*b^10*c^6*d*e*h*z + 175104*a^2*b^12*c^5*d*f*g*z + 27648*a^2*b^12*c^5*d*e*h
*z - 32440320*a^4*b^7*c^8*d*e*f*z + 4792320*a^3*b^9*c^7*d*e*f*z - 350208*a^
2*b^11*c^6*d*e*f*z + 165150720*a^10*b*c^8*d*l*m*z + 4608*a^6*b^12*c*k*l*m*z
 + 23592960*a^11*b*c^7*h*l*m*z + 3145728*a^11*b*c^7*j*k*m*z - 1536*a^5*b^13
*c*j*k*m*z + 165150720*a^9*b*c^9*d*g*m*z + 346816512*a^7*b*c^11*d^2*g*z + 1
9660800*a^12*b*c^6*l*m^2*z - 34560*a^7*b^11*c*l*m^2*z - 7077888*a^11*b*c^7*
k^2*l*z + 11008*a^6*b^12*c*j*m^2*z + 19660800*a^11*b*c^7*g*m^2*z + 7077888*
a^10*b*c^8*h^2*l*z + 768*a^5*b^13*c*g*m^2*z - 19660800*a^9*b*c^9*f^2*l*z -
7077888*a^10*b*c^8*g*k^2*z - 6912*a*b^15*c^3*d^2*l*z + 7077888*a^9*b*c^9*g*
h^2*z - 19660800*a^8*b*c^10*f^2*g*z - 66816*a*b^14*c^4*d^2*j*z + 214272*a*b
^13*c^5*d^2*g*z - 428544*a*b^12*c^6*d^2*e*z - 330301440*a^9*c^10*d*e*m*z -
110100480*a^10*c^9*d*j*m*z - 15728640*a^11*c^8*h*j*m*z - 47185920*a^10*c^9*
e*h*m*z - 198180864*a^8*c^11*d*e*h*z + 15728640*a^10*c^9*f*j*k*z - 66060288
*a^9*c^10*d*h*j*z + 47185920*a^9*c^10*e*f*k*z + 1022754816*a^6*b^2*c^11*d^2
*e*z - 642318336*a^5*b^4*c^10*d^2*e*z - 511377408*a^7*b^3*c^9*d^2*l*z - 511
377408*a^6*b^3*c^10*d^2*g*z + 321159168*a^6*b^5*c^8*d^2*l*z + 321159168*a^5
*b^5*c^9*d^2*g*z + 225312768*a^7*b^2*c^10*d^2*j*z - 25362432*a^11*b^3*c^5*l
*m^2*z + 13271040*a^10*b^5*c^4*l*m^2*z - 3563520*a^9*b^7*c^3*l*m^2*z + 5068
80*a^8*b^9*c^2*l*m^2*z + 10354688*a^11*b^2*c^6*j*m^2*z + 8847360*a^10*b^3*c
^6*k^2*l*z - 4423680*a^9*b^5*c^5*k^2*l*z - 2048000*a^9*b^6*c^4*j*m^2*z + 11
05920*a^8*b^7*c^4*k^2*l*z + 849920*a^8*b^8*c^3*j*m^2*z - 393216*a^10*b^4*c^
5*j*m^2*z - 145920*a^7*b^10*c^2*j*m^2*z - 138240*a^7*b^9*c^3*k^2*l*z + 6912
*a^6*b^11*c^2*k^2*l*z - 111697920*a^5*b^7*c^7*d^2*l*z + 223395840*a^4*b^6*c
^9*d^2*e*z - 25362432*a^10*b^3*c^6*g*m^2*z - 3538944*a^10*b^2*c^7*j*k^2*z +
 737280*a^8*b^6*c^5*j*k^2*z + 50724864*a^10*b^2*c^7*e*m^2*z - 276480*a^7*b^
8*c^4*j*k^2*z + 41472*a^6*b^10*c^3*j*k^2*z - 2304*a^5*b^12*c^2*j*k^2*z + 13

271040*a^9*b^5*c^5*g*m^2*z - 8847360*a^9*b^3*c^7*h^2*l*z + 4423680*a^8*b^5*c^6*h^2*l*z - 3563520*a^8*b^7*c^4*g*m^2*z - 1105920*a^7*b^7*c^5*h^2*l*z + 506880*a^7*b^9*c^3*g*m^2*z + 138240*a^6*b^9*c^4*h^2*l*z - 34560*a^6*b^11*c^2*g*m^2*z - 6912*a^5*b^11*c^3*h^2*l*z - 26542080*a^9*b^4*c^6*e*m^2*z + 25362432*a^8*b^3*c^8*f^2*l*z - 13271040*a^7*b^5*c^7*f^2*l*z + 8847360*a^9*b^3*c^7*g*k^2*z + 7127040*a^8*b^6*c^5*e*m^2*z - 4423680*a^8*b^5*c^6*g*k^2*z + 3563520*a^6*b^7*c^6*f^2*l*z + 3538944*a^9*b^2*c^8*h^2*j*z + 1105920*a^7*b^7*c^5*g*k^2*z - 1013760*a^7*b^8*c^4*e*m^2*z - 737280*a^7*b^6*c^6*h^2*j*z - 506880*a^5*b^9*c^5*f^2*l*z + 276480*a^6*b^8*c^5*h^2*j*z - 138240*a^6*b^9*c^4*g*k^2*z + 69120*a^6*b^10*c^3*e*m^2*z - 41472*a^5*b^10*c^4*h^2*j*z + 34560*a^4*b^11*c^4*f^2*l*z + 6912*a^5*b^11*c^3*g*k^2*z + 2304*a^4*b^12*c^3*h^2*j*z - 1536*a^5*b^12*c^2*e*m^2*z - 768*a^3*b^13*c^3*f^2*l*z - 111697920*a^4*b^7*c^8*d^2*g*z + 23362560*a^4*b^9*c^6*d^2*l*z - 17694720*a^9*b^2*c^8*e*k^2*z - 10354688*a^8*b^2*c^9*f^2*j*z - 43646976*a^6*b^4*c^9*d^2*j*z + 8847360*a^8*b^4*c^7*e*k^2*z - 2965248*a^3*b^11*c^5*d^2*l*z - 2211840*a^7*b^6*c^6*e*k^2*z + 2048000*a^6*b^6*c^7*f^2*j*z - 849920*a^5*b^8*c^6*f^2*j*z + 393216*a^7*b^4*c^8*f^2*j*z + 276480*a^6*b^8*c^5*e*k^2*z + 214272*a^2*b^13*c^4*d^2*l*z + 145920*a^4*b^10*c^5*f^2*j*z - 13824*a^5*b^10*c^4*e*k^2*z - 11008*a^3*b^12*c^4*f^2*j*z + 256*a^2*b^14*c^3*f^2*j*z - 32587776*a^5*b^6*c^8*d^2*j*z - 8847360*a^8*b^3*c^8*g*h^2*z + 21657600*a^4*b^8*c^7*d^2*j*z + 4423680*a^7*b^5*c^7*g*h^2*z - 1105920*a^6*b^7*c^6*g*h^2*z + 138240*a^5*b^9*c^5*g*h^2*z - 6912*a^4*b^11*c^4*g*h^2*z + 25362432*a^7*b^3*c^9*f^2*g*z - 5810688*a^3*b^10*c^6*d^2*j*z + 17694720*a^8*b^2*c^9*e*h^2*z + 845568*a^2*b^12*c^5*d^2*j*z - 50724864*a^7*b^2*c^10*e*f^2*z - 13271040*a^6*b^5*c^8*f^2*g*z - 8847360*a^7*b^4*c^8*e*h^2*z + 3563520*a^5*b^7*c^7*f^2*g*z + 2211840*a^6*b^6*c^7*e*h^2*z - 506880*a^4*b^9*c^6*f^2*g*z - 276480*a^5*b^8*c^6*e*h^2*z + 34560*a^3*b^11*c^5*f^2*g*z + 13824*a^4*b^10*c^5*e*h^2*z - 768*a^2*b^13*c^4*f^2*g*z + 26542080*a^6*b^4*c^9*e*f^2*z + 23362560*a^3*b^9*c^7*d^2*g*z - 46725120*a^3*b^8*c^8*d^2*e*z - 7127040*a^5*b^6*c^8*e*f^2*z - 2965248*a^2*b^11*c^6*d^2*g*z + 1013760*a^4*b^8*c^7*e*f^2*z - 69120*a^3*b^10*c^6*e*f^2*z + 1536*a^2*b^12*c^5*e*f^2*z + 5930496*a^2*b^10*c^7*d^2*e*z + 346816512*a^8*b*c^10*d^2*l*z - 693633024*a^7*c^12*d^2*e*z - 231211008*a^8*c^11*d^2*j*z + 768*a^6*b^13*l*m^2*z - 13107200*a^12*c^7*j*m^2*z - 256*a^5*b^14*j*m^2*z + 4718592*a^11*c^8*j*k^2*z - 39321600*a^11*c^8*e*m^2*z - 4718592*a^10*c^9*h^2*j*z + 14155776*a^10*c^9*e*k^2*z + 13107200*a^9*c^10*f^2*j*z + 2304*b^16*c^3*d^2*j*z - 14155776*a^9*c^10*e*h^2*z + 39321600*a^8*c^11*e*f^2*z - 6912*b^15*c^4*d^2*g*z + 13824*b^14*c^5*d^2*e*z + 737280*a^10*b*c^5*j*k*l*m - 2304*a^6*b^9*c*j*k*l*m + 2211840*a^9*b*c^6*e*k*l*m + 1228800*a^9*b*c^6*f*j*l*m + 737280*a^9*b*c^6*g*j*k*m + 442368*a^9*b*c^6*h*j*k*l + 36*a^3*b^12*c*f*h*k*m + 3096576*a^8*b*c^7*d*j*k*l - 12745728*a^8*b*c^7*d*h*k*m + 3686400*a^8*b*c^7*e*f*l*m + 3391488*a^8*b*c^7*e*h*j*m + 2211840*a^8*b*c^7*e*g*k*m + 1327104*a^8*b*c^7*e*h*k*l + 1228800*a^8*b*c^7*f*g*j*m + 737280*a^8*b*c^7*f*h*j*l + 442368*a^8*b*c^7*g*h*j*k + 108*a^2*b^13*c*d*h*k*m + 16367616*a^7*b*c^8*d*e*j*m + 9289728*a^7*b*c^8*d*e*k*l + 5160960*a^7*b*c^8*d*f*j*l + 3391488*a^7*b*c^8*e*f*j*k + 3096576*a^7*b*c^8*d*g*j*k - 19307520*a^7*b*c^8*d*f*h*m + 3686400*a^7*b*c^8*e*f*

g*m + 2211840*a^7*b*c^8*e*f*h*l + 1327104*a^7*b*c^8*e*g*h*k + 737280*a^7*b*
c^8*f*g*h*j - 180*a*b^13*c^2*d*f*h*m - 540*a*b^12*c^3*d*f*h*k + 15482880*a^
6*b*c^9*d*e*f*l + 11059200*a^6*b*c^9*d*e*h*j + 9289728*a^6*b*c^9*d*e*g*k +
5160960*a^6*b*c^9*d*f*g*j - 2304*a*b^11*c^4*d*f*g*j + 2211840*a^6*b*c^9*e*f
*g*h + 4608*a*b^10*c^5*d*e*f*j + 15482880*a^5*b*c^10*d*e*f*g - 13824*a*b^9*
c^6*d*e*f*g + 36*a*b^14*c*d*f*k*m + 1843200*a^9*b^3*c^4*j*k*l*m + 783360*a^
8*b^5*c^3*j*k*l*m + 18432*a^7*b^7*c^2*j*k*l*m - 2211840*a^8*b^4*c^4*g*k*l*m
 - 1695744*a^9*b^2*c^5*h*j*l*m - 1400832*a^8*b^4*c^4*h*j*l*m - 1105920*a^9*
b^2*c^5*g*k*l*m - 253440*a^7*b^6*c^3*h*j*l*m - 69120*a^7*b^6*c^3*g*k*l*m +
11520*a^6*b^8*c^2*h*j*l*m + 6912*a^6*b^8*c^2*g*k*l*m + 4423680*a^8*b^3*c^5*
e*k*l*m + 2506752*a^8*b^3*c^5*f*j*l*m + 1843200*a^8*b^3*c^5*g*j*k*m + 13271
04*a^8*b^3*c^5*h*j*k*l + 838656*a^7*b^5*c^4*f*j*l*m + 783360*a^7*b^5*c^4*g*
j*k*m + 691200*a^7*b^5*c^4*h*j*k*l + 138240*a^7*b^5*c^4*e*k*l*m + 69120*a^6
*b^7*c^3*h*j*k*l - 53760*a^6*b^7*c^3*f*j*l*m + 18432*a^6*b^7*c^3*g*j*k*m -
13824*a^6*b^7*c^3*e*k*l*m - 2304*a^5*b^9*c^2*g*j*k*m + 2543616*a^8*b^3*c^5*
g*h*l*m + 829440*a^7*b^5*c^4*g*h*l*m - 34560*a^6*b^7*c^3*g*h*l*m - 8183808*
a^8*b^2*c^6*d*j*l*m - 3686400*a^8*b^2*c^6*e*j*k*m - 2285568*a^7*b^4*c^5*d*j
*l*m - 1695744*a^8*b^2*c^6*f*j*k*l - 1566720*a^7*b^4*c^5*e*j*k*m - 1400832*
a^7*b^4*c^5*f*j*k*l + 741888*a^6*b^6*c^4*d*j*l*m - 253440*a^6*b^6*c^4*f*j*k
*l - 80640*a^5*b^8*c^3*d*j*l*m - 36864*a^6*b^6*c^4*e*j*k*m + 11520*a^5*b^8*
c^3*f*j*k*l + 4608*a^5*b^8*c^3*e*j*k*m + 6700032*a^8*b^2*c^6*f*h*k*m + 5103
360*a^7*b^4*c^5*f*h*k*m - 5087232*a^8*b^2*c^6*e*h*l*m - 2838528*a^7*b^4*c^5
*f*g*l*m - 1843200*a^8*b^2*c^6*f*g*l*m - 1695744*a^8*b^2*c^6*g*h*j*m - 1658
880*a^7*b^4*c^5*g*h*k*l - 1658880*a^7*b^4*c^5*e*h*l*m - 1400832*a^7*b^4*c^5
*g*h*j*m - 663552*a^8*b^2*c^6*g*h*k*l + 483840*a^6*b^6*c^4*f*h*k*m - 253440
*a^6*b^6*c^4*g*h*j*m - 207360*a^6*b^6*c^4*g*h*k*l + 161280*a^6*b^6*c^4*f*g*
l*m + 69120*a^6*b^6*c^4*e*h*l*m - 50040*a^5*b^8*c^3*f*h*k*m + 11520*a^5*b^8
*c^3*g*h*j*m + 180*a^4*b^10*c^2*f*h*k*m + 4202496*a^7*b^3*c^6*d*j*k*l + 635
904*a^6*b^5*c^5*d*j*k*l - 276480*a^5*b^7*c^4*d*j*k*l + 34560*a^4*b^9*c^3*d*
j*k*l - 16671744*a^7*b^3*c^6*d*h*k*m + 12275712*a^7*b^3*c^6*d*g*l*m + 56770
56*a^7*b^3*c^6*e*f*l*m + 4423680*a^7*b^3*c^6*e*g*k*m + 3317760*a^7*b^3*c^6*
e*h*k*l + 2801664*a^7*b^3*c^6*e*h*j*m - 2709504*a^6*b^5*c^5*d*g*l*m + 25436
16*a^7*b^3*c^6*f*g*k*l + 2506752*a^7*b^3*c^6*f*g*j*m + 1843200*a^7*b^3*c^6*
f*h*j*l + 1327104*a^7*b^3*c^6*g*h*j*k + 838656*a^6*b^5*c^5*f*g*j*m + 829440
*a^6*b^5*c^5*f*g*k*l + 783360*a^6*b^5*c^5*f*h*j*l + 691200*a^6*b^5*c^5*g*h*
j*k + 665280*a^5*b^7*c^4*d*h*k*m + 506880*a^6*b^5*c^5*e*h*j*m + 414720*a^6*
b^5*c^5*e*h*k*l - 322560*a^6*b^5*c^5*e*f*l*m + 241920*a^5*b^7*c^4*d*g*l*m +
 138240*a^6*b^5*c^5*e*g*k*m - 108540*a^4*b^9*c^3*d*h*k*m + 69120*a^5*b^7*c^
4*g*h*j*k - 53760*a^5*b^7*c^4*f*g*j*m - 51840*a^6*b^5*c^5*d*h*k*m - 34560*a
^5*b^7*c^4*f*g*k*l - 23040*a^5*b^7*c^4*e*h*j*m + 18432*a^5*b^7*c^4*f*h*j*l
- 13824*a^5*b^7*c^4*e*g*k*m - 2304*a^4*b^9*c^3*f*h*j*l + 1296*a^3*b^11*c^2*
d*h*k*m + 31924224*a^7*b^2*c^7*d*f*k*m - 24551424*a^7*b^2*c^7*d*e*l*m + 106
16832*a^7*b^2*c^7*e*g*j*l - 8183808*a^7*b^2*c^7*d*g*j*m - 5529600*a^7*b^2*c
^7*d*h*j*l + 5419008*a^6*b^4*c^6*d*e*l*m + 5308416*a^6*b^4*c^6*e*g*j*l - 50
87232*a^7*b^2*c^7*e*f*k*l - 5013504*a^7*b^2*c^7*e*f*j*m + 4868352*a^6*b^4*c

^6*d*f*k*m - 4644864*a^7*b^2*c^7*d*g*k*l - 3981312*a^6*b^4*c^6*d*g*k*l - 26
54208*a^7*b^2*c^7*e*h*j*k - 2367360*a^5*b^6*c^5*d*f*k*m - 2285568*a^6*b^4*c
^6*d*g*j*m - 2211840*a^6*b^4*c^6*d*h*j*l - 1695744*a^7*b^2*c^7*f*g*j*k - 16
77312*a^6*b^4*c^6*e*f*j*m - 1658880*a^6*b^4*c^6*e*f*k*l - 1400832*a^6*b^4*c
^6*f*g*j*k - 1382400*a^6*b^4*c^6*e*h*j*k + 1036800*a^5*b^6*c^5*d*g*k*l + 74
1888*a^5*b^6*c^5*d*g*j*m - 483840*a^5*b^6*c^5*d*e*l*m + 317952*a^5*b^6*c^5*
d*h*j*l + 268920*a^4*b^8*c^4*d*f*k*m - 253440*a^5*b^6*c^5*f*g*j*k - 138240*
a^5*b^6*c^5*e*h*j*k + 107520*a^5*b^6*c^5*e*f*j*m - 103680*a^4*b^8*c^4*d*g*k
*l - 80640*a^4*b^8*c^4*d*g*j*m + 69120*a^5*b^6*c^5*e*f*k*l + 11520*a^4*b^8*
c^4*f*g*j*k + 6912*a^4*b^8*c^4*d*h*j*l - 6912*a^3*b^10*c^3*d*h*j*l + 6120*a
^3*b^10*c^3*d*f*k*m - 1368*a^2*b^12*c^2*d*f*k*m - 5087232*a^7*b^2*c^7*e*g*h
*m - 2211840*a^6*b^4*c^6*f*g*h*l - 1658880*a^6*b^4*c^6*e*g*h*m - 1105920*a^
7*b^2*c^7*f*g*h*l - 69120*a^5*b^6*c^5*f*g*h*l + 69120*a^5*b^6*c^5*e*g*h*m +
 6912*a^4*b^8*c^4*f*g*h*l + 7962624*a^6*b^3*c^7*d*e*k*l - 22164480*a^6*b^3*
c^7*d*f*h*m + 5160960*a^6*b^3*c^7*d*f*j*l + 4571136*a^6*b^3*c^7*d*e*j*m + 4
202496*a^6*b^3*c^7*d*g*j*k + 2801664*a^6*b^3*c^7*e*f*j*k - 2073600*a^5*b^5*
c^6*d*e*k*l - 1483776*a^5*b^5*c^6*d*e*j*m + 635904*a^5*b^5*c^6*d*g*j*k + 50
6880*a^5*b^5*c^6*e*f*j*k - 354816*a^4*b^7*c^5*d*f*j*l + 322560*a^5*b^5*c^6*
d*f*j*l - 276480*a^4*b^7*c^5*d*g*j*k + 207360*a^4*b^7*c^5*d*e*k*l + 161280*
a^4*b^7*c^5*d*e*j*m + 59904*a^3*b^9*c^4*d*f*j*l + 34560*a^3*b^9*c^4*d*g*j*k
 - 23040*a^4*b^7*c^5*e*f*j*k - 2304*a^2*b^11*c^3*d*f*j*l + 8294400*a^6*b^3*
c^7*d*g*h*l + 5677056*a^6*b^3*c^7*e*f*g*m + 4423680*a^6*b^3*c^7*e*f*h*l + 3
317760*a^6*b^3*c^7*e*g*h*k + 2805120*a^5*b^5*c^6*d*f*h*m + 1843200*a^6*b^3*
c^7*f*g*h*j - 829440*a^5*b^5*c^6*d*g*h*l + 783360*a^5*b^5*c^6*f*g*h*j + 437
184*a^4*b^7*c^5*d*f*h*m + 414720*a^5*b^5*c^6*e*g*h*k - 322560*a^5*b^5*c^6*e
*f*g*m - 146268*a^3*b^9*c^4*d*f*h*m + 138240*a^5*b^5*c^6*e*f*h*l - 62208*a^
4*b^7*c^5*d*g*h*l + 20736*a^3*b^9*c^4*d*g*h*l + 18432*a^4*b^7*c^5*f*g*h*j -
 13824*a^4*b^7*c^5*e*f*h*l + 9360*a^2*b^11*c^3*d*f*h*m - 2304*a^3*b^9*c^4*f
*g*h*j - 8404992*a^6*b^2*c^8*d*e*j*k - 24551424*a^6*b^2*c^8*d*e*g*m + 21150
720*a^6*b^2*c^8*d*f*h*k - 1271808*a^5*b^4*c^7*d*e*j*k + 552960*a^4*b^6*c^6*
d*e*j*k - 69120*a^3*b^8*c^5*d*e*j*k - 16588800*a^6*b^2*c^8*d*e*h*l - 774144
0*a^6*b^2*c^8*d*f*g*l + 6946560*a^5*b^4*c^7*d*f*h*k - 5529600*a^6*b^2*c^8*d
*g*h*j + 5419008*a^5*b^4*c^7*d*e*g*m - 5087232*a^6*b^2*c^8*e*f*g*k - 387072
0*a^5*b^4*c^7*d*f*g*l - 3686400*a^6*b^2*c^8*e*f*h*j - 2211840*a^5*b^4*c^7*d
*g*h*j - 1755648*a^4*b^6*c^6*d*f*h*k - 1658880*a^5*b^4*c^7*e*f*g*k + 165888
0*a^5*b^4*c^7*d*e*h*l - 1566720*a^5*b^4*c^7*e*f*h*j + 1451520*a^4*b^6*c^6*d
*f*g*l - 483840*a^4*b^6*c^6*d*e*g*m + 317952*a^4*b^6*c^6*d*g*h*j - 193536*a
^3*b^8*c^5*d*f*g*l + 124416*a^4*b^6*c^6*d*e*h*l + 114696*a^3*b^8*c^5*d*f*h*
k + 69120*a^4*b^6*c^6*e*f*g*k - 41472*a^3*b^8*c^5*d*e*h*l - 36864*a^4*b^6*c
^6*e*f*h*j + 14580*a^2*b^10*c^4*d*f*h*k + 6912*a^3*b^8*c^5*d*g*h*j - 6912*a
^2*b^10*c^4*d*g*h*j + 6912*a^2*b^10*c^4*d*f*g*l + 4608*a^3*b^8*c^5*e*f*h*j
+ 7962624*a^5*b^3*c^8*d*e*g*k + 7741440*a^5*b^3*c^8*d*e*f*l + 5160960*a^5*b
^3*c^8*d*f*g*j + 4423680*a^5*b^3*c^8*d*e*h*j - 2903040*a^4*b^5*c^7*d*e*f*l
- 2073600*a^4*b^5*c^7*d*e*g*k - 635904*a^4*b^5*c^7*d*e*h*j + 387072*a^3*b^7
*c^6*d*e*f*l - 354816*a^3*b^7*c^6*d*f*g*j + 322560*a^4*b^5*c^7*d*f*g*j + 20

7360*a^3*b^7*c^6*d*e*g*k + 59904*a^2*b^9*c^5*d*f*g*j - 13824*a^3*b^7*c^6*d*
e*h*j + 13824*a^2*b^9*c^5*d*e*h*j - 13824*a^2*b^9*c^5*d*e*f*l + 4423680*a^5
*b^3*c^8*e*f*g*h + 138240*a^4*b^5*c^7*e*f*g*h - 13824*a^3*b^7*c^6*e*f*g*h -
10321920*a^5*b^2*c^9*d*e*f*j + 709632*a^3*b^6*c^7*d*e*f*j - 645120*a^4*b^4
*c^8*d*e*f*j - 119808*a^2*b^8*c^6*d*e*f*j - 16588800*a^5*b^2*c^9*d*e*g*h +
1658880*a^4*b^4*c^8*d*e*g*h + 124416*a^3*b^6*c^7*d*e*g*h - 41472*a^2*b^8*c^
6*d*e*g*h + 7741440*a^4*b^3*c^9*d*e*f*g - 2903040*a^3*b^5*c^8*d*e*f*g + 387
072*a^2*b^7*c^7*d*e*f*g + 3456*a^7*b^8*c*k*l^2*m + 12672*a^7*b^8*c*j*l*m^2
+ 384*a^5*b^10*c*j^2*k*m - 1635840*a^10*b*c^5*h*k*m^2 - 1009152*a^9*b*c^6*h
^2*k*m + 3690*a^6*b^9*c*h*k*m^2 + 1152*a^6*b^9*c*g*l*m^2 - 540*a^5*b^10*c*h
*k^2*m + 54*a^4*b^11*c*h^2*k*m + 565248*a^9*b*c^6*h*j^2*m - 39771648*a^7*b*
c^8*d^2*k*m - 2496000*a^8*b*c^7*f^2*k*m - 1543680*a^9*b*c^6*f*k^2*m + 1980*
a^5*b^10*c*f*k*m^2 - 384*a^5*b^10*c*g*j*m^2 - 180*a^4*b^11*c*f*k^2*m + 6*a^
2*b^13*c*f^2*k*m - 10298880*a^9*b*c^6*d*k*m^2 + 2580480*a^9*b*c^6*e*j*m^2 +
5310*a^4*b^11*c*d*k*m^2 - 1674*a*b^13*c^2*d^2*k*m - 540*a^3*b^12*c*d*k^2*m
- 10616832*a^7*b*c^8*e^2*j*l - 3538944*a^8*b*c^7*e*j^2*l + 2727936*a^8*b*c
^7*d*j^2*m - 2496000*a^9*b*c^6*f*h*m^2 - 1543680*a^8*b*c^7*f*h^2*m + 565248
*a^8*b*c^7*f*j^2*k - 270*a^4*b^11*c*f*h*m^2 - 59512320*a^6*b*c^9*d^2*f*m +
5087232*a^7*b*c^8*e^2*h*m + 1105920*a^8*b*c^7*e*j*k^2 - 3456*a*b^12*c^3*d^2
*j*l - 1635840*a^7*b*c^8*f^2*h*k - 1009152*a^8*b*c^7*f*h*k^2 + 10260*a*b^12
*c^3*d^2*h*m - 684*a^3*b^12*c*d*h*m^2 - 24675840*a^6*b*c^9*d^2*h*k - 155520
00*a^8*b*c^7*d*f*m^2 + 24551424*a^6*b*c^9*d*e^2*m - 3939840*a^7*b*c^8*d*h^2
*k + 1105920*a^7*b*c^8*e*h^2*j - 25074*a*b^11*c^4*d^2*f*m + 10530*a*b^11*c^
4*d^2*h*k + 10368*a*b^11*c^4*d^2*g*l + 420*a*b^12*c^3*d*f^2*m - 378*a^2*b^1
3*c*d*f*m^2 - 10616832*a^6*b*c^9*e^2*g*j + 5087232*a^6*b*c^9*e^2*f*k - 3538
944*a^7*b*c^8*e*g*j^2 + 1843200*a^7*b*c^8*d*h*j^2 - 7994880*a^6*b*c^9*d*f^2
*k - 4990464*a^7*b*c^8*d*f*k^2 + 2580480*a^6*b*c^9*e*f^2*j + 65664*a*b^10*c
^5*d^2*g*j - 27972*a*b^10*c^5*d^2*f*k - 20736*a*b^10*c^5*d^2*e*l + 1260*a*b
^11*c^4*d*f^2*k + 54*a*b^13*c^2*d*f*k^2 + 23224320*a^5*b*c^10*d^2*e*j - 370
62144*a^5*b*c^10*d^2*f*h + 384*a*b^12*c^3*d*f*j^2 - 131328*a*b^9*c^6*d^2*e*
j - 5985792*a^6*b*c^9*d*f*h^2 + 206010*a*b^9*c^6*d^2*f*h - 6300*a*b^10*c^5*
d*f^2*h + 1350*a*b^11*c^4*d*f*h^2 + 16588800*a^5*b*c^10*d*e^2*h + 3456*a*b^
10*c^5*d*f*g^2 + 435456*a*b^8*c^7*d^2*e*g + 13824*a*b^8*c^7*d*e^2*f - 14745
60*a^9*c^7*e*j*k*m + 460800*a^9*c^7*f*h*k*m + 3225600*a^8*c^8*d*f*k*m - 245
7600*a^8*c^8*e*f*j*m - 884736*a^8*c^8*e*h*j*k - 6193152*a^7*c^9*d*e*j*k + 1
935360*a^7*c^9*d*f*h*k - 1474560*a^7*c^9*e*f*h*j - 10321920*a^6*c^10*d*e*f*
j - 1105920*a^9*b^4*c^3*k*l^2*m - 552960*a^10*b^2*c^4*k*l^2*m - 34560*a^8*b
^6*c^2*k*l^2*m - 1290240*a^10*b^2*c^4*j*l*m^2 - 860160*a^9*b^4*c^3*j*l*m^2
- 80640*a^8*b^6*c^2*j*l*m^2 - 737280*a^9*b^2*c^5*j^2*k*m - 568320*a^8*b^4*c
^4*j^2*k*m - 136704*a^7*b^6*c^3*j^2*k*m - 2304*a^6*b^8*c^2*j^2*k*m + 127180
8*a^9*b^3*c^4*h*l^2*m - 552960*a^9*b^2*c^5*j*k^2*l - 552960*a^8*b^4*c^4*j*k
^2*l + 414720*a^8*b^5*c^3*h*l^2*m - 145152*a^7*b^6*c^3*j*k^2*l - 17280*a^7*
b^7*c^2*h*l^2*m - 3456*a^6*b^8*c^2*j*k^2*l - 3640320*a^9*b^3*c^4*h*k*m^2 -
2626560*a^8*b^3*c^5*h^2*k*m + 2211840*a^9*b^2*c^5*h*k^2*m + 2056320*a^8*b^4
*c^4*h*k^2*m + 1935360*a^9*b^3*c^4*g*l*m^2 - 1143360*a^8*b^5*c^3*h*k*m^2 -

1097280*a^7*b^5*c^4*h^2*k*m + 364608*a^7*b^6*c^3*h*k^2*m + 322560*a^8*b^5*c
^3*g*l*m^2 - 56160*a^6*b^7*c^3*h^2*k*m - 40320*a^7*b^7*c^2*g*l*m^2 + 27936*
a^7*b^7*c^2*h*k*m^2 - 3780*a^6*b^8*c^2*h*k^2*m + 2970*a^5*b^9*c^2*h^2*k*m -
1419264*a^8*b^4*c^4*f*l^2*m - 1105920*a^7*b^4*c^5*g^2*k*m - 921600*a^9*b^2
*c^5*f*l^2*m - 829440*a^8*b^4*c^4*h*k*l^2 + 749568*a^8*b^3*c^5*h*j^2*m - 55
2960*a^8*b^2*c^6*g^2*k*m - 331776*a^9*b^2*c^5*h*k*l^2 + 317952*a^7*b^5*c^4*
h*j^2*m - 103680*a^7*b^6*c^3*h*k*l^2 + 80640*a^7*b^6*c^3*f*l^2*m + 38400*a^
6*b^7*c^3*h*j^2*m - 34560*a^6*b^6*c^4*g^2*k*m + 3456*a^5*b^8*c^3*g^2*k*m -
1920*a^5*b^9*c^2*h*j^2*m - 5142528*a^7*b^3*c^6*f^2*k*m + 5068800*a^9*b^2*c^
5*f*k*m^2 - 3870720*a^9*b^2*c^5*e*l*m^2 - 3755520*a^8*b^3*c^5*f*k^2*m + 300
0960*a^8*b^4*c^4*f*k*m^2 - 1290240*a^9*b^2*c^5*g*j*m^2 - 1085760*a^7*b^5*c^
4*f*k^2*m - 959040*a^6*b^5*c^5*f^2*k*m - 860160*a^8*b^4*c^4*g*j*m^2 + 82944
0*a^8*b^3*c^5*g*k^2*l - 645120*a^8*b^4*c^4*e*l*m^2 - 552960*a^8*b^2*c^6*h^2
*j*l - 552960*a^7*b^4*c^5*h^2*j*l + 414720*a^7*b^5*c^4*g*k^2*l - 145152*a^6
*b^6*c^4*h^2*j*l + 103200*a^5*b^7*c^4*f^2*k*m - 80640*a^7*b^6*c^3*g*j*m^2 +
80640*a^7*b^6*c^3*e*l*m^2 + 41280*a^7*b^6*c^3*f*k*m^2 - 37188*a^6*b^8*c^2*
f*k*m^2 + 13536*a^6*b^7*c^3*f*k^2*m + 12672*a^6*b^8*c^2*g*j*m^2 + 10368*a^6
*b^7*c^3*g*k^2*l + 5490*a^5*b^9*c^2*f*k^2*m - 3456*a^5*b^8*c^3*h^2*j*l - 23
04*a^6*b^8*c^2*e*l*m^2 + 810*a^4*b^9*c^3*f^2*k*m - 270*a^3*b^11*c^2*f^2*k*m
+ 6137856*a^8*b^3*c^5*d*l^2*m - 4423680*a^7*b^2*c^7*e^2*k*m - 2654208*a^8*
b^3*c^5*g*j*l^2 - 2654208*a^7*b^3*c^6*g^2*j*l + 1769472*a^8*b^2*c^6*g*j^2*l
+ 1769472*a^7*b^4*c^5*g*j^2*l - 1354752*a^7*b^5*c^4*d*l^2*m - 1327104*a^7*
b^5*c^4*g*j*l^2 - 1327104*a^6*b^5*c^5*g^2*j*l + 1271808*a^8*b^3*c^5*f*k*l^2
- 1040384*a^8*b^2*c^6*f*j^2*m - 697344*a^7*b^4*c^5*f*j^2*m - 516096*a^8*b^
2*c^6*h*j^2*k - 451584*a^7*b^4*c^5*h*j^2*k + 442368*a^6*b^6*c^4*g*j^2*l + 4
14720*a^7*b^5*c^4*f*k*l^2 - 138240*a^6*b^6*c^4*h*j^2*k - 138240*a^6*b^4*c^6
*e^2*k*m - 121856*a^6*b^6*c^4*f*j^2*m + 120960*a^6*b^7*c^3*d*l^2*m - 17280*
a^6*b^7*c^3*f*k*l^2 + 13824*a^5*b^6*c^5*e^2*k*m - 11520*a^5*b^8*c^3*h*j^2*k
+ 8960*a^5*b^8*c^3*f*j^2*m + 10851840*a^8*b^2*c^6*d*k^2*m - 10464768*a^6*b
^3*c^7*d^2*k*m - 10275840*a^8*b^3*c^5*d*k*m^2 + 7121088*a^5*b^5*c^6*d^2*k*m
+ 3127680*a^7*b^4*c^5*d*k^2*m + 1720320*a^8*b^3*c^5*e*j*m^2 - 1658880*a^8*
b^2*c^6*e*k^2*l - 1290240*a^7*b^2*c^7*f^2*j*l + 1271808*a^7*b^3*c^6*g^2*h*m
- 1222560*a^4*b^7*c^5*d^2*k*m + 999360*a^7*b^5*c^4*d*k*m^2 - 860160*a^6*b^
4*c^6*f^2*j*l - 829440*a^7*b^4*c^5*e*k^2*l - 705024*a^6*b^6*c^4*d*k^2*m - 5
52960*a^8*b^2*c^6*g*j*k^2 - 552960*a^7*b^4*c^5*g*j*k^2 + 414720*a^6*b^5*c^5
*g^2*h*m + 319392*a^6*b^7*c^3*d*k*m^2 + 161280*a^7*b^5*c^4*e*j*m^2 - 145152
*a^6*b^6*c^4*g*j*k^2 - 85734*a^5*b^9*c^2*d*k*m^2 - 80640*a^5*b^6*c^5*f^2*j*
l - 25344*a^6*b^7*c^3*e*j*m^2 + 23490*a^3*b^9*c^4*d^2*k*m - 20736*a^6*b^6*c
^4*e*k^2*l - 17280*a^5*b^7*c^4*g^2*h*m + 14148*a^5*b^8*c^3*d*k^2*m + 13716*
a^2*b^11*c^3*d^2*k*m + 12690*a^4*b^10*c^2*d*k^2*m + 12672*a^4*b^8*c^4*f^2*j
*l - 3456*a^5*b^8*c^3*g*j*k^2 + 768*a^5*b^9*c^2*e*j*m^2 - 384*a^3*b^10*c^3*
f^2*j*l + 5308416*a^8*b^2*c^6*e*j*l^2 - 5308416*a^6*b^3*c^7*e^2*j*l - 51425
28*a^8*b^3*c^5*f*h*m^2 + 5068800*a^7*b^2*c^7*f^2*h*m - 3755520*a^7*b^3*c^6*
f*h^2*m - 3538944*a^7*b^3*c^6*e*j^2*l + 3000960*a^6*b^4*c^6*f^2*h*m + 26542
08*a^7*b^4*c^5*e*j*l^2 - 2322432*a^8*b^2*c^6*d*k*l^2 + 2125824*a^7*b^3*c^6*

d*j^2*m - 1990656*a^7*b^4*c^5*d*k*l^2 - 1085760*a^6*b^5*c^5*f*h^2*m - 95904
0*a^7*b^5*c^4*f*h*m^2 - 884736*a^6*b^5*c^5*e*j^2*l + 829440*a^7*b^3*c^6*g*h
^2*l + 749568*a^7*b^3*c^6*f*j^2*k + 518400*a^6*b^6*c^4*d*k*l^2 + 414720*a^6
*b^5*c^5*g*h^2*l + 317952*a^6*b^5*c^5*f*j^2*k + 133632*a^6*b^5*c^5*d*j^2*m
+ 103200*a^6*b^7*c^3*f*h*m^2 - 96768*a^5*b^7*c^4*d*j^2*m - 51840*a^5*b^8*c^
3*d*k*l^2 + 41280*a^5*b^6*c^5*f^2*h*m + 38400*a^5*b^7*c^4*f*j^2*k - 37188*a
^4*b^8*c^4*f^2*h*m + 13536*a^5*b^7*c^4*f*h^2*m + 13440*a^4*b^9*c^3*d*j^2*m
+ 10368*a^5*b^7*c^4*g*h^2*l + 5490*a^4*b^9*c^3*f*h^2*m + 1980*a^3*b^10*c^3*
f^2*h*m - 1920*a^4*b^9*c^3*f*j^2*k + 810*a^5*b^9*c^2*f*h*m^2 - 180*a^3*b^11
*c^2*f*h^2*m - 30*a^2*b^12*c^2*f^2*h*m + 30067200*a^6*b^2*c^8*d^2*h*m - 116
12160*a^6*b^2*c^8*d^2*j*l + 1658880*a^6*b^3*c^7*e^2*h*m + 1596672*a^4*b^6*c
^6*d^2*j*l - 1419264*a^6*b^4*c^6*f*g^2*m - 1105920*a^7*b^4*c^5*f*h*l^2 + 11
05920*a^7*b^3*c^6*e*j*k^2 - 921600*a^7*b^2*c^7*f*g^2*m - 829440*a^6*b^4*c^6
*g^2*h*k - 552960*a^8*b^2*c^6*f*h*l^2 - 508032*a^3*b^8*c^5*d^2*j*l - 331776
*a^7*b^2*c^7*g^2*h*k + 290304*a^6*b^5*c^5*e*j*k^2 - 103680*a^5*b^6*c^5*g^2*
h*k + 80640*a^5*b^6*c^5*f*g^2*m - 69120*a^5*b^5*c^6*e^2*h*m + 65664*a^2*b^1
0*c^4*d^2*j*l - 34560*a^6*b^6*c^4*f*h*l^2 + 6912*a^5*b^7*c^4*e*j*k^2 + 3456
*a^5*b^8*c^3*f*h*l^2 + 11930112*a^8*b^2*c^6*d*h*m^2 + 8432640*a^7*b^2*c^7*d
*h^2*m + 4450176*a^7*b^4*c^5*d*h*m^2 + 4337280*a^6*b^4*c^6*d*h^2*m - 387072
0*a^8*b^2*c^6*e*g*m^2 - 3640320*a^6*b^3*c^7*f^2*h*k - 2885760*a^5*b^4*c^7*d
^2*h*m - 2844288*a^4*b^6*c^6*d^2*h*m - 2626560*a^7*b^3*c^6*f*h*k^2 + 221184
0*a^7*b^2*c^7*f*h^2*k + 2056320*a^6*b^4*c^6*f*h^2*k + 1935360*a^6*b^3*c^7*f
^2*g*l - 1916928*a^7*b^2*c^7*d*j^2*k - 1687680*a^6*b^6*c^4*d*h*m^2 - 165888
0*a^7*b^2*c^7*e*h^2*l - 1143360*a^5*b^5*c^6*f^2*h*k - 1097280*a^6*b^5*c^5*f
*h*k^2 + 1019412*a^3*b^8*c^5*d^2*h*m - 1007424*a^5*b^6*c^5*d*h^2*m - 912384
*a^6*b^4*c^6*d*j^2*k - 829440*a^6*b^4*c^6*e*h^2*l - 645120*a^7*b^4*c^5*e*g*
m^2 - 552960*a^7*b^2*c^7*g*h^2*j - 552960*a^6*b^4*c^6*g*h^2*j + 364608*a^5*
b^6*c^5*f*h^2*k + 322560*a^5*b^5*c^6*f^2*g*l + 197460*a^5*b^8*c^3*d*h*m^2 -
145152*a^5*b^6*c^5*g*h^2*j - 143802*a^2*b^10*c^4*d^2*h*m + 80640*a^6*b^6*c
^4*e*g*m^2 - 56160*a^5*b^7*c^4*f*h*k^2 + 51948*a^4*b^8*c^4*d*h^2*m - 40320*
a^4*b^7*c^5*f^2*g*l + 34560*a^4*b^8*c^4*d*j^2*k + 27936*a^4*b^7*c^5*f^2*h*k
- 20736*a^5*b^6*c^5*e*h^2*l - 13824*a^5*b^6*c^5*d*j^2*k + 10800*a^3*b^10*c
^3*d*h^2*m - 5760*a^3*b^10*c^3*d*j^2*k - 3780*a^4*b^8*c^4*f*h^2*k + 3690*a^
3*b^9*c^4*f^2*h*k - 3456*a^4*b^8*c^4*g*h^2*j + 2970*a^4*b^9*c^3*f*h*k^2 - 2
304*a^5*b^8*c^3*e*g*m^2 + 1152*a^3*b^9*c^4*f^2*g*l - 540*a^3*b^10*c^3*f*h^2
*k - 540*a^2*b^12*c^2*d*h^2*m - 90*a^4*b^10*c^2*d*h*m^2 - 90*a^2*b^11*c^3*f
^2*h*k + 54*a^3*b^11*c^2*f*h*k^2 + 15925248*a^6*b^2*c^8*e^2*g*l - 7962624*a
^7*b^3*c^6*e*g*l^2 - 7962624*a^6*b^3*c^7*e*g^2*l + 23385600*a^6*b^2*c^8*d*f
^2*m + 6137856*a^6*b^3*c^7*d*g^2*m - 5677056*a^6*b^2*c^8*e^2*f*m + 4147200*
a^7*b^3*c^6*d*h*l^2 - 3317760*a^6*b^2*c^8*e^2*h*k - 1354752*a^5*b^5*c^6*d*g
^2*m + 1271808*a^6*b^3*c^7*f*g^2*k - 737280*a^7*b^2*c^7*f*h*j^2 + 17418240*
a^5*b^3*c^8*d^2*g*l - 568320*a^6*b^4*c^6*f*h*j^2 - 414720*a^6*b^5*c^5*d*h*l
^2 + 414720*a^5*b^5*c^6*f*g^2*k - 414720*a^5*b^4*c^7*e^2*h*k + 322560*a^5*b
^4*c^7*e^2*f*m - 136704*a^5*b^6*c^5*f*h*j^2 + 120960*a^4*b^7*c^5*d*g^2*m -
31104*a^5*b^7*c^4*d*h*l^2 - 17280*a^4*b^7*c^5*f*g^2*k + 10368*a^4*b^9*c^3*d

*h*l^2 - 2304*a^4*b^8*c^4*f*h*j^2 + 384*a^3*b^10*c^3*f*h*j^2 + 50042880*a^5
*b^2*c^9*d^2*f*k - 13271040*a^5*b^3*c^8*d^2*h*k - 13149696*a^7*b^3*c^6*d*f*
m^2 + 10906560*a^4*b^5*c^7*d^2*f*m - 8709120*a^4*b^5*c^7*d^2*g*l - 7418880*
a^5*b^3*c^8*d^2*f*m + 7133184*a^7*b^2*c^7*d*h*k^2 - 6428160*a^6*b^3*c^7*d*h
^2*k + 5593536*a^4*b^5*c^7*d^2*h*k - 3870720*a^6*b^2*c^8*e*f^2*l + 3369600*
a^6*b^4*c^6*d*h*k^2 + 3148992*a^6*b^5*c^5*d*f*m^2 - 2985696*a^3*b^7*c^6*d^2
*f*m + 1959552*a^3*b^7*c^6*d^2*g*l - 1658880*a^7*b^2*c^7*e*g*k^2 - 1505280*
a^4*b^6*c^6*d*f^2*m - 1290240*a^6*b^2*c^8*f^2*g*j - 34836480*a^5*b^2*c^9*d^
2*e*l + 1105920*a^6*b^3*c^7*e*h^2*j - 860160*a^5*b^4*c^7*f^2*g*j - 829440*a
^6*b^4*c^6*e*g*k^2 - 692064*a^3*b^7*c^6*d^2*h*k - 689472*a^5*b^5*c^6*d*h^2*
k - 645120*a^5*b^4*c^7*e*f^2*l - 388800*a^5*b^6*c^5*d*h*k^2 + 378954*a^2*b^
9*c^5*d^2*f*m + 362880*a^5*b^4*c^7*d*f^2*m + 296964*a^3*b^8*c^5*d*f^2*m + 2
90304*a^5*b^5*c^6*e*h^2*j + 277344*a^4*b^7*c^5*d*h^2*k - 217728*a^2*b^9*c^5
*d^2*g*l - 80640*a^4*b^6*c^6*f^2*g*j + 80640*a^4*b^6*c^6*e*f^2*l - 77070*a^
4*b^9*c^3*d*f*m^2 - 30240*a^5*b^7*c^4*d*f*m^2 - 28350*a^3*b^9*c^4*d*h^2*k -
 26406*a^2*b^9*c^5*d^2*h*k - 21060*a^4*b^8*c^4*d*h*k^2 - 20736*a^5*b^6*c^5*
e*g*k^2 - 19278*a^2*b^10*c^4*d*f^2*m + 12672*a^3*b^8*c^5*f^2*g*j + 10044*a^
3*b^10*c^3*d*h*k^2 + 8820*a^3*b^11*c^2*d*f*m^2 + 6912*a^4*b^7*c^5*e*h^2*j -
 2304*a^3*b^8*c^5*e*f^2*l - 1620*a^2*b^11*c^3*d*h^2*k - 384*a^2*b^10*c^4*f^
2*g*j + 162*a^2*b^12*c^2*d*h*k^2 - 5419008*a^5*b^3*c^8*d*e^2*m + 5308416*a^
6*b^2*c^8*e*g^2*j - 5308416*a^5*b^3*c^8*e^2*g*j - 3870720*a^7*b^2*c^7*d*f*l
^2 - 3538944*a^6*b^3*c^7*e*g*j^2 + 2654208*a^5*b^4*c^7*e*g^2*j - 2322432*a^
6*b^2*c^8*d*g^2*k - 1990656*a^5*b^4*c^7*d*g^2*k - 1935360*a^6*b^4*c^6*d*f*l
^2 + 1658880*a^6*b^3*c^7*d*h*j^2 + 1658880*a^5*b^3*c^8*e^2*f*k - 884736*a^5
*b^5*c^6*e*g*j^2 + 725760*a^5*b^6*c^5*d*f*l^2 + 17418240*a^4*b^4*c^8*d^2*e*
l + 518400*a^4*b^6*c^6*d*g^2*k + 483840*a^4*b^5*c^7*d*e^2*m + 262656*a^5*b^
5*c^6*d*h*j^2 - 96768*a^4*b^8*c^4*d*f*l^2 - 69120*a^4*b^5*c^7*e^2*f*k - 552
96*a^4*b^7*c^5*d*h*j^2 - 51840*a^3*b^8*c^5*d*g^2*k + 3456*a^3*b^10*c^3*d*f*
l^2 + 1152*a^3*b^9*c^4*d*h*j^2 + 1152*a^2*b^11*c^3*d*h*j^2 - 15431040*a^4*b
^4*c^8*d^2*f*k - 13248000*a^5*b^3*c^8*d*f^2*k - 11612160*a^5*b^2*c^9*d^2*g*
j - 10063872*a^6*b^3*c^7*d*f*k^2 - 3919104*a^3*b^6*c^7*d^2*e*l + 2554560*a^
4*b^5*c^7*d*f^2*k + 1720320*a^5*b^3*c^8*e*f^2*j + 1596672*a^3*b^6*c^7*d^2*g
*j + 1518912*a^3*b^6*c^7*d^2*f*k - 1105920*a^5*b^4*c^7*f*g^2*h + 838080*a^5
*b^5*c^6*d*f*k^2 - 552960*a^6*b^2*c^8*f*g^2*h - 508032*a^2*b^8*c^6*d^2*g*j
+ 435456*a^2*b^8*c^6*d^2*e*l + 161280*a^4*b^5*c^7*e*f^2*j + 116640*a^4*b^7*
c^5*d*f*k^2 + 106812*a^2*b^8*c^6*d^2*f*k - 98208*a^3*b^7*c^6*d*f^2*k - 3456
0*a^4*b^6*c^6*f*g^2*h - 27270*a^3*b^9*c^4*d*f*k^2 - 26334*a^2*b^9*c^5*d*f^2
*k - 25344*a^3*b^7*c^6*e*f^2*j + 3456*a^3*b^8*c^5*f*g^2*h + 768*a^2*b^9*c^5
*e*f^2*j - 702*a^2*b^11*c^3*d*f*k^2 - 7962624*a^5*b^2*c^9*d*e^2*k - 2580480
*a^6*b^2*c^8*d*f*j^2 + 2073600*a^4*b^4*c^8*d*e^2*k - 1658880*a^6*b^2*c^8*e*
g*h^2 - 967680*a^5*b^4*c^7*d*f*j^2 - 829440*a^5*b^4*c^7*e*g*h^2 - 207360*a^
3*b^6*c^7*d*e^2*k + 64512*a^4*b^6*c^6*d*f*j^2 + 39168*a^3*b^8*c^5*d*f*j^2 -
 20736*a^4*b^6*c^6*e*g*h^2 - 9216*a^2*b^10*c^4*d*f*j^2 - 4423680*a^5*b^2*c^
9*e^2*f*h + 4147200*a^5*b^3*c^8*d*g^2*h - 3193344*a^3*b^5*c^8*d^2*e*j + 101
6064*a^2*b^7*c^7*d^2*e*j - 414720*a^4*b^5*c^7*d*g^2*h - 138240*a^4*b^4*c^8*

e^2*f*h - 31104*a^3*b^7*c^6*d*g^2*h + 13824*a^3*b^6*c^7*e^2*f*h + 10368*a^2
*b^9*c^5*d*g^2*h + 15630336*a^5*b^2*c^9*d*f^2*h - 14459904*a^4*b^3*c^9*d^2*
f*h + 9630144*a^3*b^5*c^8*d^2*f*h - 8764416*a^5*b^3*c^8*d*f*h^2 - 3870720*a
^5*b^2*c^9*e*f^2*g + 2867328*a^4*b^4*c^8*d*f^2*h - 2095200*a^2*b^7*c^7*d^2*
f*h - 1414080*a^3*b^6*c^7*d*f^2*h - 34836480*a^4*b^2*c^10*d^2*e*g - 645120*
a^4*b^4*c^8*e*f^2*g + 306720*a^3*b^7*c^6*d*f*h^2 + 197820*a^2*b^8*c^6*d*f^2
*h + 146880*a^4*b^5*c^7*d*f*h^2 + 80640*a^3*b^6*c^7*e*f^2*g - 55350*a^2*b^9
*c^5*d*f*h^2 - 2304*a^2*b^8*c^6*e*f^2*g - 3870720*a^5*b^2*c^9*d*f*g^2 - 193
5360*a^4*b^4*c^8*d*f*g^2 - 1658880*a^4*b^3*c^9*d*e^2*h + 725760*a^3*b^6*c^7
*d*f*g^2 + 17418240*a^3*b^4*c^9*d^2*e*g - 124416*a^3*b^5*c^8*d*e^2*h - 9676
8*a^2*b^8*c^6*d*f*g^2 + 41472*a^2*b^7*c^7*d*e^2*h - 3919104*a^2*b^6*c^8*d^2
*e*g - 7741440*a^4*b^2*c^10*d*e^2*f + 2903040*a^3*b^4*c^9*d*e^2*f - 387072*
a^2*b^6*c^8*d*e^2*f - 20160*a^8*b^7*c*l^2*m^2 - 1648128*a^10*b^3*c^3*k*m^3
- 898560*a^9*b^3*c^4*k^3*m - 354240*a^9*b^5*c^2*k*m^3 - 354240*a^8*b^5*c^3*
k^3*m - 21600*a^7*b^7*c^2*k^3*m - 13950*a^7*b^8*c*k^2*m^2 + 430080*a^10*b*c
^5*j^2*m^2 - 1984*a^6*b^9*c*j^2*m^2 - 884736*a^9*b^3*c^4*j*l^3 - 589824*a^8
*b^3*c^5*j^3*l - 442368*a^8*b^5*c^3*j*l^3 - 294912*a^7*b^5*c^4*j^3*l - 4915
2*a^6*b^7*c^3*j^3*l + 1359360*a^10*b^2*c^4*h*m^3 + 1173120*a^9*b^4*c^3*h*m^
3 + 743040*a^7*b^4*c^5*h^3*m + 622080*a^8*b^2*c^6*h^3*m + 184320*a^9*b*c^6*
j^2*k^2 + 107136*a^6*b^6*c^4*h^3*m - 32640*a^8*b^6*c^2*h*m^3 + 540*a^5*b^8*
c^3*h^3*m - 270*a^4*b^10*c^2*h^3*m - 180*a^5*b^10*c*h^2*m^2 - 2293760*a^9*b
^3*c^4*f*m^3 - 2293760*a^6*b^3*c^7*f^3*m + 1327104*a^8*b^4*c^4*g*l^3 + 1327
104*a^6*b^4*c^6*g^3*l - 622080*a^8*b^3*c^5*h*k^3 - 622080*a^7*b^3*c^6*h^3*k
- 326592*a^7*b^5*c^4*h*k^3 - 326592*a^6*b^5*c^5*h^3*k - 199360*a^8*b^5*c^3
*f*m^3 - 199360*a^5*b^5*c^6*f^3*m + 61920*a^7*b^7*c^2*f*m^3 + 61920*a^4*b^7
*c^5*f^3*m - 38880*a^6*b^7*c^3*h*k^3 - 38880*a^5*b^7*c^4*h^3*k - 3682*a^3*b
^9*c^4*f^3*m - 810*a^5*b^9*c^2*h*k^3 - 810*a^4*b^9*c^3*h^3*k - 70*a^3*b^12*
c*f^2*m^2 + 70*a^2*b^11*c^3*f^3*m + 3870720*a^8*b*c^7*e^2*m^2 + 184320*a^8*
b*c^7*h^2*j^2 - 14152320*a^4*b^4*c^8*d^3*m + 10644480*a^5*b^2*c^9*d^3*m + 5
483520*a^9*b^2*c^5*d*m^3 + 4269888*a^3*b^6*c^7*d^3*m - 2654208*a^8*b^3*c^5*
e*l^3 + 1359360*a^6*b^2*c^8*f^3*k + 1330560*a^8*b^4*c^4*d*m^3 + 1173120*a^5
*b^4*c^7*f^3*k - 884736*a^6*b^3*c^7*g^3*j - 826560*a^7*b^6*c^3*d*m^3 + 7430
40*a^7*b^4*c^5*f*k^3 + 622080*a^8*b^2*c^6*f*k^3 - 607068*a^2*b^8*c^6*d^3*m
- 589824*a^7*b^3*c^6*g*j^3 - 442368*a^5*b^5*c^6*g^3*j - 294912*a^6*b^5*c^5*
g*j^3 + 145188*a^6*b^8*c^2*d*m^3 + 107136*a^6*b^6*c^4*f*k^3 - 49152*a^5*b^7
*c^4*g*j^3 - 32640*a^4*b^6*c^6*f^3*k - 5796*a^3*b^8*c^5*f^3*k + 540*a^5*b^8
*c^3*f*k^3 - 270*a^4*b^10*c^2*f*k^3 + 210*a^2*b^10*c^4*f^3*k + 19077120*a^4
*b^3*c^9*d^3*k + 1658880*a^7*b*c^8*e^2*k^2 + 430080*a^7*b*c^8*f^2*j^2 + 353
8944*a^5*b^2*c^9*e^3*j - 2488320*a^7*b^3*c^6*d*k^3 - 2379456*a^3*b^5*c^8*d^
3*k + 1179648*a^7*b^2*c^7*e*j^3 + 589824*a^6*b^4*c^6*e*j^3 + 98304*a^5*b^6*
c^5*e*j^3 - 95904*a^2*b^7*c^7*d^3*k - 57024*a^6*b^5*c^5*d*k^3 + 49248*a^5*b
^7*c^4*d*k^3 - 4050*a^4*b^9*c^3*d*k^3 - 810*a^3*b^11*c^2*d*k^3 - 486*a*b^12
*c^3*d^2*k^2 + 3870720*a^6*b*c^9*d^2*j^2 - 1648128*a^5*b^3*c^8*f^3*h - 8985
60*a^6*b^3*c^7*f*h^3 - 354240*a^5*b^5*c^6*f*h^3 - 354240*a^4*b^5*c^7*f^3*h
+ 43680*a^3*b^7*c^6*f^3*h - 21600*a^4*b^7*c^5*f*h^3 - 9792*a*b^11*c^4*d^2*j

^2 + 1350*a^3*b^9*c^4*f*h^3 - 1050*a^2*b^9*c^5*f^3*h + 1658880*a^6*b*c^9*e^
2*h^2 + 16547328*a^4*b^2*c^10*d^3*h - 12306816*a^3*b^4*c^9*d^3*h + 37310976
*a^3*b^3*c^10*d^3*f + 3037824*a^2*b^6*c^8*d^3*h - 2654208*a^5*b^3*c^8*e*g^3
 + 1949184*a^6*b^2*c^8*d*h^3 + 1296000*a^5*b^4*c^7*d*h^3 - 155520*a^4*b^6*c
^6*d*h^3 - 40500*a*b^10*c^5*d^2*h^2 - 8100*a^3*b^8*c^5*d*h^3 + 4050*a^2*b^1
0*c^4*d*h^3 + 3870720*a^5*b*c^10*e^2*f^2 + 34836480*a^4*b*c^11*d^2*e^2 - 10
8864*a*b^9*c^6*d^2*g^2 - 8068032*a^2*b^5*c^9*d^3*f - 5623296*a^4*b^3*c^9*d*
f^3 + 1737792*a^3*b^5*c^8*d*f^3 - 260190*a*b^8*c^7*d^2*f^2 - 211680*a^2*b^7
*c^7*d*f^3 - 435456*a*b^7*c^8*d^2*e^2 - 245760*a^10*c^6*j^2*k*m - 384*a^6*b
^10*j*l*m^2 + 138240*a^10*c^6*h*k^2*m - 90*a^5*b^11*h*k*m^2 + 384000*a^10*c
^6*f*k*m^2 - 2211840*a^8*c^8*e^2*k*m - 409600*a^9*c^7*f*j^2*m - 147456*a^9*
c^7*h*j^2*k - 30*a^4*b^12*f*k*m^2 + 967680*a^9*c^7*d*k^2*m + 384000*a^8*c^8
*f^2*h*m - 90*a^3*b^13*d*k*m^2 + 20321280*a^7*c^9*d^2*h*m - 883200*a^11*b*c
^4*k*m^3 - 317952*a^10*b*c^5*k^3*m + 43680*a^8*b^7*c*k*m^3 + 1350*a^6*b^9*c
*k^3*m - 270*b^14*c^2*d^2*h*m + 6*a^3*b^13*f*h*m^2 + 4838400*a^9*c^7*d*h*m^
2 + 2903040*a^8*c^8*d*h^2*m - 1032192*a^8*c^8*d*j^2*k + 138240*a^8*c^8*f*h^
2*k - 3686400*a^7*c^9*e^2*f*m - 1327104*a^7*c^9*e^2*h*k - 393216*a^9*b*c^6*
j^3*l - 245760*a^8*c^8*f*h*j^2 - 810*b^13*c^3*d^2*h*k + 630*b^13*c^3*d^2*f*
m + 18*a^2*b^14*d*h*m^2 + 2688000*a^7*c^9*d*f^2*m + 580608*a^8*c^8*d*h*k^2
 - 5796*a^7*b^8*c*h*m^3 - 3456*b^12*c^4*d^2*g*j + 1890*b^12*c^4*d^2*f*k + 67
73760*a^6*c^10*d^2*f*k - 1344000*a^10*b*c^5*f*m^3 - 1344000*a^7*b*c^8*f^3*m
 - 207360*a^9*b*c^6*h*k^3 - 207360*a^8*b*c^7*h^3*k - 3682*a^6*b^9*c*f*m^3 -
 9289728*a^6*c^10*d*e^2*k - 1720320*a^7*c^9*d*f*j^2 - 50803200*a^5*b*c^10*d
^3*k + 6912*b^11*c^5*d^2*e*j - 10616832*a^6*b*c^9*e^3*l - 2211840*a^6*c^10*
e^2*f*h - 393216*a^8*b*c^7*g*j^3 + 43416*a*b^10*c^5*d^3*m - 9576*a^5*b^10*c
*d*m^3 - 9450*b^11*c^5*d^2*f*h - 504*a*b^14*c*d^2*m^2 + 1612800*a^6*c^10*d*
f^2*h - 1036800*a^8*b*c^7*d*k^3 + 45198*a*b^9*c^6*d^3*k - 20736*b^10*c^6*d^
2*e*g - 75188736*a^4*b*c^11*d^3*f - 883200*a^6*b*c^9*f^3*h - 317952*a^7*b*c
^8*f*h^3 - 15482880*a^5*c^11*d*e^2*f - 10616832*a^5*b*c^10*e^3*g - 345060*a
*b^8*c^7*d^3*h - 4262400*a^5*b*c^10*d*f^3 + 852768*a*b^7*c^8*d^3*f + 7350*a
*b^9*c^6*d*f^3 + 967680*a^10*b^3*c^3*l^2*m^2 + 161280*a^9*b^5*c^2*l^2*m^2 +
 1684224*a^10*b^2*c^4*k^2*m^2 + 1264320*a^9*b^4*c^3*k^2*m^2 + 126720*a^8*b^
6*c^2*k^2*m^2 + 501760*a^9*b^3*c^4*j^2*m^2 + 414720*a^9*b^3*c^4*k^2*l^2 + 2
07360*a^8*b^5*c^3*k^2*l^2 + 170240*a^8*b^5*c^3*j^2*m^2 + 9216*a^7*b^7*c^2*j
^2*m^2 + 5184*a^7*b^7*c^2*k^2*l^2 + 884736*a^9*b^2*c^5*j^2*l^2 + 884736*a^8
*b^4*c^4*j^2*l^2 + 221184*a^7*b^6*c^3*j^2*l^2 + 1419840*a^8*b^4*c^4*h^2*m^2
 + 1387008*a^9*b^2*c^5*h^2*m^2 + 276480*a^8*b^3*c^5*j^2*k^2 + 140544*a^7*b^
5*c^4*j^2*k^2 + 84960*a^7*b^6*c^3*h^2*m^2 + 25344*a^6*b^7*c^3*j^2*k^2 - 801
0*a^6*b^8*c^2*h^2*m^2 + 576*a^5*b^9*c^2*j^2*k^2 + 967680*a^8*b^3*c^5*g^2*m^
2 + 414720*a^8*b^3*c^5*h^2*l^2 + 207360*a^7*b^5*c^4*h^2*l^2 + 161280*a^7*b^
5*c^4*g^2*m^2 - 20160*a^6*b^7*c^3*g^2*m^2 + 5184*a^6*b^7*c^3*h^2*l^2 + 576*
a^5*b^9*c^2*g^2*m^2 + 3808000*a^8*b^2*c^6*f^2*m^2 + 1990656*a^7*b^4*c^5*g^2
*l^2 + 1643712*a^7*b^4*c^5*f^2*m^2 + 803520*a^7*b^4*c^5*h^2*k^2 + 725760*a^
8*b^2*c^6*h^2*k^2 + 207360*a^6*b^6*c^4*h^2*k^2 - 125440*a^6*b^6*c^4*f^2*m^2
 - 13790*a^5*b^8*c^3*f^2*m^2 + 10530*a^5*b^8*c^3*h^2*k^2 + 1785*a^4*b^10*c^

2*f^2*m^2 + 81*a^4*b^10*c^2*h^2*k^2 + 18427392*a^7*b^2*c^7*d^2*m^2 + 967680
*a^7*b^3*c^6*f^2*l^2 + 645120*a^7*b^3*c^6*e^2*m^2 + 414720*a^7*b^3*c^6*g^2*
k^2 + 276480*a^7*b^3*c^6*h^2*j^2 + 207360*a^6*b^5*c^5*g^2*k^2 + 161280*a^6*
b^5*c^5*f^2*l^2 + 140544*a^6*b^5*c^5*h^2*j^2 - 80640*a^6*b^5*c^5*e^2*m^2 +
25344*a^5*b^7*c^4*h^2*j^2 - 20160*a^5*b^7*c^4*f^2*l^2 + 5184*a^5*b^7*c^4*g^
2*k^2 + 2304*a^5*b^7*c^4*e^2*m^2 + 576*a^4*b^9*c^3*h^2*j^2 + 576*a^4*b^9*c^
3*f^2*l^2 + 7962624*a^7*b^2*c^7*e^2*l^2 - 4148928*a^6*b^4*c^6*d^2*m^2 + 141
9840*a^6*b^4*c^6*f^2*k^2 + 1387008*a^7*b^2*c^7*f^2*k^2 - 1183392*a^5*b^6*c^
5*d^2*m^2 + 884736*a^7*b^2*c^7*g^2*j^2 + 884736*a^6*b^4*c^6*g^2*j^2 + 64575
0*a^4*b^8*c^4*d^2*m^2 + 221184*a^5*b^6*c^5*g^2*j^2 - 115920*a^3*b^10*c^3*d^
2*m^2 + 84960*a^5*b^6*c^5*f^2*k^2 + 10836*a^2*b^12*c^2*d^2*m^2 - 8010*a^4*b
^8*c^4*f^2*k^2 - 180*a^3*b^10*c^3*f^2*k^2 + 9*a^2*b^12*c^2*f^2*k^2 + 870912
0*a^6*b^3*c^7*d^2*l^2 - 4354560*a^5*b^5*c^6*d^2*l^2 + 979776*a^4*b^7*c^5*d^
2*l^2 + 829440*a^6*b^3*c^7*e^2*k^2 + 17480448*a^6*b^2*c^8*d^2*k^2 + 501760*
a^6*b^3*c^7*f^2*j^2 + 170240*a^5*b^5*c^6*f^2*j^2 - 108864*a^3*b^9*c^4*d^2*l
^2 + 20736*a^5*b^5*c^6*e^2*k^2 + 9216*a^4*b^7*c^5*f^2*j^2 + 5184*a^2*b^11*c
^3*d^2*l^2 - 1984*a^3*b^9*c^4*f^2*j^2 + 64*a^2*b^11*c^3*f^2*j^2 + 3538944*a
^6*b^2*c^8*e^2*j^2 - 3302208*a^5*b^4*c^7*d^2*k^2 + 884736*a^5*b^4*c^7*e^2*j
^2 + 414720*a^6*b^3*c^7*g^2*h^2 + 207360*a^5*b^5*c^6*g^2*h^2 - 103680*a^4*b
^6*c^6*d^2*k^2 + 101250*a^3*b^8*c^5*d^2*k^2 - 5751*a^2*b^10*c^4*d^2*k^2 + 5
184*a^4*b^7*c^5*g^2*h^2 + 1935360*a^5*b^3*c^8*d^2*j^2 + 1684224*a^6*b^2*c^8
*f^2*h^2 + 1264320*a^5*b^4*c^7*f^2*h^2 - 532224*a^4*b^5*c^7*d^2*j^2 + 12672
0*a^4*b^6*c^6*f^2*h^2 - 96768*a^3*b^7*c^6*d^2*j^2 + 62784*a^2*b^9*c^5*d^2*j
^2 - 13950*a^3*b^8*c^5*f^2*h^2 + 225*a^2*b^10*c^4*f^2*h^2 + 967680*a^5*b^3*
c^8*f^2*g^2 + 829440*a^5*b^3*c^8*e^2*h^2 + 161280*a^4*b^5*c^7*f^2*g^2 + 207
36*a^4*b^5*c^7*e^2*h^2 - 20160*a^3*b^7*c^6*f^2*g^2 + 576*a^2*b^9*c^5*f^2*g^
2 + 11487744*a^5*b^2*c^9*d^2*h^2 + 7962624*a^5*b^2*c^9*e^2*g^2 + 35525376*a
^4*b^2*c^10*d^2*f^2 - 1412640*a^3*b^6*c^7*d^2*h^2 + 461376*a^4*b^4*c^8*d^2*
h^2 + 375030*a^2*b^8*c^6*d^2*h^2 + 8709120*a^4*b^3*c^9*d^2*g^2 - 4354560*a^
3*b^5*c^8*d^2*g^2 + 979776*a^2*b^7*c^7*d^2*g^2 + 645120*a^4*b^3*c^9*e^2*f^2
 - 80640*a^3*b^5*c^8*e^2*f^2 + 2304*a^2*b^7*c^7*e^2*f^2 - 15269184*a^3*b^4*
c^9*d^2*f^2 + 2870784*a^2*b^6*c^8*d^2*f^2 - 17418240*a^3*b^3*c^10*d^2*e^2 +
 3919104*a^2*b^5*c^9*d^2*e^2 + 54*b^15*c*d^2*k*m + 6*a*b^15*d*f*m^2 + 11520
0*a^11*c^5*k^2*m^2 + 576*a^7*b^9*l^2*m^2 + 225*a^6*b^10*k^2*m^2 + 64*a^5*b^
11*j^2*m^2 + 345600*a^10*c^6*h^2*m^2 + 9*a^4*b^12*h^2*m^2 + 320000*a^9*c^7*
f^2*m^2 + 41472*a^9*c^7*h^2*k^2 + 16934400*a^8*c^8*d^2*m^2 + 345600*a^8*c^8
*f^2*k^2 + 81*b^14*c^2*d^2*k^2 + 3538944*a^7*c^9*e^2*j^2 + 2032128*a^7*c^9*
d^2*k^2 + 492800*a^11*b^2*c^3*m^4 + 351456*a^10*b^4*c^2*m^4 + 576*b^13*c^3*
d^2*j^2 + 331776*a^9*b^4*c^3*l^4 + 115200*a^7*c^9*f^2*h^2 + 142560*a^8*b^4*
c^4*k^4 + 103680*a^9*b^2*c^5*k^4 + 32400*a^7*b^6*c^3*k^4 + 2025*b^12*c^4*d^
2*h^2 + 2025*a^6*b^8*c^2*k^4 + 6096384*a^6*c^10*d^2*h^2 + 131072*a^8*b^2*c^
6*j^4 + 98304*a^7*b^4*c^5*j^4 + 32768*a^6*b^6*c^4*j^4 + 5184*b^11*c^5*d^2*g
^2 + 4096*a^5*b^8*c^3*j^4 + 11025*b^10*c^6*d^2*f^2 + 5644800*a^5*c^11*d^2*f
^2 + 142560*a^6*b^4*c^6*h^4 + 103680*a^7*b^2*c^7*h^4 + 32400*a^5*b^6*c^5*h^
4 + 20736*b^9*c^7*d^2*e^2 + 2025*a^4*b^8*c^4*h^4 + 331776*a^5*b^4*c^7*g^4 +

492800*a^5*b^2*c^9*f^4 + 351456*a^4*b^4*c^8*f^4 - 43120*a^3*b^6*c^7*f^4 + 1225*a^2*b^8*c^6*f^4 - 27433728*a^3*b^2*c^11*d^4 + 6446304*a^2*b^4*c^10*d^4 - 1050*a^7*b^9*k*m^3 + 384000*a^11*c^5*h*m^3 + 138240*a^9*c^7*h^3*m + 210*a^6*b^10*h*m^3 + 47416320*a^6*c^10*d^3*m - 1134*b^12*c^4*d^3*m + 70*a^5*b^11*f*m^3 + 2688000*a^10*c^6*d*m^3 + 384000*a^7*c^9*f^3*k + 138240*a^9*c^7*f*k^3 - 3402*b^11*c^5*d^3*k + 210*a^4*b^12*d*m^3 + 7077888*a^6*c^10*e^3*j + 786432*a^8*c^8*e*j^3 - 43120*a^9*b^6*c*m^4 + 28449792*a^5*c^11*d^3*h + 17010*b^10*c^6*d^3*h + 580608*a^7*c^9*d*h^3 - 39690*b^9*c^7*d^3*f - 734832*a*b^6*c^9*d^4 + 9*b^16*d^2*m^2 + 160000*a^12*c^4*m^4 + 1225*a^8*b^8*m^4 + 20736*a^10*c^6*k^4 + 65536*a^9*c^7*j^4 + 20736*a^8*c^8*h^4 + 49787136*a^4*c^12*d^4 + 160000*a^6*c^10*f^4 + 5308416*a^5*c^11*e^4 + 35721*b^8*c^8*d^4 + a^2*b^14*f^2*m^2, z, k1)*x*(8388608*a^11*b*c^10 - 512*a^4*b^15*c^3 + 14336*a^5*b^13*c^4 - 172032*a^6*b^11*c^5 + 1146880*a^7*b^9*c^6 - 4587520*a^8*b^7*c^7 + 11010048*a^9*b^5*c^8 - 14680064*a^10*b^3*c^9))/(64*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 + 240*a^6*b^8*c^3 - 1280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6))) - (983040*a^7*c^9*e*f + 589824*a^8*c^8*e*k + 327680*a^8*c^8*f*j + 196608*a^9*c^7*j*k - 3244032*a^6*b*c^9*d*e - 884736*a^7*b*c^8*e*h - 491520*a^7*b*c^8*f*g - 1081344*a^7*b*c^8*d*j - 1277952*a^8*b*c^7*e*m - 491520*a^8*b*c^7*f*l - 294912*a^8*b*c^7*g*k - 294912*a^8*b*c^7*h*j - 425984*a^9*b*c^6*j*m - 294912*a^9*b*c^6*k*l - 4608*a^2*b^9*c^5*d*e + 87552*a^3*b^7*c^6*d*e - 681984*a^4*b^5*c^7*d*e + 2433024*a^5*b^3*c^8*d*e + 2304*a^2*b^10*c^4*d*g - 43776*a^3*b^8*c^5*d*g - 1536*a^3*b^8*c^5*e*f + 340992*a^4*b^6*c^6*d*g + 39936*a^4*b^6*c^6*e*f - 1216512*a^5*b^4*c^7*d*g - 184320*a^5*b^4*c^7*e*f + 1622016*a^6*b^2*c^8*d*g - 49152*a^6*b^2*c^8*e*f + 768*a^3*b^9*c^4*f*g - 4608*a^4*b^7*c^5*e*h - 19968*a^4*b^7*c^5*f*g - 18432*a^5*b^5*c^6*e*h + 92160*a^5*b^5*c^6*f*g + 368640*a^6*b^3*c^7*e*h + 24576*a^6*b^3*c^7*f*g - 768*a^2*b^11*c^3*d*j + 13056*a^3*b^9*c^4*d*j - 84480*a^4*b^7*c^5*d*j + 178176*a^5*b^5*c^6*d*j + 270336*a^6*b^3*c^7*d*j + 2304*a^4*b^8*c^4*g*h + 9216*a^5*b^6*c^5*g*h - 184320*a^6*b^4*c^6*g*h + 442368*a^7*b^2*c^7*g*h + 2304*a^3*b^10*c^3*d*l - 256*a^3*b^10*c^3*f*j - 43776*a^4*b^8*c^4*d*l + 6144*a^4*b^8*c^4*f*j + 340992*a^5*b^6*c^5*d*l + 27648*a^5*b^6*c^5*e*k - 17408*a^5*b^6*c^5*f*j - 1216512*a^6*b^4*c^6*d*l - 184320*a^6*b^4*c^6*e*k - 69632*a^6*b^4*c^6*f*j + 1622016*a^7*b^2*c^7*d*l + 147456*a^7*b^2*c^7*e*k + 147456*a^7*b^2*c^7*f*j + 768*a^4*b^9*c^3*f*l - 768*a^4*b^9*c^3*h*j + 1536*a^5*b^7*c^4*e*m - 19968*a^5*b^7*c^4*f*l - 13824*a^5*b^7*c^4*g*k - 4608*a^5*b^7*c^4*h*j - 92160*a^6*b^5*c^5*e*m + 92160*a^6*b^5*c^5*f*l + 92160*a^6*b^5*c^5*g*k + 55296*a^6*b^5*c^5*h*j + 663552*a^7*b^3*c^6*e*m + 24576*a^7*b^3*c^6*f*l - 73728*a^7*b^3*c^6*g*k - 24576*a^7*b^3*c^6*h*j - 768*a^5*b^8*c^3*g*m + 2304*a^5*b^8*c^3*h*l + 46080*a^6*b^6*c^4*g*m + 9216*a^6*b^6*c^4*h*l - 331776*a^7*b^4*c^5*g*m - 184320*a^7*b^4*c^5*h*l + 638976*a^8*b^2*c^6*g*m + 442368*a^8*b^2*c^6*h*l + 4608*a^5*b^8*c^3*j*k - 21504*a^6*b^6*c^4*j*k - 36864*a^7*b^4*c^5*j*k + 147456*a^8*b^2*c^6*j*k + 256*a^5*b^9*c^2*j*m - 14848*a^6*b^7*c^3*j*m - 13824*a^6*b^7*c^3*k*l + 79872*a^7*b^5*c^4*j*m + 92160*a^7*b^5*c^4*k*l + 8192*a^8*b^3*c^5*j*m - 73728*a^8*b^3*c^5*k*l - 768*a^6*b^8*c^2*l*m + 46080*a^7*b^6*c^3*l*m - 331776*a^8*b^4*c^4*l*m + 638976*a^9*b^2*c^5*l*m)/(512*(

4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 + 240*a^6*b^8*c^3 - 1280*a^7*b
^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6)) + (x*(25600*a^7*c^9*f^2 - 18
*b^12*c^4*d^2 - 451584*a^6*c^10*d^2 - 9216*a^8*c^8*h^2 + 9216*a^9*c^7*k^2 -
2*a^4*b^12*m^2 - 25600*a^10*c^6*m^2 + 504*a*b^10*c^5*d^2 + 73728*a^6*b*c^9
*e^2 + 8192*a^8*b*c^7*j^2 + 88*a^5*b^10*c*m^2 - 6228*a^2*b^8*c^6*d^2 + 4262
4*a^3*b^6*c^7*d^2 - 176256*a^4*b^4*c^8*d^2 + 423936*a^5*b^2*c^9*d^2 + 4608*
a^4*b^5*c^7*e^2 - 36864*a^5*b^3*c^8*e^2 - 2*a^2*b^10*c^4*f^2 + 84*a^3*b^8*c
^5*f^2 - 3520*a^4*b^6*c^6*f^2 + 26240*a^5*b^4*c^7*f^2 - 59904*a^6*b^2*c^8*f
^2 + 1152*a^4*b^7*c^5*g^2 - 9216*a^5*b^5*c^6*g^2 + 18432*a^6*b^3*c^7*g^2 -
468*a^4*b^8*c^4*h^2 + 3456*a^5*b^6*c^5*h^2 - 5760*a^6*b^4*c^6*h^2 + 128*a^4
*b^9*c^3*j^2 - 512*a^5*b^7*c^4*j^2 - 1536*a^6*b^5*c^5*j^2 + 4096*a^7*b^3*c^
6*j^2 - 18*a^4*b^10*c^2*k^2 - 108*a^5*b^8*c^3*k^2 + 576*a^6*b^6*c^4*k^2 + 5
760*a^7*b^4*c^5*k^2 - 23040*a^8*b^2*c^6*k^2 + 1152*a^6*b^7*c^3*l^2 - 9216*a
^7*b^5*c^4*l^2 + 18432*a^8*b^3*c^5*l^2 - 1236*a^6*b^8*c^2*m^2 + 5760*a^7*b^
6*c^3*m^2 - 8320*a^8*b^4*c^4*m^2 + 6144*a^9*b^2*c^5*m^2 - 129024*a^7*c^9*d*
h - 215040*a^8*c^8*d*m + 30720*a^8*c^8*f*k - 30720*a^9*c^7*h*m - 12*a*b^11*
c^4*d*f + 218112*a^6*b*c^9*d*f + 9216*a^7*b*c^8*f*h + 156672*a^7*b*c^8*d*k
+ 49152*a^7*b*c^8*e*j + 25600*a^8*b*c^7*f*m + 9216*a^8*b*c^7*h*k - 12*a^4*b
^11*c*k*m + 21504*a^9*b*c^6*k*m + 420*a^2*b^9*c^5*d*f - 4992*a^3*b^7*c^6*d*
f + 36480*a^4*b^5*c^7*d*f - 144384*a^5*b^3*c^8*d*f - 36*a^2*b^10*c^4*d*h +
360*a^3*b^8*c^5*d*h - 3456*a^4*b^6*c^6*d*h - 4608*a^4*b^6*c^6*e*g + 11520*a
^5*b^4*c^7*d*h + 36864*a^5*b^4*c^7*e*g + 27648*a^6*b^2*c^8*d*h - 73728*a^6*
b^2*c^8*e*g - 12*a^3*b^9*c^4*f*h + 2304*a^4*b^7*c^5*f*h - 17280*a^5*b^5*c^6
*f*h + 30720*a^6*b^3*c^7*f*h + 180*a^3*b^9*c^4*d*k - 2304*a^4*b^7*c^5*d*k +
1536*a^4*b^7*c^5*e*j + 19584*a^5*b^5*c^6*d*k - 9216*a^5*b^5*c^6*e*j - 9216
0*a^6*b^3*c^7*d*k - 168*a^4*b^8*c^4*d*m - 360*a^4*b^8*c^4*f*k - 768*a^4*b^8
*c^4*g*j + 768*a^5*b^6*c^5*d*m - 4608*a^5*b^6*c^5*e*l - 768*a^5*b^6*c^5*f*k
+ 4608*a^5*b^6*c^5*g*j - 11520*a^6*b^4*c^6*d*m + 36864*a^6*b^4*c^6*e*l + 2
5344*a^6*b^4*c^6*f*k + 98304*a^7*b^2*c^7*d*m - 73728*a^7*b^2*c^7*e*l - 7372
8*a^7*b^2*c^7*f*k - 24576*a^7*b^2*c^7*g*j - 140*a^4*b^9*c^3*f*m + 180*a^4*b
^9*c^3*h*k + 3584*a^5*b^7*c^4*f*m + 2304*a^5*b^7*c^4*g*l - 20352*a^6*b^5*c^
5*f*m - 18432*a^6*b^5*c^5*g*l - 8064*a^6*b^5*c^5*h*k + 26624*a^7*b^3*c^6*f*
m + 36864*a^7*b^3*c^6*g*l + 18432*a^7*b^3*c^6*h*k + 60*a^4*b^10*c^2*h*m - 1
560*a^5*b^8*c^3*h*m + 8832*a^6*b^6*c^4*h*m - 13056*a^7*b^4*c^5*h*m + 3072*a
^8*b^2*c^6*h*m - 768*a^5*b^8*c^3*j*l + 4608*a^6*b^6*c^4*j*l - 24576*a^8*b^2
*c^6*j*l + 228*a^5*b^9*c^2*k*m + 384*a^6*b^7*c^3*k*m - 9600*a^7*b^5*c^4*k*m
+ 15360*a^8*b^3*c^5*k*m))/(64*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^
2 + 240*a^6*b^8*c^3 - 1280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^
6))) + (35*a^6*b^7*m^3 - 8000*a^5*c^8*f^3 - 1728*a^8*c^5*k^3 - 567*b^7*c^6*
d^3 + 10368*a*b^5*c^7*d^3 + 169344*a^3*b*c^9*d^3 + 193536*a^4*c^9*d*e^2 - 1
41120*a^4*c^9*d^2*f + 1728*a^6*b*c^6*h^3 + 315*b^8*c^5*d^2*f + 27648*a^5*c^
8*e^2*h - 135*b^9*c^4*d^2*h + 21504*a^6*c^7*d*j^2 - 2880*a^6*c^7*f*h^2 - 84
672*a^5*c^8*d^2*k - 1176*a^7*b^5*c*m^3 + 6400*a^9*b*c^3*m^3 + 3*a^2*b^11*d*
m^2 + 27*b^10*c^3*d^2*k - 14400*a^6*c^7*f^2*k - 8640*a^7*c^6*f*k^2 + a^3*b^
10*f*m^2 + 46080*a^6*c^7*e^2*m + 3072*a^7*c^6*h*j^2 + 9*b^11*c^2*d^2*m - 17

28*a^7*c^6*h^2*k - 8000*a^8*c^5*f*m^2 + 3*a^4*b^9*h*m^2 - 15*a^5*b^8*k*m^2 + 5120*a^8*c^5*j^2*m - 4800*a^9*c^4*k*m^2 - 67824*a^2*b^3*c^8*d^3 + 35*a^2*b^6*c^5*f^3 + 84*a^3*b^4*c^6*f^3 - 12720*a^4*b^2*c^7*f^3 + 540*a^4*b^5*c^4*h^3 + 4320*a^5*b^3*c^5*h^3 - 135*a^5*b^6*c^2*k^3 - 1620*a^6*b^4*c^3*k^3 - 4752*a^7*b^2*c^4*k^3 + 9456*a^8*b^3*c^2*m^3 - 40320*a^5*c^8*d*f*h + 129024*a^5*c^8*d*e*j - 67200*a^6*c^7*d*f*m - 24192*a^6*c^7*d*h*k + 18432*a^6*c^7*e*h*j - 9600*a^7*c^6*f*h*m - 40320*a^7*c^6*d*k*m + 30720*a^7*c^6*e*j*m - 5760*a^8*c^5*h*k*m - 6237*a*b^6*c^6*d^2*f + 210*a*b^7*c^5*d*f^2 + 116160*a^4*b*c^8*d*f^2 - 36864*a^4*b*c^8*e^2*f + 2430*a*b^7*c^5*d^2*h + 133056*a^4*b*c^8*d^2*h + 27648*a^5*b*c^7*d*h^2 + 26880*a^5*b*c^7*f^2*h - 297*a*b^8*c^4*d^2*k + 46656*a^6*b*c^6*d*k^2 - 27648*a^5*b*c^7*e^2*k - 4096*a^6*b*c^6*f*j^2 - 324*a*b^9*c^3*d^2*m - 132*a^3*b^9*c*d*m^2 + 193536*a^5*b*c^7*d^2*m + 63360*a^7*b*c^5*d*m^2 - 51*a^4*b^8*c*f*m^2 + 40000*a^6*b*c^6*f^2*m + 10368*a^7*b*c^5*h*k^2 - 78*a^5*b^7*c*h*m^2 + 8064*a^7*b*c^5*h^2*m - 3072*a^7*b*c^5*j^2*k + 12480*a^8*b*c^4*h*m^2 - 90*a^5*b^7*c*k^2*m + 705*a^6*b^6*c*k*m^2 + 15552*a^8*b*c^4*k^2*m + 6912*a^2*b^4*c^7*d*e^2 - 62208*a^3*b^2*c^8*d*e^2 + 42372*a^2*b^4*c^7*d^2*f - 1764*a^2*b^5*c^6*d*f^2 - 96048*a^3*b^2*c^8*d^2*f - 4608*a^3*b^3*c^7*d*f^2 + 1728*a^2*b^6*c^5*d*g^2 + 2304*a^3*b^3*c^7*e^2*f - 15552*a^3*b^4*c^6*d*g^2 + 48384*a^4*b^2*c^7*d*g^2 - 13716*a^2*b^5*c^6*d^2*h + 405*a^2*b^7*c^4*d*h^2 + 12096*a^3*b^3*c^7*d^2*h - 5400*a^3*b^5*c^5*d*h^2 + 28944*a^4*b^3*c^6*d*h^2 + 576*a^3*b^5*c^5*f*g^2 + 6912*a^4*b^2*c^7*e^2*h - 9216*a^4*b^3*c^6*f*g^2 - 15*a^2*b^7*c^4*f^2*h + 192*a^2*b^8*c^3*d*j^2 - 360*a^3*b^5*c^5*f^2*h - 960*a^3*b^6*c^4*d*j^2 + 135*a^3*b^6*c^4*f*h^2 + 15696*a^4*b^3*c^6*f^2*h - 768*a^4*b^4*c^5*d*j^2 - 5580*a^4*b^4*c^5*f*h^2 + 14592*a^5*b^2*c^6*d*j^2 - 20592*a^5*b^2*c^6*f*h^2 - 999*a^2*b^6*c^5*d^2*k + 27*a^2*b^9*c^2*d*k^2 + 23004*a^3*b^4*c^6*d^2*k - 108*a^3*b^7*c^3*d*k^2 - 84240*a^4*b^2*c^7*d^2*k + 1728*a^4*b^4*c^5*g^2*h - 1404*a^4*b^5*c^4*d*k^2 + 6912*a^5*b^2*c^6*g^2*h + 14688*a^5*b^3*c^5*d*k^2 + 64*a^3*b^7*c^3*f*j^2 - 768*a^4*b^5*c^4*f*j^2 + 1728*a^4*b^6*c^3*d*l^2 - 3840*a^5*b^3*c^5*f*j^2 - 15552*a^5*b^4*c^4*d*l^2 + 48384*a^6*b^2*c^5*d*l^2 + 3717*a^2*b^7*c^4*d^2*m + 3*a^2*b^8*c^3*f^2*k - 15192*a^3*b^5*c^5*d^2*m + 135*a^3*b^6*c^4*f^2*k + 9*a^3*b^8*c^2*f*k^2 - 7920*a^4*b^3*c^6*d^2*m - 2988*a^4*b^4*c^5*f^2*k - 99*a^4*b^6*c^3*f*k^2 + 2079*a^4*b^7*c^2*d*m^2 - 28272*a^5*b^2*c^6*f^2*k - 4500*a^5*b^4*c^4*f*k^2 - 14448*a^5*b^5*c^3*d*m^2 - 20304*a^6*b^2*c^5*f*k^2 + 37104*a^6*b^3*c^4*d*m^2 + 192*a^4*b^6*c^3*h*j^2 + 2304*a^5*b^2*c^6*e^2*m - 6912*a^5*b^3*c^5*g^2*k + 1536*a^5*b^4*c^4*h*j^2 + 576*a^5*b^5*c^3*f*l^2 + 3840*a^6*b^2*c^5*h*j^2 - 9216*a^6*b^3*c^4*f*l^2 + a^2*b^9*c^2*f^2*m + 20*a^3*b^7*c^3*f^2*m - 1596*a^4*b^5*c^4*f^2*m - 243*a^4*b^6*c^3*h^2*k + 27*a^4*b^7*c^2*h*k^2 + 16736*a^5*b^3*c^5*f^2*m - 5940*a^5*b^4*c^4*h^2*k + 1728*a^5*b^5*c^3*h*k^2 + 875*a^5*b^6*c^2*f*m^2 - 13392*a^6*b^2*c^5*h^2*k + 10800*a^6*b^3*c^4*h*k^2 - 2716*a^6*b^4*c^3*f*m^2 - 39600*a^7*b^2*c^4*f*m^2 + 576*a^5*b^4*c^4*g^2*m + 11520*a^6*b^2*c^5*g^2*m + 1728*a^6*b^4*c^3*h*l^2 + 6912*a^7*b^2*c^4*h*l^2 - 81*a^4*b^7*c^2*h^2*m + 720*a^5*b^5*c^3*h^2*m - 768*a^5*b^5*c^3*j^2*k + 17136*a^6*b^3*c^4*h^2*m - 3072*a^6*b^3*c^4*j^2*k - 900*a^6*b^5*c^2*h*m^2 + 22272*a^7*b^3*c^3*h*m^2 + 64*a^5*b^6*c^2*j^2*m + 1536*a^6*b^4*c^3*j^2*m

+ 5376*a^7*b^2*c^4*j^2*m - 6912*a^7*b^3*c^3*k*l^2 + 1260*a^6*b^5*c^2*k^2*m
+ 13248*a^7*b^3*c^3*k^2*m - 6084*a^7*b^4*c^2*k*m^2 - 26256*a^8*b^2*c^3*k*m^
2 + 576*a^7*b^4*c^2*l^2*m + 11520*a^8*b^2*c^3*l^2*m - 193536*a^4*b*c^8*d*e*
g - 90*a*b^8*c^4*d*f*h - 27648*a^5*b*c^7*e*g*h + 18*a*b^9*c^3*d*f*k - 19353
6*a^5*b*c^7*d*e*l + 147456*a^5*b*c^7*d*f*k - 64512*a^5*b*c^7*d*g*j - 24576*
a^5*b*c^7*e*f*j + 6*a*b^10*c^2*d*f*m + 84096*a^6*b*c^6*d*h*m - 46080*a^6*b*
c^6*e*g*m - 27648*a^6*b*c^6*e*h*l + 33408*a^6*b*c^6*f*h*k - 9216*a^6*b*c^6*
g*h*j - 64512*a^6*b*c^6*d*j*l - 18432*a^6*b*c^6*e*j*k + 18*a^2*b^10*c*d*k*m
 + 6*a^3*b^9*c*f*k*m - 46080*a^7*b*c^5*e*l*m + 49920*a^7*b*c^5*f*k*m - 1536
0*a^7*b*c^5*g*j*m - 9216*a^7*b*c^5*h*j*l + 18*a^4*b^8*c*h*k*m - 15360*a^8*b
*c^4*j*l*m - 6912*a^2*b^5*c^6*d*e*g + 62208*a^3*b^3*c^7*d*e*g - 270*a^2*b^6
*c^5*d*f*h + 16056*a^3*b^4*c^6*d*f*h - 2304*a^3*b^4*c^6*e*f*g - 127008*a^4*
b^2*c^7*d*f*h + 36864*a^4*b^2*c^7*e*f*g + 2304*a^2*b^6*c^5*d*e*j - 16128*a^
3*b^4*c^6*d*e*j + 23040*a^4*b^2*c^7*d*e*j - 6912*a^4*b^3*c^6*e*g*h + 306*a^
2*b^7*c^4*d*f*k - 1152*a^2*b^7*c^4*d*g*j - 6912*a^3*b^5*c^5*d*e*l - 5328*a^
3*b^5*c^5*d*f*k + 8064*a^3*b^5*c^5*d*g*j + 768*a^3*b^5*c^5*e*f*j + 62208*a^
4*b^3*c^6*d*e*l + 19872*a^4*b^3*c^6*d*f*k - 11520*a^4*b^3*c^6*d*g*j - 10752
*a^4*b^3*c^6*e*f*j - 48*a^2*b^8*c^3*d*f*m - 216*a^2*b^8*c^3*d*h*k - 2226*a^
3*b^6*c^4*d*f*m + 3456*a^3*b^6*c^4*d*g*l + 1998*a^3*b^6*c^4*d*h*k - 384*a^3
*b^6*c^4*f*g*j + 33384*a^4*b^4*c^5*d*f*m - 31104*a^4*b^4*c^5*d*g*l - 1944*a
^4*b^4*c^5*d*h*k - 2304*a^4*b^4*c^5*e*f*l + 2304*a^4*b^4*c^5*e*h*j + 5376*a
^4*b^4*c^5*f*g*j - 162528*a^5*b^2*c^6*d*f*m + 96768*a^5*b^2*c^6*d*g*l - 872
64*a^5*b^2*c^6*d*h*k + 36864*a^5*b^2*c^6*e*f*l + 27648*a^5*b^2*c^6*e*g*k +
13824*a^5*b^2*c^6*e*h*j + 12288*a^5*b^2*c^6*f*g*j - 72*a^2*b^9*c^2*d*h*m +
2016*a^3*b^7*c^3*d*h*m - 72*a^3*b^7*c^3*f*h*k - 18648*a^4*b^5*c^4*d*h*m + 1
152*a^4*b^5*c^4*f*g*l + 1800*a^4*b^5*c^4*f*h*k - 1152*a^4*b^5*c^4*g*h*j + 6
7392*a^5*b^3*c^5*d*h*m - 2304*a^5*b^3*c^5*e*g*m - 6912*a^5*b^3*c^5*e*h*l -
18432*a^5*b^3*c^5*f*g*l + 27072*a^5*b^3*c^5*f*h*k - 6912*a^5*b^3*c^5*g*h*j
- 1152*a^3*b^7*c^3*d*j*l + 8064*a^4*b^5*c^4*d*j*l - 11520*a^5*b^3*c^5*d*j*l
 - 9216*a^5*b^3*c^5*e*j*k - 24*a^3*b^8*c^2*f*h*m + 1050*a^4*b^6*c^3*f*h*m -
 9576*a^5*b^4*c^4*f*h*m + 3456*a^5*b^4*c^4*g*h*l - 57504*a^6*b^2*c^5*f*h*m
+ 13824*a^6*b^2*c^5*g*h*l - 432*a^3*b^8*c^2*d*k*m + 2394*a^4*b^6*c^3*d*k*m
- 384*a^4*b^6*c^3*f*j*l + 6552*a^5*b^4*c^4*d*k*m + 768*a^5*b^4*c^4*e*j*m +
5376*a^5*b^4*c^4*f*j*l + 4608*a^5*b^4*c^4*g*j*k - 114336*a^6*b^2*c^5*d*k*m
+ 16896*a^6*b^2*c^5*e*j*m + 27648*a^6*b^2*c^5*e*k*l + 12288*a^6*b^2*c^5*f*j
*l + 9216*a^6*b^2*c^5*g*j*k - 186*a^4*b^7*c^2*f*k*m - 384*a^5*b^5*c^3*g*j*m
 - 1152*a^5*b^5*c^3*h*j*l - 2304*a^6*b^3*c^4*e*l*m + 31584*a^6*b^3*c^4*f*k*
m - 8448*a^6*b^3*c^4*g*j*m - 13824*a^6*b^3*c^4*g*k*l - 6912*a^6*b^3*c^4*h*j
*l + 342*a^5*b^6*c^2*h*k*m + 1152*a^6*b^4*c^3*g*l*m - 12600*a^6*b^4*c^3*h*k
*m + 23040*a^7*b^2*c^4*g*l*m - 37728*a^7*b^2*c^4*h*k*m + 4608*a^6*b^4*c^3*j
*k*l + 9216*a^7*b^2*c^4*j*k*l - 384*a^6*b^5*c^2*j*l*m - 8448*a^7*b^3*c^3*j*
l*m)/(512*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 + 240*a^6*b^8*c^3 -
 1280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6)) + (x*(13824*a^4*c
^9*e^3 + 512*a^7*c^6*j^3 - 54*b^7*c^6*d^2*e + 27*b^8*c^5*d^2*g + 13824*a^5*
c^8*e^2*j + 4608*a^6*c^7*e*j^2 - 9*b^9*c^4*d^2*j + a^4*b^9*j*m^2 - 3*a^5*b^

8*l*m^2 - 1728*a^4*b^3*c^6*g^3 + 64*a^4*b^6*c^3*j^3 + 384*a^5*b^4*c^4*j^3 +
768*a^6*b^2*c^5*j^3 - 1728*a^7*b^3*c^3*l^3 - 20160*a^4*c^9*d*e*f - 2880*a^
5*c^8*e*f*h - 12096*a^5*c^8*d*e*k - 6720*a^5*c^8*d*f*j - 4800*a^6*c^7*e*f*m
- 1728*a^6*c^7*e*h*k - 960*a^6*c^7*f*h*j - 4032*a^6*c^7*d*j*k - 2880*a^7*c
^6*e*k*m - 1600*a^7*c^6*f*j*m - 576*a^7*c^6*h*j*k - 960*a^8*c^5*j*k*m + 972
*a*b^5*c^7*d^2*e + 24192*a^3*b*c^9*d^2*e - 486*a*b^6*c^6*d^2*g + 6240*a^4*b
*c^8*e*f^2 - 20736*a^4*b*c^8*e^2*g + 1728*a^5*b*c^7*e*h^2 + 144*a*b^7*c^5*d
^2*j + 8064*a^4*b*c^8*d^2*j + 27*a*b^8*c^4*d^2*l + 2080*a^5*b*c^7*f^2*j + 2
592*a^6*b*c^6*e*k^2 - 20736*a^5*b*c^7*e^2*l - 2304*a^6*b*c^6*g*j^2 + 576*a^
6*b*c^6*h^2*j + 3840*a^7*b*c^5*e*m^2 - 3*a^4*b^8*c*g*m^2 + 864*a^7*b*c^5*j*
k^2 - 2304*a^7*b*c^5*j^2*l - 32*a^5*b^7*c*j*m^2 + 1280*a^8*b*c^4*j*m^2 + 10
2*a^6*b^6*c*l*m^2 - 7344*a^2*b^3*c^8*d^2*e + 3672*a^2*b^4*c^7*d^2*g - 6*a^2
*b^5*c^6*e*f^2 - 12096*a^3*b^2*c^8*d^2*g + 192*a^3*b^3*c^7*e*f^2 + 10368*a^
4*b^2*c^7*e*g^2 + 3*a^2*b^6*c^5*f^2*g - 96*a^3*b^4*c^6*f^2*g - 3120*a^4*b^2
*c^7*f^2*g + 1296*a^4*b^3*c^6*e*h^2 - 900*a^2*b^5*c^6*d^2*j + 1584*a^3*b^3*
c^7*d^2*j + 6912*a^4*b^2*c^7*e^2*j + 1152*a^4*b^4*c^5*e*j^2 - 648*a^4*b^4*c
^5*g*h^2 + 4608*a^5*b^2*c^6*e*j^2 - 864*a^5*b^2*c^6*g*h^2 - 486*a^2*b^6*c^5
*d^2*l - a^2*b^7*c^4*f^2*j + 3672*a^3*b^4*c^6*d^2*l + 30*a^3*b^5*c^5*f^2*j
- 12096*a^4*b^2*c^7*d^2*l + 1104*a^4*b^3*c^6*f^2*j + 54*a^4*b^5*c^4*e*k^2 +
864*a^5*b^3*c^5*e*k^2 + 1728*a^4*b^4*c^5*g^2*j - 576*a^4*b^5*c^4*g*j^2 + 3
456*a^5*b^2*c^6*g^2*j - 2304*a^5*b^3*c^5*g*j^2 + 10368*a^6*b^2*c^5*e*l^2 +
3*a^3*b^6*c^4*f^2*l - 96*a^4*b^4*c^5*f^2*l + 216*a^4*b^5*c^4*h^2*j - 27*a^4
*b^6*c^3*g*k^2 + 6*a^4*b^7*c^2*e*m^2 - 3120*a^5*b^2*c^6*f^2*l + 720*a^5*b^3
*c^5*h^2*j - 432*a^5*b^4*c^4*g*k^2 - 204*a^5*b^5*c^3*e*m^2 - 1296*a^6*b^2*c
^5*g*k^2 + 1488*a^6*b^3*c^4*e*m^2 - 5184*a^5*b^3*c^5*g^2*l - 5184*a^6*b^3*c
^4*g*l^2 - 648*a^5*b^4*c^4*h^2*l + 102*a^5*b^6*c^2*g*m^2 - 864*a^6*b^2*c^5*
h^2*l - 744*a^6*b^4*c^3*g*m^2 - 1920*a^7*b^2*c^4*g*m^2 + 9*a^4*b^7*c^2*j*k^
2 + 162*a^5*b^5*c^3*j*k^2 + 720*a^6*b^3*c^4*j*k^2 - 576*a^5*b^5*c^3*j^2*l -
2304*a^6*b^3*c^4*j^2*l + 1728*a^6*b^4*c^3*j*l^2 + 3456*a^7*b^2*c^4*j*l^2 -
27*a^5*b^6*c^2*k^2*l - 432*a^6*b^4*c^3*k^2*l + 180*a^6*b^5*c^2*j*m^2 - 129
6*a^7*b^2*c^4*k^2*l + 1136*a^7*b^3*c^3*j*m^2 - 744*a^7*b^4*c^2*l*m^2 - 1920
*a^8*b^2*c^3*l*m^2 - 36*a*b^6*c^6*d*e*f + 18*a*b^7*c^5*d*f*g + 15552*a^4*b*
c^8*d*e*h + 10080*a^4*b*c^8*d*f*g - 6*a*b^8*c^4*d*f*j + 1440*a^5*b*c^7*f*g*
h + 21888*a^5*b*c^7*d*e*m + 10080*a^5*b*c^7*d*f*l + 6048*a^5*b*c^7*d*g*k +
5184*a^5*b*c^7*d*h*j + 8064*a^5*b*c^7*e*f*k - 13824*a^5*b*c^7*e*g*j + 5184*
a^6*b*c^6*e*h*m + 2400*a^6*b*c^6*f*g*m + 1440*a^6*b*c^6*f*h*l + 864*a^6*b*c
^6*g*h*k + 7296*a^6*b*c^6*d*j*m + 6048*a^6*b*c^6*d*k*l - 13824*a^6*b*c^6*e*
j*l + 2688*a^6*b*c^6*f*j*k + 2400*a^7*b*c^5*f*l*m + 1440*a^7*b*c^5*g*k*m +
1728*a^7*b*c^5*h*j*m + 864*a^7*b*c^5*h*k*l + 6*a^4*b^8*c*j*k*m - 18*a^5*b^7
*c*k*l*m + 1440*a^8*b*c^4*k*l*m + 900*a^2*b^4*c^7*d*e*f - 4896*a^3*b^2*c^8*
d*e*f - 108*a^2*b^5*c^6*d*e*h - 450*a^2*b^5*c^6*d*f*g + 2448*a^3*b^3*c^7*d*
f*g + 54*a^2*b^6*c^5*d*g*h - 36*a^3*b^4*c^6*e*f*h - 7776*a^4*b^2*c^7*d*g*h
- 6048*a^4*b^2*c^7*e*f*h + 138*a^2*b^6*c^5*d*f*j + 540*a^3*b^4*c^6*d*e*k -
516*a^3*b^4*c^6*d*f*j - 6048*a^4*b^2*c^7*d*e*k - 4992*a^4*b^2*c^7*d*f*j + 1
8*a^3*b^5*c^5*f*g*h + 3024*a^4*b^3*c^6*f*g*h + 18*a^2*b^7*c^4*d*f*l - 18*a^

2*b^7*c^4*d*h*j - 450*a^3*b^5*c^5*d*f*l - 270*a^3*b^5*c^5*d*g*k - 36*a^3*b^
5*c^5*d*h*j - 2016*a^4*b^3*c^6*d*e*m + 2448*a^4*b^3*c^6*d*f*l + 3024*a^4*b^
3*c^6*d*g*k + 2592*a^4*b^3*c^6*d*h*j + 1440*a^4*b^3*c^6*e*f*k - 6912*a^4*b^
3*c^6*e*g*j + 54*a^3*b^6*c^4*d*h*l - 6*a^3*b^6*c^4*f*h*j + 1008*a^4*b^4*c^5
*d*g*m + 420*a^4*b^4*c^5*e*f*m - 540*a^4*b^4*c^5*e*h*k - 720*a^4*b^4*c^5*f*
g*k - 1020*a^4*b^4*c^5*f*h*j - 10944*a^5*b^2*c^6*d*g*m - 7776*a^5*b^2*c^6*d
*h*l - 7392*a^5*b^2*c^6*e*f*m + 20736*a^5*b^2*c^6*e*g*l - 4320*a^5*b^2*c^6*
e*h*k - 4032*a^5*b^2*c^6*f*g*k - 2496*a^5*b^2*c^6*f*h*j + 90*a^3*b^6*c^4*d*
j*k - 828*a^4*b^4*c^5*d*j*k - 4032*a^5*b^2*c^6*d*j*k - 180*a^4*b^5*c^4*e*h*
m - 210*a^4*b^5*c^4*f*g*m + 18*a^4*b^5*c^4*f*h*l + 270*a^4*b^5*c^4*g*h*k +
2880*a^5*b^3*c^5*e*h*m + 3696*a^5*b^3*c^5*f*g*m + 3024*a^5*b^3*c^5*f*h*l +
2160*a^5*b^3*c^5*g*h*k - 336*a^4*b^5*c^4*d*j*m - 270*a^4*b^5*c^4*d*k*l + 24
0*a^4*b^5*c^4*f*j*k + 2976*a^5*b^3*c^5*d*j*m + 3024*a^5*b^3*c^5*d*k*l - 691
2*a^5*b^3*c^5*e*j*l + 1824*a^5*b^3*c^5*f*j*k + 90*a^4*b^6*c^3*g*h*m - 1440*
a^5*b^4*c^4*g*h*m - 2592*a^6*b^2*c^5*g*h*m + 36*a^4*b^6*c^3*e*k*m + 70*a^4*
b^6*c^3*f*j*m - 90*a^4*b^6*c^3*h*j*k + 1008*a^5*b^4*c^4*d*l*m - 324*a^5*b^4
*c^4*e*k*m - 1092*a^5*b^4*c^4*f*j*m - 720*a^5*b^4*c^4*f*k*l + 3456*a^5*b^4*
c^4*g*j*l - 900*a^5*b^4*c^4*h*j*k - 10944*a^6*b^2*c^5*d*l*m - 5472*a^6*b^2*
c^5*e*k*m - 3264*a^6*b^2*c^5*f*j*m - 4032*a^6*b^2*c^5*f*k*l + 6912*a^6*b^2*
c^5*g*j*l - 1728*a^6*b^2*c^5*h*j*k - 18*a^4*b^7*c^2*g*k*m - 30*a^4*b^7*c^2*
h*j*m - 210*a^5*b^5*c^3*f*l*m + 162*a^5*b^5*c^3*g*k*m + 420*a^5*b^5*c^3*h*j
*m + 270*a^5*b^5*c^3*h*k*l + 3696*a^6*b^3*c^4*f*l*m + 2736*a^6*b^3*c^4*g*k*
m + 1824*a^6*b^3*c^4*h*j*m + 2160*a^6*b^3*c^4*h*k*l + 90*a^5*b^6*c^2*h*l*m
- 1440*a^6*b^4*c^3*h*l*m - 2592*a^7*b^2*c^4*h*l*m - 42*a^5*b^6*c^2*j*k*m -
1020*a^6*b^4*c^3*j*k*m - 2304*a^7*b^2*c^4*j*k*m + 162*a^6*b^5*c^2*k*l*m + 2
736*a^7*b^3*c^3*k*l*m))/(64*(4096*a^10*c^7 + a^4*b^12*c - 24*a^5*b^10*c^2 +
240*a^6*b^8*c^3 - 1280*a^7*b^6*c^4 + 3840*a^8*b^4*c^5 - 6144*a^9*b^2*c^6))
)*root(56371445760*a^11*b^8*c^9*z^4 - 503316480*a^8*b^14*c^6*z^4 + 47185920
*a^7*b^16*c^5*z^4 - 2621440*a^6*b^18*c^4*z^4 + 65536*a^5*b^20*c^3*z^4 - 171
798691840*a^14*b^2*c^12*z^4 + 193273528320*a^13*b^4*c^11*z^4 - 128849018880
*a^12*b^6*c^10*z^4 - 16911433728*a^10*b^10*c^8*z^4 + 3523215360*a^9*b^12*c^
7*z^4 + 68719476736*a^15*c^13*z^4 + 1536*a^5*b^16*c*k*m*z^2 + 1536*a*b^18*c
^3*d*f*z^2 - 2571632640*a^9*b^5*c^8*d*m*z^2 + 2548039680*a^9*b^3*c^10*d*h*z
^2 + 1509949440*a^10*b^3*c^9*e*l*z^2 + 1509949440*a^9*b^3*c^10*e*g*z^2 - 14
01421824*a^8*b^5*c^9*d*h*z^2 - 1321205760*a^9*b^2*c^11*d*f*z^2 - 2793406464
*a^11*b*c^10*d*m*z^2 + 890634240*a^8*b^7*c^7*d*m*z^2 - 754974720*a^10*b^4*c
^8*g*l*z^2 - 754974720*a^9*b^5*c^8*e*l*z^2 + 719585280*a^8*b^6*c^8*d*k*z^2
- 707788800*a^9*b^4*c^9*d*k*z^2 - 754974720*a^8*b^5*c^9*e*g*z^2 + 603979776
*a^11*b^2*c^9*g*l*z^2 - 581959680*a^10*b^4*c^8*f*m*z^2 + 732168192*a^7*b^6*
c^9*d*f*z^2 + 534773760*a^11*b^3*c^8*h*m*z^2 - 456130560*a^11*b^4*c^7*k*m*z
^2 - 603979776*a^10*b^2*c^10*e*j*z^2 + 534773760*a^10*b^3*c^9*f*k*z^2 + 384
040960*a^9*b^6*c^7*f*m*z^2 + 377487360*a^9*b^6*c^7*g*l*z^2 - 456130560*a^9*
b^4*c^9*f*h*z^2 + 301989888*a^11*b^3*c^8*j*l*z^2 - 415236096*a^10*b^2*c^10*
d*k*z^2 + 254017536*a^10*b^6*c^6*k*m*z^2 - 330301440*a^10*b^4*c^8*h*k*z^2 +
390463488*a^7*b^7*c^8*d*h*z^2 + 188743680*a^12*b^2*c^8*k*m*z^2 + 301989888

*a^10*b^3*c^9*g*j*z^2 - 297861120*a^7*b^8*c^7*d*k*z^2 - 366280704*a^6*b^8*c
^8*d*f*z^2 + 188743680*a^11*b^2*c^9*h*k*z^2 - 330301440*a^8*b^4*c^10*d*f*z^
2 + 254017536*a^8*b^6*c^8*f*h*z^2 - 1887436800*a^10*b*c^11*d*h*z^2 + 188743
680*a^8*b^7*c^7*e*l*z^2 + 153354240*a^9*b^6*c^7*h*k*z^2 - 185303040*a^7*b^9
*c^6*d*m*z^2 - 117964800*a^10*b^5*c^7*h*m*z^2 - 61931520*a^9*b^8*c^5*k*m*z^
2 + 121634816*a^11*b^2*c^9*f*m*z^2 - 115671040*a^8*b^8*c^6*f*m*z^2 - 629145
60*a^9*b^7*c^6*j*l*z^2 + 188743680*a^10*b^2*c^10*f*h*z^2 - 94371840*a^8*b^8
*c^6*g*l*z^2 + 6144000*a^8*b^10*c^4*k*m*z^2 - 117964800*a^9*b^5*c^8*f*k*z^2
 + 61440*a^7*b^12*c^3*k*m*z^2 - 46080*a^6*b^14*c^2*k*m*z^2 + 23592960*a^8*b
^9*c^5*j*l*z^2 + 188743680*a^7*b^7*c^8*e*g*z^2 - 37355520*a^9*b^7*c^6*h*m*z
^2 + 125829120*a^8*b^6*c^8*e*j*z^2 + 23101440*a^8*b^9*c^5*h*m*z^2 - 3538944
*a^7*b^11*c^4*j*l*z^2 + 196608*a^6*b^13*c^3*j*l*z^2 - 4349952*a^7*b^11*c^4*
h*m*z^2 + 337920*a^6*b^13*c^3*h*m*z^2 - 7680*a^5*b^15*c^2*h*m*z^2 - 6291456
0*a^8*b^7*c^7*g*j*z^2 - 26542080*a^8*b^8*c^6*h*k*z^2 + 17940480*a^7*b^10*c^
5*f*m*z^2 + 11796480*a^7*b^10*c^5*g*l*z^2 - 37355520*a^8*b^7*c^7*f*k*z^2 -
1347584*a^6*b^12*c^4*f*m*z^2 + 68272128*a^6*b^10*c^6*d*k*z^2 - 589824*a^6*b
^12*c^4*g*l*z^2 + 552960*a^6*b^12*c^4*h*k*z^2 - 147456*a^7*b^10*c^5*h*k*z^2
 - 46080*a^5*b^14*c^3*h*k*z^2 + 35840*a^5*b^14*c^3*f*m*z^2 + 23592960*a^7*b
^9*c^6*g*j*z^2 - 23592960*a^7*b^9*c^6*e*l*z^2 + 23371776*a^6*b^11*c^5*d*m*z
^2 + 23101440*a^7*b^9*c^6*f*k*z^2 - 47185920*a^7*b^8*c^7*e*j*z^2 - 61931520
*a^7*b^8*c^7*f*h*z^2 - 4349952*a^6*b^11*c^5*f*k*z^2 - 3538944*a^6*b^11*c^5*
g*j*z^2 - 1677312*a^5*b^13*c^4*d*m*z^2 + 1179648*a^6*b^11*c^5*e*l*z^2 + 337
920*a^5*b^13*c^4*f*k*z^2 + 196608*a^5*b^13*c^4*g*j*z^2 + 53760*a^4*b^15*c^3
*d*m*z^2 - 7680*a^4*b^15*c^3*f*k*z^2 + 96583680*a^5*b^10*c^7*d*f*z^2 - 9179
136*a^5*b^12*c^5*d*k*z^2 + 7077888*a^6*b^10*c^6*e*j*z^2 - 51609600*a^6*b^9*
c^7*d*h*z^2 + 691200*a^4*b^14*c^4*d*k*z^2 - 393216*a^5*b^12*c^5*e*j*z^2 - 2
3040*a^3*b^16*c^3*d*k*z^2 + 6144000*a^6*b^10*c^6*f*h*z^2 + 61440*a^5*b^12*c
^5*f*h*z^2 - 46080*a^4*b^14*c^4*f*h*z^2 + 1536*a^3*b^16*c^3*f*h*z^2 - 23592
960*a^6*b^9*c^7*e*g*z^2 + 1179648*a^5*b^11*c^6*e*g*z^2 + 829440*a^4*b^13*c^
5*d*h*z^2 + 368640*a^5*b^11*c^6*d*h*z^2 - 105984*a^3*b^15*c^4*d*h*z^2 + 460
8*a^2*b^17*c^3*d*h*z^2 - 15175680*a^4*b^12*c^6*d*f*z^2 + 1428480*a^3*b^14*c
^5*d*f*z^2 - 73728*a^2*b^16*c^4*d*f*z^2 + 4108320768*a^10*b^3*c^9*d*m*z^2 -
 1207959552*a^11*b*c^10*e*l*z^2 - 1207959552*a^10*b*c^11*e*g*z^2 - 57881395
2*a^12*b*c^9*h*m*z^2 - 578813952*a^11*b*c^10*f*k*z^2 - 402653184*a^12*b*c^9
*j*l*z^2 - 402653184*a^11*b*c^10*g*j*z^2 - 440401920*a^10*b*c^11*f^2*z^2 -
188743680*a^12*b*c^9*k^2*z^2 - 188743680*a^11*b*c^10*h^2*z^2 + 1761607680*a
^10*c^12*d*f*z^2 - 14080*a^6*b^15*c*m^2*z^2 - 94464*a*b^17*c^4*d^2*z^2 + 69
36330240*a^8*b^3*c^11*d^2*z^2 + 2464874496*a^6*b^7*c^9*d^2*z^2 - 3963617280
*a^9*b*c^12*d^2*z^2 + 1056964608*a^11*c^11*d*k*z^2 + 805306368*a^11*c^11*e*
j*z^2 + 419430400*a^12*c^10*f*m*z^2 + 251658240*a^13*c^9*k*m*z^2 - 15099494
40*a^9*b^2*c^11*e^2*z^2 + 251658240*a^11*c^11*f*h*z^2 + 150994944*a^12*c^10
*h*k*z^2 - 5400428544*a^7*b^5*c^10*d^2*z^2 + 754974720*a^8*b^4*c^10*e^2*z^2
 - 730054656*a^5*b^9*c^8*d^2*z^2 + 477102080*a^12*b^3*c^7*m^2*z^2 - 3774873
60*a^11*b^4*c^7*l^2*z^2 + 477102080*a^9*b^3*c^10*f^2*z^2 + 301989888*a^12*b
^2*c^8*l^2*z^2 - 377487360*a^9*b^4*c^9*g^2*z^2 + 301989888*a^10*b^2*c^10*g^

2*z^2 - 174325760*a^11*b^5*c^6*m^2*z^2 + 188743680*a^10*b^6*c^6*l^2*z^2 + 1
41557760*a^11*b^3*c^8*k^2*z^2 + 188743680*a^8*b^6*c^8*g^2*z^2 + 141557760*a
^10*b^3*c^9*h^2*z^2 - 174325760*a^8*b^5*c^9*f^2*z^2 - 188743680*a^7*b^6*c^9
*e^2*z^2 - 47185920*a^9*b^8*c^5*l^2*z^2 + 11206656*a^10*b^7*c^5*m^2*z^2 + 8
929280*a^9*b^9*c^4*m^2*z^2 - 2600960*a^8*b^11*c^3*m^2*z^2 + 291840*a^7*b^13
*c^2*m^2*z^2 - 50331648*a^10*b^4*c^8*j^2*z^2 + 146165760*a^4*b^11*c^7*d^2*z
^2 - 26542080*a^9*b^7*c^6*k^2*z^2 + 5898240*a^8*b^10*c^4*l^2*z^2 - 294912*a
^7*b^12*c^3*l^2*z^2 - 33554432*a^11*b^2*c^9*j^2*z^2 + 9584640*a^8*b^9*c^5*k
^2*z^2 + 20971520*a^9*b^6*c^7*j^2*z^2 - 2359296*a^10*b^5*c^7*k^2*z^2 - 1290
240*a^7*b^11*c^4*k^2*z^2 + 46080*a^6*b^13*c^3*k^2*z^2 + 2304*a^5*b^15*c^2*k
^2*z^2 - 2752512*a^7*b^10*c^5*j^2*z^2 + 2621440*a^8*b^8*c^6*j^2*z^2 + 52428
8*a^6*b^12*c^4*j^2*z^2 - 32768*a^5*b^14*c^3*j^2*z^2 - 47185920*a^7*b^8*c^7*
g^2*z^2 - 26542080*a^8*b^7*c^7*h^2*z^2 + 9584640*a^7*b^9*c^6*h^2*z^2 - 2359
296*a^9*b^5*c^8*h^2*z^2 - 1290240*a^6*b^11*c^5*h^2*z^2 + 46080*a^5*b^13*c^4
*h^2*z^2 + 2304*a^4*b^15*c^3*h^2*z^2 + 5898240*a^6*b^10*c^6*g^2*z^2 - 29491
2*a^5*b^12*c^5*g^2*z^2 + 11206656*a^7*b^7*c^8*f^2*z^2 + 8929280*a^6*b^9*c^7
*f^2*z^2 + 23592960*a^6*b^8*c^8*e^2*z^2 - 2600960*a^5*b^11*c^6*f^2*z^2 + 29
1840*a^4*b^13*c^5*f^2*z^2 - 14080*a^3*b^15*c^4*f^2*z^2 + 256*a^2*b^17*c^3*f
^2*z^2 - 19860480*a^3*b^13*c^6*d^2*z^2 - 1179648*a^5*b^10*c^7*e^2*z^2 + 177
1776*a^2*b^15*c^5*d^2*z^2 - 440401920*a^13*b*c^8*m^2*z^2 + 1207959552*a^10*
c^12*e^2*z^2 + 134217728*a^12*c^10*j^2*z^2 + 256*a^5*b^17*m^2*z^2 + 2304*b^
19*c^3*d^2*z^2 - 23592960*a^10*b*c^8*f*k*l*z + 99090432*a^9*b*c^9*d*h*l*z +
 9437184*a^10*b*c^8*e*k*m*z + 23592960*a^10*b*c^8*g*h*m*z + 141557760*a^8*b
*c^10*d*e*k*z + 47185920*a^9*b*c^9*d*j*k*z - 23592960*a^9*b*c^9*f*g*k*z + 1
69869312*a^7*b*c^11*d*e*f*z + 99090432*a^8*b*c^10*d*g*h*z - 3145728*a^9*b*c
^9*f*h*j*z + 56623104*a^8*b*c^10*d*f*j*z + 1536*a*b^15*c^3*d*f*j*z - 943718
4*a^8*b*c^10*e*f*h*z - 4608*a*b^14*c^4*d*f*g*z + 9216*a*b^13*c^5*d*e*f*z +
412876800*a^8*b^2*c^9*d*e*m*z - 206438400*a^9*b^3*c^7*d*l*m*z + 5898240*a^1
0*b^4*c^5*k*l*m*z - 206438400*a^8*b^3*c^8*d*g*m*z - 4718592*a^11*b^2*c^6*k*
l*m*z - 2949120*a^9*b^6*c^4*k*l*m*z + 737280*a^8*b^8*c^3*k*l*m*z - 92160*a^
7*b^10*c^2*k*l*m*z + 103219200*a^8*b^5*c^6*d*l*m*z - 29491200*a^10*b^3*c^6*
h*l*m*z - 206438400*a^7*b^4*c^8*d*e*m*z - 2359296*a^10*b^3*c^6*j*k*m*z + 49
1520*a^8*b^7*c^4*j*k*m*z - 184320*a^7*b^9*c^3*j*k*m*z + 27648*a^6*b^11*c^2*
j*k*m*z + 14745600*a^9*b^5*c^5*h*l*m*z - 3686400*a^8*b^7*c^4*h*l*m*z + 4608
00*a^7*b^9*c^3*h*l*m*z - 23040*a^6*b^11*c^2*h*l*m*z + 88473600*a^8*b^4*c^7*
d*k*l*z + 82575360*a^9*b^2*c^8*d*j*m*z + 11796480*a^10*b^2*c^7*h*j*m*z + 58
98240*a^9*b^4*c^6*g*k*m*z - 4718592*a^10*b^2*c^7*g*k*m*z - 70778880*a^9*b^2
*c^8*d*k*l*z - 2949120*a^8*b^6*c^5*g*k*m*z - 2457600*a^8*b^6*c^5*h*j*m*z +
921600*a^7*b^8*c^4*h*j*m*z + 737280*a^7*b^8*c^4*g*k*m*z - 138240*a^6*b^10*c
^3*h*j*m*z - 92160*a^6*b^10*c^3*g*k*m*z + 7680*a^5*b^12*c^2*h*j*m*z + 4608*
a^5*b^12*c^2*g*k*m*z + 29491200*a^9*b^3*c^7*f*k*l*z - 176947200*a^7*b^3*c^9
*d*e*k*z - 109707264*a^8*b^3*c^8*d*h*l*z - 25804800*a^7*b^7*c^5*d*l*m*z + 1
03219200*a^7*b^5*c^7*d*g*m*z + 219414528*a^7*b^2*c^10*d*e*h*z - 14745600*a^
8*b^5*c^6*f*k*l*z - 29491200*a^9*b^3*c^7*g*h*m*z - 11796480*a^9*b^3*c^7*e*k
*m*z - 44236800*a^7*b^6*c^6*d*k*l*z + 58982400*a^9*b^2*c^8*e*h*m*z + 589824

0*a^8*b^5*c^6*e*k*m*z + 3686400*a^7*b^7*c^5*f*k*l*z + 3225600*a^6*b^9*c^4*d
*l*m*z - 1474560*a^7*b^7*c^5*e*k*m*z - 460800*a^6*b^9*c^4*f*k*l*z + 184320*
a^6*b^9*c^4*e*k*m*z - 161280*a^5*b^11*c^3*d*l*m*z + 23040*a^5*b^11*c^3*f*k*
l*z - 9216*a^5*b^11*c^3*e*k*m*z + 14745600*a^8*b^5*c^6*g*h*m*z + 110886912*
a^7*b^4*c^8*d*f*l*z - 3686400*a^7*b^7*c^5*g*h*m*z - 221773824*a^6*b^3*c^10*
d*e*f*z + 460800*a^6*b^9*c^4*g*h*m*z - 17203200*a^7*b^6*c^6*d*j*m*z - 23040
*a^5*b^11*c^3*g*h*m*z - 29491200*a^8*b^4*c^7*e*h*m*z - 11796480*a^9*b^2*c^8
*f*j*k*z + 11059200*a^6*b^8*c^5*d*k*l*z + 6451200*a^6*b^8*c^5*d*j*m*z + 884
73600*a^7*b^4*c^8*d*g*k*z + 2457600*a^7*b^6*c^6*f*j*k*z - 35389440*a^8*b^3*
c^8*d*j*k*z - 1382400*a^5*b^10*c^4*d*k*l*z - 84934656*a^8*b^2*c^9*d*f*l*z -
967680*a^5*b^10*c^4*d*j*m*z - 921600*a^6*b^8*c^5*f*j*k*z + 138240*a^5*b^10
*c^4*f*j*k*z + 69120*a^4*b^12*c^3*d*k*l*z + 53760*a^4*b^12*c^3*d*j*m*z - 76
80*a^4*b^12*c^3*f*j*k*z + 44236800*a^7*b^5*c^7*d*h*l*z + 7372800*a^7*b^6*c^
6*e*h*m*z - 5898240*a^8*b^4*c^7*f*h*l*z + 4718592*a^9*b^2*c^8*f*h*l*z - 707
78880*a^8*b^2*c^9*d*g*k*z + 2949120*a^7*b^6*c^6*f*h*l*z - 921600*a^6*b^8*c^
5*e*h*m*z - 737280*a^6*b^8*c^5*f*h*l*z + 92160*a^5*b^10*c^4*f*h*l*z + 46080
*a^5*b^10*c^4*e*h*m*z - 4608*a^4*b^12*c^3*f*h*l*z + 29491200*a^8*b^3*c^8*f*
g*k*z - 109707264*a^7*b^3*c^9*d*g*h*z - 25804800*a^6*b^7*c^6*d*g*m*z - 5898
2400*a^8*b^2*c^9*e*f*k*z - 58982400*a^6*b^6*c^7*d*f*l*z + 7372800*a^6*b^7*c
^6*d*j*k*z + 88473600*a^6*b^5*c^8*d*e*k*z - 2764800*a^5*b^9*c^5*d*j*k*z + 5
1609600*a^6*b^6*c^7*d*e*m*z + 414720*a^4*b^11*c^4*d*j*k*z - 23040*a^3*b^13*
c^3*d*j*k*z - 14745600*a^7*b^5*c^7*f*g*k*z - 44236800*a^6*b^6*c^7*d*g*k*z -
6635520*a^6*b^7*c^6*d*h*l*z + 40108032*a^8*b^2*c^9*d*h*j*z + 3686400*a^6*b
^7*c^6*f*g*k*z + 3225600*a^5*b^9*c^5*d*g*m*z + 2359296*a^8*b^3*c^8*f*h*j*z
- 491520*a^6*b^7*c^6*f*h*j*z - 460800*a^5*b^9*c^5*f*g*k*z - 276480*a^5*b^9*
c^5*d*h*l*z + 184320*a^5*b^9*c^5*f*h*j*z + 179712*a^4*b^11*c^4*d*h*l*z - 16
1280*a^4*b^11*c^4*d*g*m*z - 27648*a^4*b^11*c^4*f*h*j*z + 23040*a^4*b^11*c^4
*f*g*k*z - 13824*a^3*b^13*c^3*d*h*l*z + 1536*a^3*b^13*c^3*f*h*j*z + 2949120
0*a^7*b^4*c^8*e*f*k*z + 110886912*a^6*b^4*c^9*d*f*g*z + 16220160*a^5*b^8*c^
6*d*f*l*z - 45613056*a^7*b^3*c^9*d*f*j*z + 11059200*a^5*b^8*c^6*d*g*k*z - 1
0321920*a^6*b^6*c^7*d*h*j*z - 7372800*a^6*b^6*c^7*e*f*k*z + 7077888*a^7*b^4
*c^8*d*h*j*z - 6451200*a^5*b^8*c^6*d*e*m*z - 88473600*a^6*b^4*c^9*d*e*h*z +
2396160*a^5*b^8*c^6*d*h*j*z - 2396160*a^4*b^10*c^5*d*f*l*z - 1382400*a^4*b
^10*c^5*d*g*k*z - 84934656*a^7*b^2*c^10*d*f*g*z + 921600*a^5*b^8*c^6*e*f*k*
z + 117964800*a^5*b^5*c^9*d*e*f*z + 322560*a^4*b^10*c^5*d*e*m*z + 175104*a^
3*b^12*c^4*d*f*l*z + 69120*a^3*b^12*c^4*d*g*k*z - 50688*a^3*b^12*c^4*d*h*j*
z - 46080*a^4*b^10*c^5*e*f*k*z - 27648*a^4*b^10*c^5*d*h*j*z + 4608*a^2*b^14
*c^3*d*h*j*z - 4608*a^2*b^14*c^3*d*f*l*z + 44236800*a^6*b^5*c^8*d*g*h*z - 5
898240*a^7*b^4*c^8*f*g*h*z - 22118400*a^5*b^7*c^7*d*e*k*z + 4718592*a^8*b^2
*c^9*f*g*h*z + 2949120*a^6*b^6*c^7*f*g*h*z - 737280*a^5*b^8*c^6*f*g*h*z + 9
2160*a^4*b^10*c^5*f*g*h*z - 4608*a^3*b^12*c^4*f*g*h*z + 8847360*a^5*b^7*c^7
*d*f*j*z - 58982400*a^5*b^6*c^8*d*f*g*z - 3809280*a^4*b^9*c^6*d*f*j*z + 276
4800*a^4*b^9*c^6*d*e*k*z + 2359296*a^6*b^5*c^8*d*f*j*z + 681984*a^3*b^11*c^
5*d*f*j*z - 138240*a^3*b^11*c^5*d*e*k*z - 55296*a^2*b^13*c^4*d*f*j*z + 1179
6480*a^7*b^3*c^9*e*f*h*z - 6635520*a^5*b^7*c^7*d*g*h*z - 5898240*a^6*b^5*c^

8*e*f*h*z + 1474560*a^5*b^7*c^7*e*f*h*z - 276480*a^4*b^9*c^6*d*g*h*z - 1843
20*a^4*b^9*c^6*e*f*h*z + 179712*a^3*b^11*c^5*d*g*h*z - 13824*a^2*b^13*c^4*d
*g*h*z + 9216*a^3*b^11*c^5*e*f*h*z + 16220160*a^4*b^8*c^7*d*f*g*z + 1327104
0*a^5*b^6*c^8*d*e*h*z - 2396160*a^3*b^10*c^6*d*f*g*z + 552960*a^4*b^8*c^7*d
*e*h*z - 359424*a^3*b^10*c^6*d*e*h*z + 175104*a^2*b^12*c^5*d*f*g*z + 27648*
a^2*b^12*c^5*d*e*h*z - 32440320*a^4*b^7*c^8*d*e*f*z + 4792320*a^3*b^9*c^7*d
*e*f*z - 350208*a^2*b^11*c^6*d*e*f*z + 165150720*a^10*b*c^8*d*l*m*z + 4608*
a^6*b^12*c*k*l*m*z + 23592960*a^11*b*c^7*h*l*m*z + 3145728*a^11*b*c^7*j*k*m
*z - 1536*a^5*b^13*c*j*k*m*z + 165150720*a^9*b*c^9*d*g*m*z + 346816512*a^7*
b*c^11*d^2*g*z + 19660800*a^12*b*c^6*l*m^2*z - 34560*a^7*b^11*c*l*m^2*z - 7
077888*a^11*b*c^7*k^2*l*z + 11008*a^6*b^12*c*j*m^2*z + 19660800*a^11*b*c^7*
g*m^2*z + 7077888*a^10*b*c^8*h^2*l*z + 768*a^5*b^13*c*g*m^2*z - 19660800*a^
9*b*c^9*f^2*l*z - 7077888*a^10*b*c^8*g*k^2*z - 6912*a*b^15*c^3*d^2*l*z + 70
77888*a^9*b*c^9*g*h^2*z - 19660800*a^8*b*c^10*f^2*g*z - 66816*a*b^14*c^4*d^
2*j*z + 214272*a*b^13*c^5*d^2*g*z - 428544*a*b^12*c^6*d^2*e*z - 330301440*a
^9*c^10*d*e*m*z - 110100480*a^10*c^9*d*j*m*z - 15728640*a^11*c^8*h*j*m*z -
47185920*a^10*c^9*e*h*m*z - 198180864*a^8*c^11*d*e*h*z + 15728640*a^10*c^9*
f*j*k*z - 66060288*a^9*c^10*d*h*j*z + 47185920*a^9*c^10*e*f*k*z + 102275481
6*a^6*b^2*c^11*d^2*e*z - 642318336*a^5*b^4*c^10*d^2*e*z - 511377408*a^7*b^3
*c^9*d^2*l*z - 511377408*a^6*b^3*c^10*d^2*g*z + 321159168*a^6*b^5*c^8*d^2*l
*z + 321159168*a^5*b^5*c^9*d^2*g*z + 225312768*a^7*b^2*c^10*d^2*j*z - 25362
432*a^11*b^3*c^5*l*m^2*z + 13271040*a^10*b^5*c^4*l*m^2*z - 3563520*a^9*b^7*
c^3*l*m^2*z + 506880*a^8*b^9*c^2*l*m^2*z + 10354688*a^11*b^2*c^6*j*m^2*z +
8847360*a^10*b^3*c^6*k^2*l*z - 4423680*a^9*b^5*c^5*k^2*l*z - 2048000*a^9*b^
6*c^4*j*m^2*z + 1105920*a^8*b^7*c^4*k^2*l*z + 849920*a^8*b^8*c^3*j*m^2*z -
393216*a^10*b^4*c^5*j*m^2*z - 145920*a^7*b^10*c^2*j*m^2*z - 138240*a^7*b^9*
c^3*k^2*l*z + 6912*a^6*b^11*c^2*k^2*l*z - 111697920*a^5*b^7*c^7*d^2*l*z + 2
23395840*a^4*b^6*c^9*d^2*e*z - 25362432*a^10*b^3*c^6*g*m^2*z - 3538944*a^10
*b^2*c^7*j*k^2*z + 737280*a^8*b^6*c^5*j*k^2*z + 50724864*a^10*b^2*c^7*e*m^2
*z - 276480*a^7*b^8*c^4*j*k^2*z + 41472*a^6*b^10*c^3*j*k^2*z - 2304*a^5*b^1
2*c^2*j*k^2*z + 13271040*a^9*b^5*c^5*g*m^2*z - 8847360*a^9*b^3*c^7*h^2*l*z
+ 4423680*a^8*b^5*c^6*h^2*l*z - 3563520*a^8*b^7*c^4*g*m^2*z - 1105920*a^7*b
^7*c^5*h^2*l*z + 506880*a^7*b^9*c^3*g*m^2*z + 138240*a^6*b^9*c^4*h^2*l*z -
34560*a^6*b^11*c^2*g*m^2*z - 6912*a^5*b^11*c^3*h^2*l*z - 26542080*a^9*b^4*c
^6*e*m^2*z + 25362432*a^8*b^3*c^8*f^2*l*z - 13271040*a^7*b^5*c^7*f^2*l*z +
8847360*a^9*b^3*c^7*g*k^2*z + 7127040*a^8*b^6*c^5*e*m^2*z - 4423680*a^8*b^5
*c^6*g*k^2*z + 3563520*a^6*b^7*c^6*f^2*l*z + 3538944*a^9*b^2*c^8*h^2*j*z +
1105920*a^7*b^7*c^5*g*k^2*z - 1013760*a^7*b^8*c^4*e*m^2*z - 737280*a^7*b^6*
c^6*h^2*j*z - 506880*a^5*b^9*c^5*f^2*l*z + 276480*a^6*b^8*c^5*h^2*j*z - 138
240*a^6*b^9*c^4*g*k^2*z + 69120*a^6*b^10*c^3*e*m^2*z - 41472*a^5*b^10*c^4*h
^2*j*z + 34560*a^4*b^11*c^4*f^2*l*z + 6912*a^5*b^11*c^3*g*k^2*z + 2304*a^4*
b^12*c^3*h^2*j*z - 1536*a^5*b^12*c^2*e*m^2*z - 768*a^3*b^13*c^3*f^2*l*z - 1
11697920*a^4*b^7*c^8*d^2*g*z + 23362560*a^4*b^9*c^6*d^2*l*z - 17694720*a^9*
b^2*c^8*e*k^2*z - 10354688*a^8*b^2*c^9*f^2*j*z - 43646976*a^6*b^4*c^9*d^2*j
*z + 8847360*a^8*b^4*c^7*e*k^2*z - 2965248*a^3*b^11*c^5*d^2*l*z - 2211840*a

^7*b^6*c^6*e*k^2*z + 2048000*a^6*b^6*c^7*f^2*j*z - 849920*a^5*b^8*c^6*f^2*j
*z + 393216*a^7*b^4*c^8*f^2*j*z + 276480*a^6*b^8*c^5*e*k^2*z + 214272*a^2*b
^13*c^4*d^2*l*z + 145920*a^4*b^10*c^5*f^2*j*z - 13824*a^5*b^10*c^4*e*k^2*z
- 11008*a^3*b^12*c^4*f^2*j*z + 256*a^2*b^14*c^3*f^2*j*z - 32587776*a^5*b^6*
c^8*d^2*j*z - 8847360*a^8*b^3*c^8*g*h^2*z + 21657600*a^4*b^8*c^7*d^2*j*z +
4423680*a^7*b^5*c^7*g*h^2*z - 1105920*a^6*b^7*c^6*g*h^2*z + 138240*a^5*b^9*
c^5*g*h^2*z - 6912*a^4*b^11*c^4*g*h^2*z + 25362432*a^7*b^3*c^9*f^2*g*z - 58
10688*a^3*b^10*c^6*d^2*j*z + 17694720*a^8*b^2*c^9*e*h^2*z + 845568*a^2*b^12
*c^5*d^2*j*z - 50724864*a^7*b^2*c^10*e*f^2*z - 13271040*a^6*b^5*c^8*f^2*g*z
- 8847360*a^7*b^4*c^8*e*h^2*z + 3563520*a^5*b^7*c^7*f^2*g*z + 2211840*a^6*
b^6*c^7*e*h^2*z - 506880*a^4*b^9*c^6*f^2*g*z - 276480*a^5*b^8*c^6*e*h^2*z +
34560*a^3*b^11*c^5*f^2*g*z + 13824*a^4*b^10*c^5*e*h^2*z - 768*a^2*b^13*c^4
*f^2*g*z + 26542080*a^6*b^4*c^9*e*f^2*z + 23362560*a^3*b^9*c^7*d^2*g*z - 46
725120*a^3*b^8*c^8*d^2*e*z - 7127040*a^5*b^6*c^8*e*f^2*z - 2965248*a^2*b^11
*c^6*d^2*g*z + 1013760*a^4*b^8*c^7*e*f^2*z - 69120*a^3*b^10*c^6*e*f^2*z + 1
536*a^2*b^12*c^5*e*f^2*z + 5930496*a^2*b^10*c^7*d^2*e*z + 346816512*a^8*b*c
^10*d^2*l*z - 693633024*a^7*c^12*d^2*e*z - 231211008*a^8*c^11*d^2*j*z + 768
*a^6*b^13*l*m^2*z - 13107200*a^12*c^7*j*m^2*z - 256*a^5*b^14*j*m^2*z + 4718
592*a^11*c^8*j*k^2*z - 39321600*a^11*c^8*e*m^2*z - 4718592*a^10*c^9*h^2*j*z
+ 14155776*a^10*c^9*e*k^2*z + 13107200*a^9*c^10*f^2*j*z + 2304*b^16*c^3*d^
2*j*z - 14155776*a^9*c^10*e*h^2*z + 39321600*a^8*c^11*e*f^2*z - 6912*b^15*c
^4*d^2*g*z + 13824*b^14*c^5*d^2*e*z + 737280*a^10*b*c^5*j*k*l*m - 2304*a^6*
b^9*c*j*k*l*m + 2211840*a^9*b*c^6*e*k*l*m + 1228800*a^9*b*c^6*f*j*l*m + 737
280*a^9*b*c^6*g*j*k*m + 442368*a^9*b*c^6*h*j*k*l + 36*a^3*b^12*c*f*h*k*m +
3096576*a^8*b*c^7*d*j*k*l - 12745728*a^8*b*c^7*d*h*k*m + 3686400*a^8*b*c^7*
e*f*l*m + 3391488*a^8*b*c^7*e*h*j*m + 2211840*a^8*b*c^7*e*g*k*m + 1327104*a
^8*b*c^7*e*h*k*l + 1228800*a^8*b*c^7*f*g*j*m + 737280*a^8*b*c^7*f*h*j*l + 4
42368*a^8*b*c^7*g*h*j*k + 108*a^2*b^13*c*d*h*k*m + 16367616*a^7*b*c^8*d*e*j
*m + 9289728*a^7*b*c^8*d*e*k*l + 5160960*a^7*b*c^8*d*f*j*l + 3391488*a^7*b*
c^8*e*f*j*k + 3096576*a^7*b*c^8*d*g*j*k - 19307520*a^7*b*c^8*d*f*h*m + 3686
400*a^7*b*c^8*e*f*g*m + 2211840*a^7*b*c^8*e*f*h*l + 1327104*a^7*b*c^8*e*g*h
*k + 737280*a^7*b*c^8*f*g*h*j - 180*a*b^13*c^2*d*f*h*m - 540*a*b^12*c^3*d*f
*h*k + 15482880*a^6*b*c^9*d*e*f*l + 11059200*a^6*b*c^9*d*e*h*j + 9289728*a^
6*b*c^9*d*e*g*k + 5160960*a^6*b*c^9*d*f*g*j - 2304*a*b^11*c^4*d*f*g*j + 221
1840*a^6*b*c^9*e*f*g*h + 4608*a*b^10*c^5*d*e*f*j + 15482880*a^5*b*c^10*d*e*
f*g - 13824*a*b^9*c^6*d*e*f*g + 36*a*b^14*c*d*f*k*m + 1843200*a^9*b^3*c^4*j
*k*l*m + 783360*a^8*b^5*c^3*j*k*l*m + 18432*a^7*b^7*c^2*j*k*l*m - 2211840*a
^8*b^4*c^4*g*k*l*m - 1695744*a^9*b^2*c^5*h*j*l*m - 1400832*a^8*b^4*c^4*h*j*
l*m - 1105920*a^9*b^2*c^5*g*k*l*m - 253440*a^7*b^6*c^3*h*j*l*m - 69120*a^7*
b^6*c^3*g*k*l*m + 11520*a^6*b^8*c^2*h*j*l*m + 6912*a^6*b^8*c^2*g*k*l*m + 44
23680*a^8*b^3*c^5*e*k*l*m + 2506752*a^8*b^3*c^5*f*j*l*m + 1843200*a^8*b^3*c
^5*g*j*k*m + 1327104*a^8*b^3*c^5*h*j*k*l + 838656*a^7*b^5*c^4*f*j*l*m + 783
360*a^7*b^5*c^4*g*j*k*m + 691200*a^7*b^5*c^4*h*j*k*l + 138240*a^7*b^5*c^4*e
*k*l*m + 69120*a^6*b^7*c^3*h*j*k*l - 53760*a^6*b^7*c^3*f*j*l*m + 18432*a^6*
b^7*c^3*g*j*k*m - 13824*a^6*b^7*c^3*e*k*l*m - 2304*a^5*b^9*c^2*g*j*k*m + 25

43616*a^8*b^3*c^5*g*h*l*m + 829440*a^7*b^5*c^4*g*h*l*m - 34560*a^6*b^7*c^3*
g*h*l*m - 8183808*a^8*b^2*c^6*d*j*l*m - 3686400*a^8*b^2*c^6*e*j*k*m - 22855
68*a^7*b^4*c^5*d*j*l*m - 1695744*a^8*b^2*c^6*f*j*k*l - 1566720*a^7*b^4*c^5*
e*j*k*m - 1400832*a^7*b^4*c^5*f*j*k*l + 741888*a^6*b^6*c^4*d*j*l*m - 253440
*a^6*b^6*c^4*f*j*k*l - 80640*a^5*b^8*c^3*d*j*l*m - 36864*a^6*b^6*c^4*e*j*k*
m + 11520*a^5*b^8*c^3*f*j*k*l + 4608*a^5*b^8*c^3*e*j*k*m + 6700032*a^8*b^2*
c^6*f*h*k*m + 5103360*a^7*b^4*c^5*f*h*k*m - 5087232*a^8*b^2*c^6*e*h*l*m - 2
838528*a^7*b^4*c^5*f*g*l*m - 1843200*a^8*b^2*c^6*f*g*l*m - 1695744*a^8*b^2*
c^6*g*h*j*m - 1658880*a^7*b^4*c^5*g*h*k*l - 1658880*a^7*b^4*c^5*e*h*l*m - 1
400832*a^7*b^4*c^5*g*h*j*m - 663552*a^8*b^2*c^6*g*h*k*l + 483840*a^6*b^6*c^
4*f*h*k*m - 253440*a^6*b^6*c^4*g*h*j*m - 207360*a^6*b^6*c^4*g*h*k*l + 16128
0*a^6*b^6*c^4*f*g*l*m + 69120*a^6*b^6*c^4*e*h*l*m - 50040*a^5*b^8*c^3*f*h*k
*m + 11520*a^5*b^8*c^3*g*h*j*m + 180*a^4*b^10*c^2*f*h*k*m + 4202496*a^7*b^3
*c^6*d*j*k*l + 635904*a^6*b^5*c^5*d*j*k*l - 276480*a^5*b^7*c^4*d*j*k*l + 34
560*a^4*b^9*c^3*d*j*k*l - 16671744*a^7*b^3*c^6*d*h*k*m + 12275712*a^7*b^3*c
^6*d*g*l*m + 5677056*a^7*b^3*c^6*e*f*l*m + 4423680*a^7*b^3*c^6*e*g*k*m + 33
17760*a^7*b^3*c^6*e*h*k*l + 2801664*a^7*b^3*c^6*e*h*j*m - 2709504*a^6*b^5*c
^5*d*g*l*m + 2543616*a^7*b^3*c^6*f*g*k*l + 2506752*a^7*b^3*c^6*f*g*j*m + 18
43200*a^7*b^3*c^6*f*h*j*l + 1327104*a^7*b^3*c^6*g*h*j*k + 838656*a^6*b^5*c^
5*f*g*j*m + 829440*a^6*b^5*c^5*f*g*k*l + 783360*a^6*b^5*c^5*f*h*j*l + 69120
0*a^6*b^5*c^5*g*h*j*k + 665280*a^5*b^7*c^4*d*h*k*m + 506880*a^6*b^5*c^5*e*h
*j*m + 414720*a^6*b^5*c^5*e*h*k*l - 322560*a^6*b^5*c^5*e*f*l*m + 241920*a^5
*b^7*c^4*d*g*l*m + 138240*a^6*b^5*c^5*e*g*k*m - 108540*a^4*b^9*c^3*d*h*k*m
+ 69120*a^5*b^7*c^4*g*h*j*k - 53760*a^5*b^7*c^4*f*g*j*m - 51840*a^6*b^5*c^5
*d*h*k*m - 34560*a^5*b^7*c^4*f*g*k*l - 23040*a^5*b^7*c^4*e*h*j*m + 18432*a^
5*b^7*c^4*f*h*j*l - 13824*a^5*b^7*c^4*e*g*k*m - 2304*a^4*b^9*c^3*f*h*j*l +
1296*a^3*b^11*c^2*d*h*k*m + 31924224*a^7*b^2*c^7*d*f*k*m - 24551424*a^7*b^2
*c^7*d*e*l*m + 10616832*a^7*b^2*c^7*e*g*j*l - 8183808*a^7*b^2*c^7*d*g*j*m -
5529600*a^7*b^2*c^7*d*h*j*l + 5419008*a^6*b^4*c^6*d*e*l*m + 5308416*a^6*b^
4*c^6*e*g*j*l - 5087232*a^7*b^2*c^7*e*f*k*l - 5013504*a^7*b^2*c^7*e*f*j*m +
4868352*a^6*b^4*c^6*d*f*k*m - 4644864*a^7*b^2*c^7*d*g*k*l - 3981312*a^6*b^
4*c^6*d*g*k*l - 2654208*a^7*b^2*c^7*e*h*j*k - 2367360*a^5*b^6*c^5*d*f*k*m -
2285568*a^6*b^4*c^6*d*g*j*m - 2211840*a^6*b^4*c^6*d*h*j*l - 1695744*a^7*b^
2*c^7*f*g*j*k - 1677312*a^6*b^4*c^6*e*f*j*m - 1658880*a^6*b^4*c^6*e*f*k*l -
1400832*a^6*b^4*c^6*f*g*j*k - 1382400*a^6*b^4*c^6*e*h*j*k + 1036800*a^5*b^
6*c^5*d*g*k*l + 741888*a^5*b^6*c^5*d*g*j*m - 483840*a^5*b^6*c^5*d*e*l*m + 3
17952*a^5*b^6*c^5*d*h*j*l + 268920*a^4*b^8*c^4*d*f*k*m - 253440*a^5*b^6*c^5
*f*g*j*k - 138240*a^5*b^6*c^5*e*h*j*k + 107520*a^5*b^6*c^5*e*f*j*m - 103680
*a^4*b^8*c^4*d*g*k*l - 80640*a^4*b^8*c^4*d*g*j*m + 69120*a^5*b^6*c^5*e*f*k*
l + 11520*a^4*b^8*c^4*f*g*j*k + 6912*a^4*b^8*c^4*d*h*j*l - 6912*a^3*b^10*c^
3*d*h*j*l + 6120*a^3*b^10*c^3*d*f*k*m - 1368*a^2*b^12*c^2*d*f*k*m - 5087232
*a^7*b^2*c^7*e*g*h*m - 2211840*a^6*b^4*c^6*f*g*h*l - 1658880*a^6*b^4*c^6*e*
g*h*m - 1105920*a^7*b^2*c^7*f*g*h*l - 69120*a^5*b^6*c^5*f*g*h*l + 69120*a^5
*b^6*c^5*e*g*h*m + 6912*a^4*b^8*c^4*f*g*h*l + 7962624*a^6*b^3*c^7*d*e*k*l -
22164480*a^6*b^3*c^7*d*f*h*m + 5160960*a^6*b^3*c^7*d*f*j*l + 4571136*a^6*b

^3*c^7*d*e*j*m + 4202496*a^6*b^3*c^7*d*g*j*k + 2801664*a^6*b^3*c^7*e*f*j*k
- 2073600*a^5*b^5*c^6*d*e*k*l - 1483776*a^5*b^5*c^6*d*e*j*m + 635904*a^5*b^
5*c^6*d*g*j*k + 506880*a^5*b^5*c^6*e*f*j*k - 354816*a^4*b^7*c^5*d*f*j*l + 3
22560*a^5*b^5*c^6*d*f*j*l - 276480*a^4*b^7*c^5*d*g*j*k + 207360*a^4*b^7*c^5
*d*e*k*l + 161280*a^4*b^7*c^5*d*e*j*m + 59904*a^3*b^9*c^4*d*f*j*l + 34560*a
^3*b^9*c^4*d*g*j*k - 23040*a^4*b^7*c^5*e*f*j*k - 2304*a^2*b^11*c^3*d*f*j*l
+ 8294400*a^6*b^3*c^7*d*g*h*l + 5677056*a^6*b^3*c^7*e*f*g*m + 4423680*a^6*b
^3*c^7*e*f*h*l + 3317760*a^6*b^3*c^7*e*g*h*k + 2805120*a^5*b^5*c^6*d*f*h*m
+ 1843200*a^6*b^3*c^7*f*g*h*j - 829440*a^5*b^5*c^6*d*g*h*l + 783360*a^5*b^5
*c^6*f*g*h*j + 437184*a^4*b^7*c^5*d*f*h*m + 414720*a^5*b^5*c^6*e*g*h*k - 32
2560*a^5*b^5*c^6*e*f*g*m - 146268*a^3*b^9*c^4*d*f*h*m + 138240*a^5*b^5*c^6*
e*f*h*l - 62208*a^4*b^7*c^5*d*g*h*l + 20736*a^3*b^9*c^4*d*g*h*l + 18432*a^4
*b^7*c^5*f*g*h*j - 13824*a^4*b^7*c^5*e*f*h*l + 9360*a^2*b^11*c^3*d*f*h*m -
2304*a^3*b^9*c^4*f*g*h*j - 8404992*a^6*b^2*c^8*d*e*j*k - 24551424*a^6*b^2*c
^8*d*e*g*m + 21150720*a^6*b^2*c^8*d*f*h*k - 1271808*a^5*b^4*c^7*d*e*j*k + 5
52960*a^4*b^6*c^6*d*e*j*k - 69120*a^3*b^8*c^5*d*e*j*k - 16588800*a^6*b^2*c^
8*d*e*h*l - 7741440*a^6*b^2*c^8*d*f*g*l + 6946560*a^5*b^4*c^7*d*f*h*k - 552
9600*a^6*b^2*c^8*d*g*h*j + 5419008*a^5*b^4*c^7*d*e*g*m - 5087232*a^6*b^2*c^
8*e*f*g*k - 3870720*a^5*b^4*c^7*d*f*g*l - 3686400*a^6*b^2*c^8*e*f*h*j - 221
1840*a^5*b^4*c^7*d*g*h*j - 1755648*a^4*b^6*c^6*d*f*h*k - 1658880*a^5*b^4*c^
7*e*f*g*k + 1658880*a^5*b^4*c^7*d*e*h*l - 1566720*a^5*b^4*c^7*e*f*h*j + 145
1520*a^4*b^6*c^6*d*f*g*l - 483840*a^4*b^6*c^6*d*e*g*m + 317952*a^4*b^6*c^6*
d*g*h*j - 193536*a^3*b^8*c^5*d*f*g*l + 124416*a^4*b^6*c^6*d*e*h*l + 114696*
a^3*b^8*c^5*d*f*h*k + 69120*a^4*b^6*c^6*e*f*g*k - 41472*a^3*b^8*c^5*d*e*h*l
 - 36864*a^4*b^6*c^6*e*f*h*j + 14580*a^2*b^10*c^4*d*f*h*k + 6912*a^3*b^8*c^
5*d*g*h*j - 6912*a^2*b^10*c^4*d*g*h*j + 6912*a^2*b^10*c^4*d*f*g*l + 4608*a^
3*b^8*c^5*e*f*h*j + 7962624*a^5*b^3*c^8*d*e*g*k + 7741440*a^5*b^3*c^8*d*e*f
*l + 5160960*a^5*b^3*c^8*d*f*g*j + 4423680*a^5*b^3*c^8*d*e*h*j - 2903040*a^
4*b^5*c^7*d*e*f*l - 2073600*a^4*b^5*c^7*d*e*g*k - 635904*a^4*b^5*c^7*d*e*h*
j + 387072*a^3*b^7*c^6*d*e*f*l - 354816*a^3*b^7*c^6*d*f*g*j + 322560*a^4*b^
5*c^7*d*f*g*j + 207360*a^3*b^7*c^6*d*e*g*k + 59904*a^2*b^9*c^5*d*f*g*j - 13
824*a^3*b^7*c^6*d*e*h*j + 13824*a^2*b^9*c^5*d*e*h*j - 13824*a^2*b^9*c^5*d*e
*f*l + 4423680*a^5*b^3*c^8*e*f*g*h + 138240*a^4*b^5*c^7*e*f*g*h - 13824*a^3
*b^7*c^6*e*f*g*h - 10321920*a^5*b^2*c^9*d*e*f*j + 709632*a^3*b^6*c^7*d*e*f*
j - 645120*a^4*b^4*c^8*d*e*f*j - 119808*a^2*b^8*c^6*d*e*f*j - 16588800*a^5*
b^2*c^9*d*e*g*h + 1658880*a^4*b^4*c^8*d*e*g*h + 124416*a^3*b^6*c^7*d*e*g*h
- 41472*a^2*b^8*c^6*d*e*g*h + 7741440*a^4*b^3*c^9*d*e*f*g - 2903040*a^3*b^5
*c^8*d*e*f*g + 387072*a^2*b^7*c^7*d*e*f*g + 3456*a^7*b^8*c*k*l^2*m + 12672*
a^7*b^8*c*j*l*m^2 + 384*a^5*b^10*c*j^2*k*m - 1635840*a^10*b*c^5*h*k*m^2 - 1
009152*a^9*b*c^6*h^2*k*m + 3690*a^6*b^9*c*h*k*m^2 + 1152*a^6*b^9*c*g*l*m^2
- 540*a^5*b^10*c*h*k^2*m + 54*a^4*b^11*c*h^2*k*m + 565248*a^9*b*c^6*h*j^2*m
 - 39771648*a^7*b*c^8*d^2*k*m - 2496000*a^8*b*c^7*f^2*k*m - 1543680*a^9*b*c
^6*f*k^2*m + 1980*a^5*b^10*c*f*k*m^2 - 384*a^5*b^10*c*g*j*m^2 - 180*a^4*b^1
1*c*f*k^2*m + 6*a^2*b^13*c*f^2*k*m - 10298880*a^9*b*c^6*d*k*m^2 + 2580480*a
^9*b*c^6*e*j*m^2 + 5310*a^4*b^11*c*d*k*m^2 - 1674*a*b^13*c^2*d^2*k*m - 540*

a^3*b^12*c*d*k^2*m - 10616832*a^7*b*c^8*e^2*j*l - 3538944*a^8*b*c^7*e*j^2*l
 + 2727936*a^8*b*c^7*d*j^2*m - 2496000*a^9*b*c^6*f*h*m^2 - 1543680*a^8*b*c^
7*f*h^2*m + 565248*a^8*b*c^7*f*j^2*k - 270*a^4*b^11*c*f*h*m^2 - 59512320*a^
6*b*c^9*d^2*f*m + 5087232*a^7*b*c^8*e^2*h*m + 1105920*a^8*b*c^7*e*j*k^2 - 3
456*a*b^12*c^3*d^2*j*l - 1635840*a^7*b*c^8*f^2*h*k - 1009152*a^8*b*c^7*f*h*
k^2 + 10260*a*b^12*c^3*d^2*h*m - 684*a^3*b^12*c*d*h*m^2 - 24675840*a^6*b*c^
9*d^2*h*k - 15552000*a^8*b*c^7*d*f*m^2 + 24551424*a^6*b*c^9*d*e^2*m - 39398
40*a^7*b*c^8*d*h^2*k + 1105920*a^7*b*c^8*e*h^2*j - 25074*a*b^11*c^4*d^2*f*m
 + 10530*a*b^11*c^4*d^2*h*k + 10368*a*b^11*c^4*d^2*g*l + 420*a*b^12*c^3*d*f
^2*m - 378*a^2*b^13*c*d*f*m^2 - 10616832*a^6*b*c^9*e^2*g*j + 5087232*a^6*b*
c^9*e^2*f*k - 3538944*a^7*b*c^8*e*g*j^2 + 1843200*a^7*b*c^8*d*h*j^2 - 79948
80*a^6*b*c^9*d*f^2*k - 4990464*a^7*b*c^8*d*f*k^2 + 2580480*a^6*b*c^9*e*f^2*
j + 65664*a*b^10*c^5*d^2*g*j - 27972*a*b^10*c^5*d^2*f*k - 20736*a*b^10*c^5*
d^2*e*l + 1260*a*b^11*c^4*d*f^2*k + 54*a*b^13*c^2*d*f*k^2 + 23224320*a^5*b*
c^10*d^2*e*j - 37062144*a^5*b*c^10*d^2*f*h + 384*a*b^12*c^3*d*f*j^2 - 13132
8*a*b^9*c^6*d^2*e*j - 5985792*a^6*b*c^9*d*f*h^2 + 206010*a*b^9*c^6*d^2*f*h
 - 6300*a*b^10*c^5*d*f^2*h + 1350*a*b^11*c^4*d*f*h^2 + 16588800*a^5*b*c^10*d
*e^2*h + 3456*a*b^10*c^5*d*f*g^2 + 435456*a*b^8*c^7*d^2*e*g + 13824*a*b^8*c
^7*d*e^2*f - 1474560*a^9*c^7*e*j*k*m + 460800*a^9*c^7*f*h*k*m + 3225600*a^8
*c^8*d*f*k*m - 2457600*a^8*c^8*e*f*j*m - 884736*a^8*c^8*e*h*j*k - 6193152*a
^7*c^9*d*e*j*k + 1935360*a^7*c^9*d*f*h*k - 1474560*a^7*c^9*e*f*h*j - 103219
20*a^6*c^10*d*e*f*j - 1105920*a^9*b^4*c^3*k*l^2*m - 552960*a^10*b^2*c^4*k*l
^2*m - 34560*a^8*b^6*c^2*k*l^2*m - 1290240*a^10*b^2*c^4*j*l*m^2 - 860160*a^
9*b^4*c^3*j*l*m^2 - 80640*a^8*b^6*c^2*j*l*m^2 - 737280*a^9*b^2*c^5*j^2*k*m
 - 568320*a^8*b^4*c^4*j^2*k*m - 136704*a^7*b^6*c^3*j^2*k*m - 2304*a^6*b^8*c^
2*j^2*k*m + 1271808*a^9*b^3*c^4*h*l^2*m - 552960*a^9*b^2*c^5*j*k^2*l - 5529
60*a^8*b^4*c^4*j*k^2*l + 414720*a^8*b^5*c^3*h*l^2*m - 145152*a^7*b^6*c^3*j*
k^2*l - 17280*a^7*b^7*c^2*h*l^2*m - 3456*a^6*b^8*c^2*j*k^2*l - 3640320*a^9*
b^3*c^4*h*k*m^2 - 2626560*a^8*b^3*c^5*h^2*k*m + 2211840*a^9*b^2*c^5*h*k^2*m
 + 2056320*a^8*b^4*c^4*h*k^2*m + 1935360*a^9*b^3*c^4*g*l*m^2 - 1143360*a^8*
b^5*c^3*h*k*m^2 - 1097280*a^7*b^5*c^4*h^2*k*m + 364608*a^7*b^6*c^3*h*k^2*m
 + 322560*a^8*b^5*c^3*g*l*m^2 - 56160*a^6*b^7*c^3*h^2*k*m - 40320*a^7*b^7*c^
2*g*l*m^2 + 27936*a^7*b^7*c^2*h*k*m^2 - 3780*a^6*b^8*c^2*h*k^2*m + 2970*a^5
*b^9*c^2*h^2*k*m - 1419264*a^8*b^4*c^4*f*l^2*m - 1105920*a^7*b^4*c^5*g^2*k*
m - 921600*a^9*b^2*c^5*f*l^2*m - 829440*a^8*b^4*c^4*h*k*l^2 + 749568*a^8*b^
3*c^5*h*j^2*m - 552960*a^8*b^2*c^6*g^2*k*m - 331776*a^9*b^2*c^5*h*k*l^2 + 3
17952*a^7*b^5*c^4*h*j^2*m - 103680*a^7*b^6*c^3*h*k*l^2 + 80640*a^7*b^6*c^3*
f*l^2*m + 38400*a^6*b^7*c^3*h*j^2*m - 34560*a^6*b^6*c^4*g^2*k*m + 3456*a^5*
b^8*c^3*g^2*k*m - 1920*a^5*b^9*c^2*h*j^2*m - 5142528*a^7*b^3*c^6*f^2*k*m +
5068800*a^9*b^2*c^5*f*k*m^2 - 3870720*a^9*b^2*c^5*e*l*m^2 - 3755520*a^8*b^3
*c^5*f*k^2*m + 3000960*a^8*b^4*c^4*f*k*m^2 - 1290240*a^9*b^2*c^5*g*j*m^2 -
1085760*a^7*b^5*c^4*f*k^2*m - 959040*a^6*b^5*c^5*f^2*k*m - 860160*a^8*b^4*c
^4*g*j*m^2 + 829440*a^8*b^3*c^5*g*k^2*l - 645120*a^8*b^4*c^4*e*l*m^2 - 5529
60*a^8*b^2*c^6*h^2*j*l - 552960*a^7*b^4*c^5*h^2*j*l + 414720*a^7*b^5*c^4*g*
k^2*l - 145152*a^6*b^6*c^4*h^2*j*l + 103200*a^5*b^7*c^4*f^2*k*m - 80640*a^7

*b^6*c^3*g*j*m^2 + 80640*a^7*b^6*c^3*e*l*m^2 + 41280*a^7*b^6*c^3*f*k*m^2 - 37188*a^6*b^8*c^2*f*k*m^2 + 13536*a^6*b^7*c^3*f*k^2*m + 12672*a^6*b^8*c^2*g*j*m^2 + 10368*a^6*b^7*c^3*g*k^2*l + 5490*a^5*b^9*c^2*f*k^2*m - 3456*a^5*b^8*c^3*h^2*j*l - 2304*a^6*b^8*c^2*e*l*m^2 + 810*a^4*b^9*c^3*f^2*k*m - 270*a^3*b^11*c^2*f^2*k*m + 6137856*a^8*b^3*c^5*d*l^2*m - 4423680*a^7*b^2*c^7*e^2*k*m - 2654208*a^8*b^3*c^5*g*j*l^2 - 2654208*a^7*b^3*c^6*g^2*j*l + 1769472*a^8*b^2*c^6*g*j^2*l + 1769472*a^7*b^4*c^5*g*j^2*l - 1354752*a^7*b^5*c^4*d*l^2*m - 1327104*a^7*b^5*c^4*g*j*l^2 - 1327104*a^6*b^5*c^5*g^2*j*l + 1271808*a^8*b^3*c^5*f*k*l^2 - 1040384*a^8*b^2*c^6*f*j^2*m - 697344*a^7*b^4*c^5*f*j^2*m - 516096*a^8*b^2*c^6*h*j^2*k - 451584*a^7*b^4*c^5*h*j^2*k + 442368*a^6*b^6*c^4*g*j^2*l + 414720*a^7*b^5*c^4*f*k*l^2 - 138240*a^6*b^6*c^4*h*j^2*k - 138240*a^6*b^4*c^6*e^2*k*m - 121856*a^6*b^6*c^4*f*j^2*m + 120960*a^6*b^7*c^3*d*l^2*m - 17280*a^6*b^7*c^3*f*k*l^2 + 13824*a^5*b^6*c^5*e^2*k*m - 11520*a^5*b^8*c^3*h*j^2*k + 8960*a^5*b^8*c^3*f*j^2*m + 10851840*a^8*b^2*c^6*d*k^2*m - 10464768*a^6*b^3*c^7*d^2*k*m - 10275840*a^8*b^3*c^5*d*k*m^2 + 7121088*a^5*b^5*c^6*d^2*k*m + 3127680*a^7*b^4*c^5*d*k^2*m + 1720320*a^8*b^3*c^5*e*j*m^2 - 1658880*a^8*b^2*c^6*e*k^2*l - 1290240*a^7*b^2*c^7*f^2*j*l + 1271808*a^7*b^3*c^6*g^2*h*m - 1222560*a^4*b^7*c^5*d^2*k*m + 999360*a^7*b^5*c^4*d*k*m^2 - 860160*a^6*b^4*c^6*f^2*j*l - 829440*a^7*b^4*c^5*e*k^2*l - 705024*a^6*b^6*c^4*d*k^2*m - 552960*a^8*b^2*c^6*g*j*k^2 - 552960*a^7*b^4*c^5*g*j*k^2 + 414720*a^6*b^5*c^5*g^2*h*m + 319392*a^6*b^7*c^3*d*k*m^2 + 161280*a^7*b^5*c^4*e*j*m^2 - 145152*a^6*b^6*c^4*g*j*k^2 - 85734*a^5*b^9*c^2*d*k*m^2 - 80640*a^5*b^6*c^5*f^2*j*l - 25344*a^6*b^7*c^3*e*j*m^2 + 23490*a^3*b^9*c^4*d^2*k*m - 20736*a^6*b^6*c^4*e*k^2*l - 17280*a^5*b^7*c^4*g^2*h*m + 14148*a^5*b^8*c^3*d*k^2*m + 13716*a^2*b^11*c^3*d^2*k*m + 12690*a^4*b^10*c^2*d*k^2*m + 12672*a^4*b^8*c^4*f^2*j*l - 3456*a^5*b^8*c^3*g*j*k^2 + 768*a^5*b^9*c^2*e*j*m^2 - 384*a^3*b^10*c^3*f^2*j*l + 5308416*a^8*b^2*c^6*e*j*l^2 - 5308416*a^6*b^3*c^7*e^2*j*l - 5142528*a^8*b^3*c^5*f*h*m^2 + 5068800*a^7*b^2*c^7*f^2*h*m - 3755520*a^7*b^3*c^6*f*h^2*m - 3538944*a^7*b^3*c^6*e*j^2*l + 3000960*a^6*b^4*c^6*f^2*h*m + 2654208*a^7*b^4*c^5*e*j*l^2 - 2322432*a^8*b^2*c^6*d*k*l^2 + 2125824*a^7*b^3*c^6*d*j^2*m - 1990656*a^7*b^4*c^5*d*k*l^2 - 1085760*a^6*b^5*c^5*f*h^2*m - 959040*a^7*b^5*c^4*f*h*m^2 - 884736*a^6*b^5*c^5*e*j^2*l + 829440*a^7*b^3*c^6*g*h^2*l + 749568*a^7*b^3*c^6*f*j^2*k + 518400*a^6*b^6*c^4*d*k*l^2 + 414720*a^6*b^5*c^5*g*h^2*l + 317952*a^6*b^5*c^5*f*j^2*k + 133632*a^6*b^5*c^5*d*j^2*m + 103200*a^6*b^7*c^3*f*h*m^2 - 96768*a^5*b^7*c^4*d*j^2*m - 51840*a^5*b^8*c^3*d*k*l^2 + 41280*a^5*b^6*c^5*f^2*h*m + 38400*a^5*b^7*c^4*f*j^2*k - 37188*a^4*b^8*c^4*f^2*h*m + 13536*a^5*b^7*c^4*f*h^2*m + 13440*a^4*b^9*c^3*d*j^2*m + 10368*a^5*b^7*c^4*g*h^2*l + 5490*a^4*b^9*c^3*f*h^2*m + 1980*a^3*b^10*c^3*f^2*h*m - 1920*a^4*b^9*c^3*f*j^2*k + 810*a^5*b^9*c^2*f*h*m^2 - 180*a^3*b^11*c^2*f*h^2*m - 30*a^2*b^12*c^2*f^2*h*m + 30067200*a^6*b^2*c^8*d^2*h*m - 11612160*a^6*b^2*c^8*d^2*j*l + 1658880*a^6*b^3*c^7*e^2*h*m + 1596672*a^4*b^6*c^6*d^2*j*l - 1419264*a^6*b^4*c^6*f*g^2*m - 1105920*a^7*b^4*c^5*f*h*l^2 + 1105920*a^7*b^3*c^6*e*j*k^2 - 921600*a^7*b^2*c^7*f*g^2*m - 829440*a^6*b^4*c^6*g^2*h*k - 552960*a^8*b^2*c^6*f*h*l^2 - 508032*a^3*b^8*c^5*d^2*j*l - 331776*a^7*b^2*c^7*g^2*h*k + 290304*a^6*b^5*c^5*e*j*k^2 - 10368

0*a^5*b^6*c^5*g^2*h*k + 80640*a^5*b^6*c^5*f*g^2*m - 69120*a^5*b^5*c^6*e^2*h
*m + 65664*a^2*b^10*c^4*d^2*j*l - 34560*a^6*b^6*c^4*f*h*l^2 + 6912*a^5*b^7*
c^4*e*j*k^2 + 3456*a^5*b^8*c^3*f*h*l^2 + 11930112*a^8*b^2*c^6*d*h*m^2 + 843
2640*a^7*b^2*c^7*d*h^2*m + 4450176*a^7*b^4*c^5*d*h*m^2 + 4337280*a^6*b^4*c^
6*d*h^2*m - 3870720*a^8*b^2*c^6*e*g*m^2 - 3640320*a^6*b^3*c^7*f^2*h*k - 288
5760*a^5*b^4*c^7*d^2*h*m - 2844288*a^4*b^6*c^6*d^2*h*m - 2626560*a^7*b^3*c^
6*f*h*k^2 + 2211840*a^7*b^2*c^7*f*h^2*k + 2056320*a^6*b^4*c^6*f*h^2*k + 193
5360*a^6*b^3*c^7*f^2*g*l - 1916928*a^7*b^2*c^7*d*j^2*k - 1687680*a^6*b^6*c^
4*d*h*m^2 - 1658880*a^7*b^2*c^7*e*h^2*l - 1143360*a^5*b^5*c^6*f^2*h*k - 109
7280*a^6*b^5*c^5*f*h*k^2 + 1019412*a^3*b^8*c^5*d^2*h*m - 1007424*a^5*b^6*c^
5*d*h^2*m - 912384*a^6*b^4*c^6*d*j^2*k - 829440*a^6*b^4*c^6*e*h^2*l - 64512
0*a^7*b^4*c^5*e*g*m^2 - 552960*a^7*b^2*c^7*g*h^2*j - 552960*a^6*b^4*c^6*g*h
^2*j + 364608*a^5*b^6*c^5*f*h^2*k + 322560*a^5*b^5*c^6*f^2*g*l + 197460*a^5
*b^8*c^3*d*h*m^2 - 145152*a^5*b^6*c^5*g*h^2*j - 143802*a^2*b^10*c^4*d^2*h*m
 + 80640*a^6*b^6*c^4*e*g*m^2 - 56160*a^5*b^7*c^4*f*h*k^2 + 51948*a^4*b^8*c^
4*d*h^2*m - 40320*a^4*b^7*c^5*f^2*g*l + 34560*a^4*b^8*c^4*d*j^2*k + 27936*a
^4*b^7*c^5*f^2*h*k - 20736*a^5*b^6*c^5*e*h^2*l - 13824*a^5*b^6*c^5*d*j^2*k
 + 10800*a^3*b^10*c^3*d*h^2*m - 5760*a^3*b^10*c^3*d*j^2*k - 3780*a^4*b^8*c^4
*f*h^2*k + 3690*a^3*b^9*c^4*f^2*h*k - 3456*a^4*b^8*c^4*g*h^2*j + 2970*a^4*b
^9*c^3*f*h*k^2 - 2304*a^5*b^8*c^3*e*g*m^2 + 1152*a^3*b^9*c^4*f^2*g*l - 540*
a^3*b^10*c^3*f*h^2*k - 540*a^2*b^12*c^2*d*h^2*m - 90*a^4*b^10*c^2*d*h*m^2 -
 90*a^2*b^11*c^3*f^2*h*k + 54*a^3*b^11*c^2*f*h*k^2 + 15925248*a^6*b^2*c^8*e
^2*g*l - 7962624*a^7*b^3*c^6*e*g*l^2 - 7962624*a^6*b^3*c^7*e*g^2*l + 233856
00*a^6*b^2*c^8*d*f^2*m + 6137856*a^6*b^3*c^7*d*g^2*m - 5677056*a^6*b^2*c^8*
e^2*f*m + 4147200*a^7*b^3*c^6*d*h*l^2 - 3317760*a^6*b^2*c^8*e^2*h*k - 13547
52*a^5*b^5*c^6*d*g^2*m + 1271808*a^6*b^3*c^7*f*g^2*k - 737280*a^7*b^2*c^7*f
*h*j^2 + 17418240*a^5*b^3*c^8*d^2*g*l - 568320*a^6*b^4*c^6*f*h*j^2 - 414720
*a^6*b^5*c^5*d*h*l^2 + 414720*a^5*b^5*c^6*f*g^2*k - 414720*a^5*b^4*c^7*e^2*
h*k + 322560*a^5*b^4*c^7*e^2*f*m - 136704*a^5*b^6*c^5*f*h*j^2 + 120960*a^4*
b^7*c^5*d*g^2*m - 31104*a^5*b^7*c^4*d*h*l^2 - 17280*a^4*b^7*c^5*f*g^2*k + 1
0368*a^4*b^9*c^3*d*h*l^2 - 2304*a^4*b^8*c^4*f*h*j^2 + 384*a^3*b^10*c^3*f*h*
j^2 + 50042880*a^5*b^2*c^9*d^2*f*k - 13271040*a^5*b^3*c^8*d^2*h*k - 1314969
6*a^7*b^3*c^6*d*f*m^2 + 10906560*a^4*b^5*c^7*d^2*f*m - 8709120*a^4*b^5*c^7*
d^2*g*l - 7418880*a^5*b^3*c^8*d^2*f*m + 7133184*a^7*b^2*c^7*d*h*k^2 - 64281
60*a^6*b^3*c^7*d*h^2*k + 5593536*a^4*b^5*c^7*d^2*h*k - 3870720*a^6*b^2*c^8*
e*f^2*l + 3369600*a^6*b^4*c^6*d*h*k^2 + 3148992*a^6*b^5*c^5*d*f*m^2 - 29856
96*a^3*b^7*c^6*d^2*f*m + 1959552*a^3*b^7*c^6*d^2*g*l - 1658880*a^7*b^2*c^7*
e*g*k^2 - 1505280*a^4*b^6*c^6*d*f^2*m - 1290240*a^6*b^2*c^8*f^2*g*j - 34836
480*a^5*b^2*c^9*d^2*e*l + 1105920*a^6*b^3*c^7*e*h^2*j - 860160*a^5*b^4*c^7*
f^2*g*j - 829440*a^6*b^4*c^6*e*g*k^2 - 692064*a^3*b^7*c^6*d^2*h*k - 689472*
a^5*b^5*c^6*d*h^2*k - 645120*a^5*b^4*c^7*e*f^2*l - 388800*a^5*b^6*c^5*d*h*k
^2 + 378954*a^2*b^9*c^5*d^2*f*m + 362880*a^5*b^4*c^7*d*f^2*m + 296964*a^3*b
^8*c^5*d*f^2*m + 290304*a^5*b^5*c^6*e*h^2*j + 277344*a^4*b^7*c^5*d*h^2*k -
217728*a^2*b^9*c^5*d^2*g*l - 80640*a^4*b^6*c^6*f^2*g*j + 80640*a^4*b^6*c^6*
e*f^2*l - 77070*a^4*b^9*c^3*d*f*m^2 - 30240*a^5*b^7*c^4*d*f*m^2 - 28350*a^3

*b^9*c^4*d*h^2*k - 26406*a^2*b^9*c^5*d^2*h*k - 21060*a^4*b^8*c^4*d*h*k^2 -
20736*a^5*b^6*c^5*e*g*k^2 - 19278*a^2*b^10*c^4*d*f^2*m + 12672*a^3*b^8*c^5*
f^2*g*j + 10044*a^3*b^10*c^3*d*h*k^2 + 8820*a^3*b^11*c^2*d*f*m^2 + 6912*a^4
*b^7*c^5*e*h^2*j - 2304*a^3*b^8*c^5*e*f^2*l - 1620*a^2*b^11*c^3*d*h^2*k - 3
84*a^2*b^10*c^4*f^2*g*j + 162*a^2*b^12*c^2*d*h*k^2 - 5419008*a^5*b^3*c^8*d*
e^2*m + 5308416*a^6*b^2*c^8*e*g^2*j - 5308416*a^5*b^3*c^8*e^2*g*j - 3870720
*a^7*b^2*c^7*d*f*l^2 - 3538944*a^6*b^3*c^7*e*g*j^2 + 2654208*a^5*b^4*c^7*e*
g^2*j - 2322432*a^6*b^2*c^8*d*g^2*k - 1990656*a^5*b^4*c^7*d*g^2*k - 1935360
*a^6*b^4*c^6*d*f*l^2 + 1658880*a^6*b^3*c^7*d*h*j^2 + 1658880*a^5*b^3*c^8*e^
2*f*k - 884736*a^5*b^5*c^6*e*g*j^2 + 725760*a^5*b^6*c^5*d*f*l^2 + 17418240*
a^4*b^4*c^8*d^2*e*l + 518400*a^4*b^6*c^6*d*g^2*k + 483840*a^4*b^5*c^7*d*e^2
*m + 262656*a^5*b^5*c^6*d*h*j^2 - 96768*a^4*b^8*c^4*d*f*l^2 - 69120*a^4*b^5
*c^7*e^2*f*k - 55296*a^4*b^7*c^5*d*h*j^2 - 51840*a^3*b^8*c^5*d*g^2*k + 3456
*a^3*b^10*c^3*d*f*l^2 + 1152*a^3*b^9*c^4*d*h*j^2 + 1152*a^2*b^11*c^3*d*h*j^
2 - 15431040*a^4*b^4*c^8*d^2*f*k - 13248000*a^5*b^3*c^8*d*f^2*k - 11612160*
a^5*b^2*c^9*d^2*g*j - 10063872*a^6*b^3*c^7*d*f*k^2 - 3919104*a^3*b^6*c^7*d^
2*e*l + 2554560*a^4*b^5*c^7*d*f^2*k + 1720320*a^5*b^3*c^8*e*f^2*j + 1596672
*a^3*b^6*c^7*d^2*g*j + 1518912*a^3*b^6*c^7*d^2*f*k - 1105920*a^5*b^4*c^7*f*
g^2*h + 838080*a^5*b^5*c^6*d*f*k^2 - 552960*a^6*b^2*c^8*f*g^2*h - 508032*a^
2*b^8*c^6*d^2*g*j + 435456*a^2*b^8*c^6*d^2*e*l + 161280*a^4*b^5*c^7*e*f^2*j
 + 116640*a^4*b^7*c^5*d*f*k^2 + 106812*a^2*b^8*c^6*d^2*f*k - 98208*a^3*b^7*
c^6*d*f^2*k - 34560*a^4*b^6*c^6*f*g^2*h - 27270*a^3*b^9*c^4*d*f*k^2 - 26334
*a^2*b^9*c^5*d*f^2*k - 25344*a^3*b^7*c^6*e*f^2*j + 3456*a^3*b^8*c^5*f*g^2*h
 + 768*a^2*b^9*c^5*e*f^2*j - 702*a^2*b^11*c^3*d*f*k^2 - 7962624*a^5*b^2*c^9
*d*e^2*k - 2580480*a^6*b^2*c^8*d*f*j^2 + 2073600*a^4*b^4*c^8*d*e^2*k - 1658
880*a^6*b^2*c^8*e*g*h^2 - 967680*a^5*b^4*c^7*d*f*j^2 - 829440*a^5*b^4*c^7*e
*g*h^2 - 207360*a^3*b^6*c^7*d*e^2*k + 64512*a^4*b^6*c^6*d*f*j^2 + 39168*a^3
*b^8*c^5*d*f*j^2 - 20736*a^4*b^6*c^6*e*g*h^2 - 9216*a^2*b^10*c^4*d*f*j^2 -
4423680*a^5*b^2*c^9*e^2*f*h + 4147200*a^5*b^3*c^8*d*g^2*h - 3193344*a^3*b^5
*c^8*d^2*e*j + 1016064*a^2*b^7*c^7*d^2*e*j - 414720*a^4*b^5*c^7*d*g^2*h - 1
38240*a^4*b^4*c^8*e^2*f*h - 31104*a^3*b^7*c^6*d*g^2*h + 13824*a^3*b^6*c^7*e
^2*f*h + 10368*a^2*b^9*c^5*d*g^2*h + 15630336*a^5*b^2*c^9*d*f^2*h - 1445990
4*a^4*b^3*c^9*d^2*f*h + 9630144*a^3*b^5*c^8*d^2*f*h - 8764416*a^5*b^3*c^8*d
*f*h^2 - 3870720*a^5*b^2*c^9*e*f^2*g + 2867328*a^4*b^4*c^8*d*f^2*h - 209520
0*a^2*b^7*c^7*d^2*f*h - 1414080*a^3*b^6*c^7*d*f^2*h - 34836480*a^4*b^2*c^10
*d^2*e*g - 645120*a^4*b^4*c^8*e*f^2*g + 306720*a^3*b^7*c^6*d*f*h^2 + 197820
*a^2*b^8*c^6*d*f^2*h + 146880*a^4*b^5*c^7*d*f*h^2 + 80640*a^3*b^6*c^7*e*f^2
*g - 55350*a^2*b^9*c^5*d*f*h^2 - 2304*a^2*b^8*c^6*e*f^2*g - 3870720*a^5*b^2
*c^9*d*f*g^2 - 1935360*a^4*b^4*c^8*d*f*g^2 - 1658880*a^4*b^3*c^9*d*e^2*h +
725760*a^3*b^6*c^7*d*f*g^2 + 17418240*a^3*b^4*c^9*d^2*e*g - 124416*a^3*b^5*
c^8*d*e^2*h - 96768*a^2*b^8*c^6*d*f*g^2 + 41472*a^2*b^7*c^7*d*e^2*h - 39191
04*a^2*b^6*c^8*d^2*e*g - 7741440*a^4*b^2*c^10*d*e^2*f + 2903040*a^3*b^4*c^9
*d*e^2*f - 387072*a^2*b^6*c^8*d*e^2*f - 20160*a^8*b^7*c*l^2*m^2 - 1648128*a
^10*b^3*c^3*k*m^3 - 898560*a^9*b^3*c^4*k^3*m - 354240*a^9*b^5*c^2*k*m^3 - 3
54240*a^8*b^5*c^3*k^3*m - 21600*a^7*b^7*c^2*k^3*m - 13950*a^7*b^8*c*k^2*m^2

 + 430080*a^10*b*c^5*j^2*m^2 - 1984*a^6*b^9*c*j^2*m^2 - 884736*a^9*b^3*c^4*
j*l^3 - 589824*a^8*b^3*c^5*j^3*l - 442368*a^8*b^5*c^3*j*l^3 - 294912*a^7*b^
5*c^4*j^3*l - 49152*a^6*b^7*c^3*j^3*l + 1359360*a^10*b^2*c^4*h*m^3 + 117312
0*a^9*b^4*c^3*h*m^3 + 743040*a^7*b^4*c^5*h^3*m + 622080*a^8*b^2*c^6*h^3*m +
 184320*a^9*b*c^6*j^2*k^2 + 107136*a^6*b^6*c^4*h^3*m - 32640*a^8*b^6*c^2*h*
m^3 + 540*a^5*b^8*c^3*h^3*m - 270*a^4*b^10*c^2*h^3*m - 180*a^5*b^10*c*h^2*m
^2 - 2293760*a^9*b^3*c^4*f*m^3 - 2293760*a^6*b^3*c^7*f^3*m + 1327104*a^8*b^
4*c^4*g*l^3 + 1327104*a^6*b^4*c^6*g^3*l - 622080*a^8*b^3*c^5*h*k^3 - 622080
*a^7*b^3*c^6*h^3*k - 326592*a^7*b^5*c^4*h*k^3 - 326592*a^6*b^5*c^5*h^3*k -
199360*a^8*b^5*c^3*f*m^3 - 199360*a^5*b^5*c^6*f^3*m + 61920*a^7*b^7*c^2*f*m
^3 + 61920*a^4*b^7*c^5*f^3*m - 38880*a^6*b^7*c^3*h*k^3 - 38880*a^5*b^7*c^4*
h^3*k - 3682*a^3*b^9*c^4*f^3*m - 810*a^5*b^9*c^2*h*k^3 - 810*a^4*b^9*c^3*h^
3*k - 70*a^3*b^12*c*f^2*m^2 + 70*a^2*b^11*c^3*f^3*m + 3870720*a^8*b*c^7*e^2
*m^2 + 184320*a^8*b*c^7*h^2*j^2 - 14152320*a^4*b^4*c^8*d^3*m + 10644480*a^5
*b^2*c^9*d^3*m + 5483520*a^9*b^2*c^5*d*m^3 + 4269888*a^3*b^6*c^7*d^3*m - 26
54208*a^8*b^3*c^5*e*l^3 + 1359360*a^6*b^2*c^8*f^3*k + 1330560*a^8*b^4*c^4*d
*m^3 + 1173120*a^5*b^4*c^7*f^3*k - 884736*a^6*b^3*c^7*g^3*j - 826560*a^7*b^
6*c^3*d*m^3 + 743040*a^7*b^4*c^5*f*k^3 + 622080*a^8*b^2*c^6*f*k^3 - 607068*
a^2*b^8*c^6*d^3*m - 589824*a^7*b^3*c^6*g*j^3 - 442368*a^5*b^5*c^6*g^3*j - 2
94912*a^6*b^5*c^5*g*j^3 + 145188*a^6*b^8*c^2*d*m^3 + 107136*a^6*b^6*c^4*f*k
^3 - 49152*a^5*b^7*c^4*g*j^3 - 32640*a^4*b^6*c^6*f^3*k - 5796*a^3*b^8*c^5*f
^3*k + 540*a^5*b^8*c^3*f*k^3 - 270*a^4*b^10*c^2*f*k^3 + 210*a^2*b^10*c^4*f^
3*k + 19077120*a^4*b^3*c^9*d^3*k + 1658880*a^7*b*c^8*e^2*k^2 + 430080*a^7*b
*c^8*f^2*j^2 + 3538944*a^5*b^2*c^9*e^3*j - 2488320*a^7*b^3*c^6*d*k^3 - 2379
456*a^3*b^5*c^8*d^3*k + 1179648*a^7*b^2*c^7*e*j^3 + 589824*a^6*b^4*c^6*e*j^
3 + 98304*a^5*b^6*c^5*e*j^3 - 95904*a^2*b^7*c^7*d^3*k - 57024*a^6*b^5*c^5*d
*k^3 + 49248*a^5*b^7*c^4*d*k^3 - 4050*a^4*b^9*c^3*d*k^3 - 810*a^3*b^11*c^2*
d*k^3 - 486*a*b^12*c^3*d^2*k^2 + 3870720*a^6*b*c^9*d^2*j^2 - 1648128*a^5*b^
3*c^8*f^3*h - 898560*a^6*b^3*c^7*f*h^3 - 354240*a^5*b^5*c^6*f*h^3 - 354240*
a^4*b^5*c^7*f^3*h + 43680*a^3*b^7*c^6*f^3*h - 21600*a^4*b^7*c^5*f*h^3 - 979
2*a*b^11*c^4*d^2*j^2 + 1350*a^3*b^9*c^4*f*h^3 - 1050*a^2*b^9*c^5*f^3*h + 16
58880*a^6*b*c^9*e^2*h^2 + 16547328*a^4*b^2*c^10*d^3*h - 12306816*a^3*b^4*c^
9*d^3*h + 37310976*a^3*b^3*c^10*d^3*f + 3037824*a^2*b^6*c^8*d^3*h - 2654208
*a^5*b^3*c^8*e*g^3 + 1949184*a^6*b^2*c^8*d*h^3 + 1296000*a^5*b^4*c^7*d*h^3
- 155520*a^4*b^6*c^6*d*h^3 - 40500*a*b^10*c^5*d^2*h^2 - 8100*a^3*b^8*c^5*d*
h^3 + 4050*a^2*b^10*c^4*d*h^3 + 3870720*a^5*b*c^10*e^2*f^2 + 34836480*a^4*b
*c^11*d^2*e^2 - 108864*a*b^9*c^6*d^2*g^2 - 8068032*a^2*b^5*c^9*d^3*f - 5623
296*a^4*b^3*c^9*d*f^3 + 1737792*a^3*b^5*c^8*d*f^3 - 260190*a*b^8*c^7*d^2*f^
2 - 211680*a^2*b^7*c^7*d*f^3 - 435456*a*b^7*c^8*d^2*e^2 - 245760*a^10*c^6*j
^2*k*m - 384*a^6*b^10*j*l*m^2 + 138240*a^10*c^6*h*k^2*m - 90*a^5*b^11*h*k*m
^2 + 384000*a^10*c^6*f*k*m^2 - 2211840*a^8*c^8*e^2*k*m - 409600*a^9*c^7*f*j
^2*m - 147456*a^9*c^7*h*j^2*k - 30*a^4*b^12*f*k*m^2 + 967680*a^9*c^7*d*k^2*
m + 384000*a^8*c^8*f^2*h*m - 90*a^3*b^13*d*k*m^2 + 20321280*a^7*c^9*d^2*h*m
 - 883200*a^11*b*c^4*k*m^3 - 317952*a^10*b*c^5*k^3*m + 43680*a^8*b^7*c*k*m^
3 + 1350*a^6*b^9*c*k^3*m - 270*b^14*c^2*d^2*h*m + 6*a^3*b^13*f*h*m^2 + 4838

400*a^9*c^7*d*h*m^2 + 2903040*a^8*c^8*d*h^2*m - 1032192*a^8*c^8*d*j^2*k + 1
38240*a^8*c^8*f*h^2*k - 3686400*a^7*c^9*e^2*f*m - 1327104*a^7*c^9*e^2*h*k -
393216*a^9*b*c^6*j^3*l - 245760*a^8*c^8*f*h*j^2 - 810*b^13*c^3*d^2*h*k + 6
30*b^13*c^3*d^2*f*m + 18*a^2*b^14*d*h*m^2 + 2688000*a^7*c^9*d*f^2*m + 58060
8*a^8*c^8*d*h*k^2 - 5796*a^7*b^8*c*h*m^3 - 3456*b^12*c^4*d^2*g*j + 1890*b^1
2*c^4*d^2*f*k + 6773760*a^6*c^10*d^2*f*k - 1344000*a^10*b*c^5*f*m^3 - 13440
00*a^7*b*c^8*f^3*m - 207360*a^9*b*c^6*h*k^3 - 207360*a^8*b*c^7*h^3*k - 3682
*a^6*b^9*c*f*m^3 - 9289728*a^6*c^10*d*e^2*k - 1720320*a^7*c^9*d*f*j^2 - 508
03200*a^5*b*c^10*d^3*k + 6912*b^11*c^5*d^2*e*j - 10616832*a^6*b*c^9*e^3*l -
2211840*a^6*c^10*e^2*f*h - 393216*a^8*b*c^7*g*j^3 + 43416*a*b^10*c^5*d^3*m
- 9576*a^5*b^10*c*d*m^3 - 9450*b^11*c^5*d^2*f*h - 504*a*b^14*c*d^2*m^2 + 1
612800*a^6*c^10*d*f^2*h - 1036800*a^8*b*c^7*d*k^3 + 45198*a*b^9*c^6*d^3*k -
20736*b^10*c^6*d^2*e*g - 75188736*a^4*b*c^11*d^3*f - 883200*a^6*b*c^9*f^3*
h - 317952*a^7*b*c^8*f*h^3 - 15482880*a^5*c^11*d*e^2*f - 10616832*a^5*b*c^1
0*e^3*g - 345060*a*b^8*c^7*d^3*h - 4262400*a^5*b*c^10*d*f^3 + 852768*a*b^7*
c^8*d^3*f + 7350*a*b^9*c^6*d*f^3 + 967680*a^10*b^3*c^3*l^2*m^2 + 161280*a^9
*b^5*c^2*l^2*m^2 + 1684224*a^10*b^2*c^4*k^2*m^2 + 1264320*a^9*b^4*c^3*k^2*m
^2 + 126720*a^8*b^6*c^2*k^2*m^2 + 501760*a^9*b^3*c^4*j^2*m^2 + 414720*a^9*b
^3*c^4*k^2*l^2 + 207360*a^8*b^5*c^3*k^2*l^2 + 170240*a^8*b^5*c^3*j^2*m^2 +
9216*a^7*b^7*c^2*j^2*m^2 + 5184*a^7*b^7*c^2*k^2*l^2 + 884736*a^9*b^2*c^5*j^
2*l^2 + 884736*a^8*b^4*c^4*j^2*l^2 + 221184*a^7*b^6*c^3*j^2*l^2 + 1419840*a
^8*b^4*c^4*h^2*m^2 + 1387008*a^9*b^2*c^5*h^2*m^2 + 276480*a^8*b^3*c^5*j^2*k
^2 + 140544*a^7*b^5*c^4*j^2*k^2 + 84960*a^7*b^6*c^3*h^2*m^2 + 25344*a^6*b^7
*c^3*j^2*k^2 - 8010*a^6*b^8*c^2*h^2*m^2 + 576*a^5*b^9*c^2*j^2*k^2 + 967680*
a^8*b^3*c^5*g^2*m^2 + 414720*a^8*b^3*c^5*h^2*l^2 + 207360*a^7*b^5*c^4*h^2*l
^2 + 161280*a^7*b^5*c^4*g^2*m^2 - 20160*a^6*b^7*c^3*g^2*m^2 + 5184*a^6*b^7*
c^3*h^2*l^2 + 576*a^5*b^9*c^2*g^2*m^2 + 3808000*a^8*b^2*c^6*f^2*m^2 + 19906
56*a^7*b^4*c^5*g^2*l^2 + 1643712*a^7*b^4*c^5*f^2*m^2 + 803520*a^7*b^4*c^5*h
^2*k^2 + 725760*a^8*b^2*c^6*h^2*k^2 + 207360*a^6*b^6*c^4*h^2*k^2 - 125440*a
^6*b^6*c^4*f^2*m^2 - 13790*a^5*b^8*c^3*f^2*m^2 + 10530*a^5*b^8*c^3*h^2*k^2
+ 1785*a^4*b^10*c^2*f^2*m^2 + 81*a^4*b^10*c^2*h^2*k^2 + 18427392*a^7*b^2*c^
7*d^2*m^2 + 967680*a^7*b^3*c^6*f^2*l^2 + 645120*a^7*b^3*c^6*e^2*m^2 + 41472
0*a^7*b^3*c^6*g^2*k^2 + 276480*a^7*b^3*c^6*h^2*j^2 + 207360*a^6*b^5*c^5*g^2
*k^2 + 161280*a^6*b^5*c^5*f^2*l^2 + 140544*a^6*b^5*c^5*h^2*j^2 - 80640*a^6*
b^5*c^5*e^2*m^2 + 25344*a^5*b^7*c^4*h^2*j^2 - 20160*a^5*b^7*c^4*f^2*l^2 + 5
184*a^5*b^7*c^4*g^2*k^2 + 2304*a^5*b^7*c^4*e^2*m^2 + 576*a^4*b^9*c^3*h^2*j^
2 + 576*a^4*b^9*c^3*f^2*l^2 + 7962624*a^7*b^2*c^7*e^2*l^2 - 4148928*a^6*b^4
*c^6*d^2*m^2 + 1419840*a^6*b^4*c^6*f^2*k^2 + 1387008*a^7*b^2*c^7*f^2*k^2 -
1183392*a^5*b^6*c^5*d^2*m^2 + 884736*a^7*b^2*c^7*g^2*j^2 + 884736*a^6*b^4*c
^6*g^2*j^2 + 645750*a^4*b^8*c^4*d^2*m^2 + 221184*a^5*b^6*c^5*g^2*j^2 - 1159
20*a^3*b^10*c^3*d^2*m^2 + 84960*a^5*b^6*c^5*f^2*k^2 + 10836*a^2*b^12*c^2*d^
2*m^2 - 8010*a^4*b^8*c^4*f^2*k^2 - 180*a^3*b^10*c^3*f^2*k^2 + 9*a^2*b^12*c^
2*f^2*k^2 + 8709120*a^6*b^3*c^7*d^2*l^2 - 4354560*a^5*b^5*c^6*d^2*l^2 + 979
776*a^4*b^7*c^5*d^2*l^2 + 829440*a^6*b^3*c^7*e^2*k^2 + 17480448*a^6*b^2*c^8
*d^2*k^2 + 501760*a^6*b^3*c^7*f^2*j^2 + 170240*a^5*b^5*c^6*f^2*j^2 - 108864

*a^3*b^9*c^4*d^2*l^2 + 20736*a^5*b^5*c^6*e^2*k^2 + 9216*a^4*b^7*c^5*f^2*j^2 + 5184*a^2*b^11*c^3*d^2*l^2 - 1984*a^3*b^9*c^4*f^2*j^2 + 64*a^2*b^11*c^3*f ^2*j^2 + 3538944*a^6*b^2*c^8*e^2*j^2 - 3302208*a^5*b^4*c^7*d^2*k^2 + 884736 *a^5*b^4*c^7*e^2*j^2 + 414720*a^6*b^3*c^7*g^2*h^2 + 207360*a^5*b^5*c^6*g^2* h^2 - 103680*a^4*b^6*c^6*d^2*k^2 + 101250*a^3*b^8*c^5*d^2*k^2 - 5751*a^2*b^ 10*c^4*d^2*k^2 + 5184*a^4*b^7*c^5*g^2*h^2 + 1935360*a^5*b^3*c^8*d^2*j^2 + 1 684224*a^6*b^2*c^8*f^2*h^2 + 1264320*a^5*b^4*c^7*f^2*h^2 - 532224*a^4*b^5*c ^7*d^2*j^2 + 126720*a^4*b^6*c^6*f^2*h^2 - 96768*a^3*b^7*c^6*d^2*j^2 + 62784 *a^2*b^9*c^5*d^2*j^2 - 13950*a^3*b^8*c^5*f^2*h^2 + 225*a^2*b^10*c^4*f^2*h^2 + 967680*a^5*b^3*c^8*f^2*g^2 + 829440*a^5*b^3*c^8*e^2*h^2 + 161280*a^4*b^5 *c^7*f^2*g^2 + 20736*a^4*b^5*c^7*e^2*h^2 - 20160*a^3*b^7*c^6*f^2*g^2 + 576* a^2*b^9*c^5*f^2*g^2 + 11487744*a^5*b^2*c^9*d^2*h^2 + 7962624*a^5*b^2*c^9*e^ 2*g^2 + 35525376*a^4*b^2*c^10*d^2*f^2 - 1412640*a^3*b^6*c^7*d^2*h^2 + 46137 6*a^4*b^4*c^8*d^2*h^2 + 375030*a^2*b^8*c^6*d^2*h^2 + 8709120*a^4*b^3*c^9*d^ 2*g^2 - 4354560*a^3*b^5*c^8*d^2*g^2 + 979776*a^2*b^7*c^7*d^2*g^2 + 645120*a ^4*b^3*c^9*e^2*f^2 - 80640*a^3*b^5*c^8*e^2*f^2 + 2304*a^2*b^7*c^7*e^2*f^2 - 15269184*a^3*b^4*c^9*d^2*f^2 + 2870784*a^2*b^6*c^8*d^2*f^2 - 17418240*a^3* b^3*c^10*d^2*e^2 + 3919104*a^2*b^5*c^9*d^2*e^2 + 54*b^15*c*d^2*k*m + 6*a*b^ 15*d*f*m^2 + 115200*a^11*c^5*k^2*m^2 + 576*a^7*b^9*l^2*m^2 + 225*a^6*b^10*k ^2*m^2 + 64*a^5*b^11*j^2*m^2 + 345600*a^10*c^6*h^2*m^2 + 9*a^4*b^12*h^2*m^2 + 320000*a^9*c^7*f^2*m^2 + 41472*a^9*c^7*h^2*k^2 + 16934400*a^8*c^8*d^2*m^ 2 + 345600*a^8*c^8*f^2*k^2 + 81*b^14*c^2*d^2*k^2 + 3538944*a^7*c^9*e^2*j^2 + 2032128*a^7*c^9*d^2*k^2 + 492800*a^11*b^2*c^3*m^4 + 351456*a^10*b^4*c^2*m ^4 + 576*b^13*c^3*d^2*j^2 + 331776*a^9*b^4*c^3*l^4 + 115200*a^7*c^9*f^2*h^2 + 142560*a^8*b^4*c^4*k^4 + 103680*a^9*b^2*c^5*k^4 + 32400*a^7*b^6*c^3*k^4 + 2025*b^12*c^4*d^2*h^2 + 2025*a^6*b^8*c^2*k^4 + 6096384*a^6*c^10*d^2*h^2 + 131072*a^8*b^2*c^6*j^4 + 98304*a^7*b^4*c^5*j^4 + 32768*a^6*b^6*c^4*j^4 + 5 184*b^11*c^5*d^2*g^2 + 4096*a^5*b^8*c^3*j^4 + 11025*b^10*c^6*d^2*f^2 + 5644 800*a^5*c^11*d^2*f^2 + 142560*a^6*b^4*c^6*h^4 + 103680*a^7*b^2*c^7*h^4 + 32 400*a^5*b^6*c^5*h^4 + 20736*b^9*c^7*d^2*e^2 + 2025*a^4*b^8*c^4*h^4 + 331776 *a^5*b^4*c^7*g^4 + 492800*a^5*b^2*c^9*f^4 + 351456*a^4*b^4*c^8*f^4 - 43120* a^3*b^6*c^7*f^4 + 1225*a^2*b^8*c^6*f^4 - 27433728*a^3*b^2*c^11*d^4 + 644630 4*a^2*b^4*c^10*d^4 - 1050*a^7*b^9*k*m^3 + 384000*a^11*c^5*h*m^3 + 138240*a^ 9*c^7*h^3*m + 210*a^6*b^10*h*m^3 + 47416320*a^6*c^10*d^3*m - 1134*b^12*c^4* d^3*m + 70*a^5*b^11*f*m^3 + 2688000*a^10*c^6*d*m^3 + 384000*a^7*c^9*f^3*k + 138240*a^9*c^7*f*k^3 - 3402*b^11*c^5*d^3*k + 210*a^4*b^12*d*m^3 + 7077888* a^6*c^10*e^3*j + 786432*a^8*c^8*e*j^3 - 43120*a^9*b^6*c*m^4 + 28449792*a^5* c^11*d^3*h + 17010*b^10*c^6*d^3*h + 580608*a^7*c^9*d*h^3 - 39690*b^9*c^7*d^ 3*f - 734832*a*b^6*c^9*d^4 + 9*b^16*d^2*m^2 + 160000*a^12*c^4*m^4 + 1225*a^ 8*b^8*m^4 + 20736*a^10*c^6*k^4 + 65536*a^9*c^7*j^4 + 20736*a^8*c^8*h^4 + 49 787136*a^4*c^12*d^4 + 160000*a^6*c^10*f^4 + 5308416*a^5*c^11*e^4 + 35721*b^ 8*c^8*d^4 + a^2*b^14*f^2*m^2, z, k1), k1, 1, 4) - ((8*a^2*c^2*g + a^2*b^2*l + b^3*c*e + 8*a^3*c*l - 10*a*b*c^2*e + a*b^2*c*g - 6*a^2*b*c*j)/(4*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^4*(b^4*l + 9*b^2*c^2*g + 16*a^2*c^2*l - 18* b*c^3*e - 3*b^3*c*j - 6*a*b*c^2*j + a*b^2*c*l))/(4*c*(b^4 + 16*a^2*c^2 - 8*

```
a*b^2*c)) - (x^7*(3*b^3*c^2*d + 20*a^2*c^3*f + 12*a^3*c^2*k + a^2*b^3*m - 2
4*a*b*c^3*d - 16*a^3*b*c*m + a*b^2*c^2*f - 12*a^2*b*c^2*h + 3*a^2*b^2*c*k))
/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^2*(2*a^2*c^2*j - 2*b^2*c^2*e -
 10*a*c^3*e + b^3*c*g + a*b^3*l + 5*a*b*c^2*g - 5*a*b^2*c*j + 5*a^2*b*c*l))
/(2*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (c*x^6*(6*c^2*e + b^2*j - 3*b*c*g +
 2*a*c*j - 3*a*b*l))/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^3*(4*a^4*c^2*k
 - 36*a^3*c^3*f + 2*a^3*b^3*m - 3*b^5*c*d - 5*a^2*b^2*c^2*f - a*b^4*c*f + 2
8*a^4*b*c*m + 20*a*b^3*c^2*d + 4*a^2*b*c^3*d + 5*a^2*b^3*c*h + 16*a^3*b*c^2
*h - 19*a^3*b^2*c*k))/(8*a^2*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x*(12*a^3
*c^2*h - 44*a^2*c^3*d + a^3*b^2*m - 5*b^4*c*d + 20*a^4*c*m + a*b^3*c*f - 12
*a^3*b*c*k + 37*a*b^2*c^2*d - 16*a^2*b*c^2*f + 3*a^2*b^2*c*h))/(8*a*c*(b^4
+ 16*a^2*c^2 - 8*a*b^2*c)) - (x^5*(28*a^2*c^4*d + 6*b^4*c^2*d + 4*a^3*c^3*h
 - a^2*b^4*m - 36*a^4*c^2*m - 19*a^2*b^2*c^2*h - 49*a*b^2*c^3*d + 2*a*b^3*c
^2*f + 28*a^2*b*c^3*f + 5*a^2*b^3*c*k + 16*a^3*b*c^2*k - 5*a^3*b^2*c*m))/(8
*a^2*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8
+ 2*a*b*x^2 + 2*b*c*x^6)
```

3.58 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx$

Optimal result

Integrand size = 50, antiderivative size = 645

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx$$
$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+\left(bc(cd+ah)-ab^2j-2ac(cf-aj)\right)x^2\right)}{2ac\,(b^2-4ac)\,(a+bx^2+cx^4)}$$
$$-\frac{bc(ce+ai)-ab^2k-2ac(cg-ak)+\left(2c^3e-c^2(bg+2ai)-b^3k+bc(bi+3ak)\right)x^2}{2c^2\,(b^2-4ac)\,(a+bx^2+cx^4)}$$
$$+\frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)+\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\,\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)-\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}\,(b^2-4ac)\,\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+\frac{\left(4c^3e-c^2(2bg-4ai)+b^3k-6abck\right)\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2\,(b^2-4ac)^{3/2}}+\frac{k\log\left(a+bx^2+cx^4\right)}{4c^2}$$

```
[Out] 1/2*x*(c*(b^2*d-2*a*(-a*h+c*d)-a*b*(a*j+c*f)/c)+(b*c*(a*h+c*d)-a*b^2*j-2*a*
c*(-a*j+c*f))*x^2)/a/c/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*(-b*c*(a*i+c*e)+a*b
^2*k+2*a*c*(-a*k+c*g)-(2*c^3*e-c^2*(2*a*i+b*g)-b^3*k+b*c*(3*a*k+b*i))*x^2)/
c^2/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*(4*c^3*e-c^2*(-4*a*i+2*b*g)+b^3*k-6*a*
b*c*k)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/c^2/(-4*a*c+b^2)^(3/2)+1/4*k
*ln(c*x^4+b*x^2+a)/c^2+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^
(1/2))*(b*(a*h+c*d)+a*b^2*j/c-2*a*(3*a*j+c*f)+(b^2*c*(-a*h+c*d)-4*a*c^2*(a*
h+3*c*d)-a*b^3*j+4*a*b*c*(2*a*j+c*f))/c/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*
2^(1/2)/c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(
```

```
b+(-4*a*c+b^2)^(1/2))^(1/2))*(b*(a*h+c*d)+a*b^2*j/c-2*a*(3*a*j+c*f)+(-b^2*c
*(-a*h+c*d)+4*a*c^2*(a*h+3*c*d)+a*b^3*j-4*a*b*c*(2*a*j+c*f))/c/(-4*a*c+b^2)
^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 2.16 (sec) , antiderivative size = 645, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.200, Rules used = {1687, 1692, 1180, 211, 1677, 1674, 648, 632, 212, 642}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx$$
$$=\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{ab^2j}{c}+\frac{-ab^3j+b^2c(cd-ah)+4abc(2aj+cf)-4ac^2(ah+3cd)}{c\sqrt{b^2-4ac}}+b(ah+cd)-2a(3aj+cf)\right)}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(\frac{ab^2j}{c}-\frac{-ab^3j+b^2c(cd-ah)+4abc(2aj+cf)-4ac^2(ah+3cd)}{c\sqrt{b^2-4ac}}+b(ah+cd)-2a(3aj+cf)\right)}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{\sqrt{b^2-4ac}+b}}$$
$$+\frac{\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)\left(-c^2(2bg-4ai)-6abck+b^3k+4c^3e\right)}{2c^2\left(b^2-4ac\right)^{3/2}}$$
$$+\frac{x\left(x^2\left(-ab^2j+bc(ah+cd)-2ac(cf-aj)\right)+c\left(-\frac{ab(aj+cf)}{c}-2a(cd-ah)+b^2d\right)\right)}{2ac\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}$$
$$-\frac{x^2\left(-c^2(2ai+bg)+bc(3ak+bi)+b^3(-k)+2c^3e\right)-ab^2k+bc(ai+ce)-2ac(cg-ak)}{2c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)}$$
$$+\frac{k\log\left(a+bx^2+cx^4\right)}{4c^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^6 + k*x^7)/(a + b*x^2 +
c*x^4)^2,x]
```

```
[Out] (x*(c*(b^2*d - 2*a*(c*d - a*h) - (a*b*(c*f + a*j))/c) + (b*c*(c*d + a*h) -
a*b^2*j - 2*a*c*(c*f - a*j))*x^2))/(2*a*c*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4)
) - (b*c*(c*e + a*i) - a*b^2*k - 2*a*c*(c*g - a*k) + (2*c^3*e - c^2*(b*g +
2*a*i) - b^3*k + b*c*(b*i + 3*a*k))*x^2)/(2*c^2*(b^2 - 4*a*c)*(a + b*x^2 +
c*x^4)) + ((b*(c*d + a*h) + (a*b^2*j)/c - 2*a*(c*f + 3*a*j) + (b^2*c*(c*d -
 a*h) - 4*a*c^2*(3*c*d + a*h) - a*b^3*j + 4*a*b*c*(c*f + 2*a*j))/(c*Sqrt[b^
2 - 4*a*c]))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*Sq
rt[2]*a*Sqrt[c]*(b^2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((b*(c*d + a*h
) + (a*b^2*j)/c - 2*a*(c*f + 3*a*j) - (b^2*c*(c*d - a*h) - 4*a*c^2*(3*c*d +
 a*h) - a*b^3*j + 4*a*b*c*(c*f + 2*a*j))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(Sqr
t[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*Sqrt[c]*(b^2 - 4
*a*c)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((4*c^3*e - c^2*(2*b*g - 4*a*i) + b^3*
```

```
k - 6*a*b*c*k)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(2*c^2*(b^2 - 4*a*
c)^(3/2)) + (k*Log[a + b*x^2 + c*x^4])/(4*c^2)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
```

```
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2+hx^4+jx^6}{(a+bx^2+cx^4)^2}\,dx + \int \frac{x(e+gx^2+ix^4+kx^6)}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+\left(bc(cd+ah)-ab^2j-2ac(cf-aj)\right)x^2\right)}{2ac\,(b^2-4ac)\,(a+bx^2+cx^4)} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx+ix^2+kx^3}{(a+bx+cx^2)^2}\,dx, x, x^2\right) - \frac{\int \frac{-b^2d+2a(3cd+ah)-\frac{ab(cf+aj)}{c}+\left(-b(cd+ah)-\frac{ab^2j}{c}+2a(cf+3aj)\right)x^2}{a+bx^2+cx^4}\,dx}{2a\,(b^2-4ac)}$$

$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+\left(bc(cd+ah)-ab^2j-2ac(cf-aj)\right)x^2\right)}{2ac\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{bc(ce+ai)-ab^2k-2ac(cg-ak)+\left(2c^3e-c^2(bg+2ai)-b^3k+bc(bi+3ak)\right)x^2}{2c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{\text{Subst}\left(\int \frac{2ce-bg+2ai-\frac{abk}{c}+\left(4a-\frac{b^2}{c}\right)kx}{a+bx+cx^2}\,dx,x,x^2\right)}{2\left(b^2-4ac\right)} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)-\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,d}{4a\left(b^2-4ac\right)} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)+\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,d}{4a\left(b^2-4ac\right)}$$

$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+\left(bc(cd+ah)-ab^2j-2ac(cf-aj)\right)x^2\right)}{2ac\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} - \frac{bc(ce+ai)-ab^2k-2ac(cg-ak)+\left(2c^3e-c^2(bg+2ai)-b^3k+bc(bi+3ak)\right)x^2}{2c^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)+\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right.}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)-\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right.}{2\sqrt{2}a\sqrt{c}\left(b^2-4ac\right)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{k\text{Subst}\left(\int \frac{b+2cx}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^2} - \frac{\left(4c^3e-c^2(2bg-4ai)+b^3k-6abck\right)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^2\left(b^2-4ac\right)}$$

$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+(bc(cd+ah)-ab^2j-2ac(cf-aj))x^2\right)}{2ac(b^2-4ac)(a+bx^2+cx^4)} - \frac{bc(ce+ai)-ab^2k-2ac(cg-ak)+(2c^3e-c^2(bg+2ai)-b^3k+bc(bi+3ak))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)+\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)-\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{k\log(a+bx^2+cx^4)}{4c^2} + \frac{(4c^3e-c^2(2bg-4ai)+b^3k-6abck)\,\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{2c^2(b^2-4ac)}$$

$$= \frac{x\left(c\left(b^2d-2a(cd-ah)-\frac{ab(cf+aj)}{c}\right)+(bc(cd+ah)-ab^2j-2ac(cf-aj))x^2\right)}{2ac(b^2-4ac)(a+bx^2+cx^4)} - \frac{bc(ce+ai)-ab^2k-2ac(cg-ak)+(2c^3e-c^2(bg+2ai)-b^3k+bc(bi+3ak))x^2}{2c^2(b^2-4ac)(a+bx^2+cx^4)} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)+\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(b(cd+ah)+\frac{ab^2j}{c}-2a(cf+3aj)-\frac{b^2c(cd-ah)-4ac^2(3cd+ah)-ab^3j+4abc(cf+2aj)}{c\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a\sqrt{c}(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(4c^3e-c^2(2bg-4ai)+b^3k-6abck)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^2(b^2-4ac)^{3/2}} + \frac{k\log(a+bx^2+cx^4)}{4c^2}$$

Mathematica [A] (verified)

Time = 2.47 (sec) , antiderivative size = 775, normalized size of antiderivative = 1.20

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{\frac{2(2a^3ck-bc^2dx(b+cx^2)+a(-b^3kx^2+b^2cx^2(i+jx)+2c^3x(d+x(e+fx))+bc^2(e+x(f-x(g+hx))))+a^2(-b^2k+bc(i+x(j+3kx))-2c^2(g+x(h+x(i+j}{a(-b^2+4ac)(a+bx^2+cx^4)}}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^6 + k*x^7)/(a + b*
x^2 + c*x^4)^2,x]
```

```
[Out] ((2*(2*a^3*c*k - b*c^2*d*x*(b + c*x^2) + a*(-(b^3*k*x^2) + b^2*c*x^2*(i + j
*x) + 2*c^3*x*(d + x*(e + f*x)) + b*c^2*(e + x*(f - x*(g + h*x)))) + a^2*(-
(b^2*k) + b*c*(i + x*(j + 3*k*x)) - 2*c^2*(g + x*(h + x*(i + j*x))))))/(a*(
-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)) - (Sqrt[2]*Sqrt[c]*(a*b^3*j - b*c*(c*Sqr
t[b^2 - 4*a*c]*d + 4*a*c*f + a*Sqrt[b^2 - 4*a*c]*h + 8*a^2*j) - b^2*(c^2*d
- a*c*h + a*Sqrt[b^2 - 4*a*c]*j) + 2*a*c*(6*c^2*d + c*Sqrt[b^2 - 4*a*c]*f +
 2*a*c*h + 3*a*Sqrt[b^2 - 4*a*c]*j))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sq
rt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (S
qrt[2]*Sqrt[c]*(a*b^3*j + b*c*(c*Sqrt[b^2 - 4*a*c]*d - 4*a*c*f + a*Sqrt[b^2
 - 4*a*c]*h - 8*a^2*j) + 2*a*c*(6*c^2*d - c*Sqrt[b^2 - 4*a*c]*f + 2*a*c*h -
 3*a*Sqrt[b^2 - 4*a*c]*j) + b^2*(-(c^2*d) + a*c*h + a*Sqrt[b^2 - 4*a*c]*j))
*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^
(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + ((-4*c^3*e + 2*c^2*(b*g - 2*a*i) + b^2
*(-b + Sqrt[b^2 - 4*a*c])*k + a*c*(6*b*k - 4*Sqrt[b^2 - 4*a*c]*k))*Log[-b +
 Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) + ((4*c^3*e + c^2*(-2*b*
g + 4*a*i) + b^2*(b + Sqrt[b^2 - 4*a*c])*k - 2*a*c*(3*b + 2*Sqrt[b^2 - 4*a*
c])*k)*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/(4*c^2)
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.58 (sec) , antiderivative size = 422, normalized size of antiderivative = 0.65

method	result
risch	$\frac{-\frac{\left(2a^2cj-a\,b^2j+abch-2a\,c^2f+b\,c^2d\right)x^3}{2a\left(4ac-b^2\right)c}+\frac{\left(3abck-2a\,c^2i-b^3k+b^2ci-b\,c^2g+2c^3e\right)x^2}{2\left(4ac-b^2\right)c^2}+\frac{\left(a^2bj-2a^2ch+abcf+2a\,c^2d-b^2cd\right)x}{2ac\left(4ac-b^2\right)}+\frac{2a^2ck-a\,b^2k+abc\cdots}{2\left(4ac-\cdots\right.}}{c\,x^4+b\,x^2+a}$
default	Expression too large to display

```
[In] int((k*x^7+j*x^6+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,method=
_RETURNVERBOSE)
```

```
[Out] (-1/2/a*(2*a^2*c*j-a*b^2*j+a*b*c*h-2*a*c^2*f+b*c^2*d)/(4*a*c-b^2)/c*x^3+1/2
*(3*a*b*c*k-2*a*c^2*i-b^3*k+b^2*c*i-b*c^2*g+2*c^3*e)/(4*a*c-b^2)/c^2*x^2+1/
2*(a^2*b*j-2*a^2*c*h+a*b*c*f+2*a*c^2*d-b^2*c*d)/a/c/(4*a*c-b^2)*x+1/2*(2*a^
2*c*k-a*b^2*k+a*b*c*i-2*a*c^2*g+b*c^2*e)/(4*a*c-b^2)/c^2)/(c*x^4+b*x^2+a)+1
/4/c*sum((2*k*_R^3+1/a*(6*a^2*c*j-a*b^2*j-a*b*c*h+2*a*c^2*f-b*c^2*d)/(4*a*c
-b^2)*_R^2-2*(a*b*k-2*a*c*i+b*c*g-2*c^2*e)/(4*a*c-b^2)*_R-1/a*(a^2*b*j-2*a^
2*c*h+a*b*c*f-6*a*c^2*d+b^2*c*d)/(4*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=R
ootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((k*x^7+j*x^6+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,
algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx=\text{Timed out}$$

```
[In] integrate((k*x**7+j*x**6+i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+
a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx = \int \frac{kx^7+jx^6+ix^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^2}\,dx$$

```
[In] integrate((k*x^7+j*x^6+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,
algorithm="maxima")
```

```
[Out] -1/2*(a*b*c^2*e - 2*a^2*c^2*g + a^2*b*c*i - (b*c^3*d - 2*a*c^3*f + a*b*c^2*
h - (a*b^2*c - 2*a^2*c^2)*j)*x^3 + (2*a*c^3*e - a*b*c^2*g + (a*b^2*c - 2*a^
2*c^2)*i - (a*b^3 - 3*a^2*b*c)*k)*x^2 - (a^2*b^2 - 2*a^3*c)*k + (a*b*c^2*f
- 2*a^2*c^2*h + a^2*b*c*j - (b^2*c^2 - 2*a*c^3)*d)*x)/(a^2*b^2*c^2 - 4*a^3*
c^3 + (a*b^2*c^3 - 4*a^2*c^4)*x^4 + (a*b^3*c^2 - 4*a^2*b*c^3)*x^2) - 1/2*in
tegrate(-(2*(a*b^2 - 4*a^2*c)*k*x^3 + a*b*c*f - 2*a^2*c*h + a^2*b*j + (b*c^
2*d - 2*a*c^2*f + a*b*c*h + (a*b^2 - 6*a^2*c)*j)*x^2 + (b^2*c - 6*a*c^2)*d
- 2*(2*a*c^2*e - a*b*c*g + 2*a^2*c*i - a^2*b*k)*x)/(c*x^4 + b*x^2 + a), x)/
(a*b^2*c - 4*a^2*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 16214 vs. $2(594) = 1188$.

Time = 3.25 (sec) , antiderivative size = 16214, normalized size of antiderivative = 25.14

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{\left(a+bx^2+cx^4\right)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((k*x^7+j*x^6+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^2,x,
algorithm="giac")
```

```
[Out] 1/4*k*log(abs(c*x^4 + b*x^2 + a))/c^2 + 1/16*((a^2*b^4*c^3 - 8*a^3*b^2*c^4
+ 16*a^4*c^5)^2*(2*b^3*c^4 - 8*a*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*b*c^4 - 2*(b^2 - 4*a*c)*b*c^4)*d - 2*(a^2*b^4*c^3 - 8*a^3*b^2*c^4 +
16*a^4*c^5)^2*(2*a*b^2*c^4 - 8*a^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a*c^4 - 2*(b^2 - 4*a*c)*a*c^4)*f + (a^2*b^4*c^3 - 8*a^3*b^2*c^4 +
16*a^4*c^5)^2*(2*a*b^3*c^3 - 8*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^2*b*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a*b*c^3 - 2*(b^2 - 4*a*c)*a*b*c^3)*h + (a^2*b^4*c^3 - 8*a^3*b^
2*c^4 + 16*a^4*c^5)^2*(2*a*b^4*c^2 - 20*a^2*b^2*c^3 + 48*a^3*c^4 - sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4 + 10*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c + 2*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c - 24*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^2 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*c^3 - 2*(b^2 - 4*a*c)*a*b^2*c^2 + 12*(b^2 - 4*
a*c)*a^2*c^3)*j + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^8*c^5 -
18*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 - 2*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b^7*c^6 - 2*a^2*b^8*c^6 + 120*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c^7 + 28*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^3*b^5*c^7 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c^7 + 36
*a^3*b^6*c^7 - 352*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 12
8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^8 - 14*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^8 - 240*a^4*b^4*c^8 + 384*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^6*c^9 + 192*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^5*b*c^9 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^9 + 70
```

4*a^5*b^2*c^9 - 96*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*c^10 - 768*a
^6*c^10 + 2*(b^2 - 4*a*c)*a^2*b^6*c^6 - 28*(b^2 - 4*a*c)*a^3*b^4*c^7 + 128*
(b^2 - 4*a*c)*a^4*b^2*c^8 - 192*(b^2 - 4*a*c)*a^5*c^9)*d*abs(a^2*b^4*c^3 -
8*a^3*b^2*c^4 + 16*a^4*c^5) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
3*b^7*c^5 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 - 2*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 - 2*a^3*b^7*c^6 + 48*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 16*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^4*b^4*c^7 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b
^5*c^7 + 24*a^4*b^5*c^7 - 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*
c^8 - 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 8*sqrt(2)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^8 - 96*a^5*b^3*c^8 + 16*sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*c^9 + 128*a^6*b*c^9 + 2*(b^2 - 4*a*c)*a^
3*b^5*c^6 - 16*(b^2 - 4*a*c)*a^4*b^3*c^7 + 32*(b^2 - 4*a*c)*a^5*b*c^8)*f*ab
s(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) - 4*(sqrt(2)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^4*b^6*c^5 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*
b^4*c^6 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 - 2*a^4*b^6
*c^6 + 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 16*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + sqrt(2)*sqrt(b*c + sqrt(b^2 -
 4*a*c)*c)*a^4*b^4*c^7 + 24*a^5*b^4*c^7 - 64*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^7*c^8 - 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 -
8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 96*a^6*b^2*c^8 + 16
*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*c^9 + 128*a^7*c^9 + 2*(b^2 - 4
*a*c)*a^4*b^4*c^6 - 16*(b^2 - 4*a*c)*a^5*b^2*c^7 + 32*(b^2 - 4*a*c)*a^6*c^8
)*h*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + 2*(sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^4*b^7*c^4 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^5*b^5*c^5 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 - 2*a
^4*b^7*c^5 + 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 + 16*sq
rt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 + sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^4*b^5*c^6 + 24*a^5*b^5*c^6 - 64*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^7*b*c^7 - 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*
b^2*c^7 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 - 96*a^6*b^
3*c^7 + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 + 128*a^7*b*c^
8 + 2*(b^2 - 4*a*c)*a^4*b^5*c^5 - 16*(b^2 - 4*a*c)*a^5*b^3*c^6 + 32*(b^2 -
4*a*c)*a^6*b*c^7)*j*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + (2*a^4*
b^11*c^10 - 56*a^5*b^9*c^11 + 576*a^6*b^7*c^12 - 2816*a^7*b^5*c^13 + 6656*a
^8*b^3*c^14 - 6144*a^9*b*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^4*b^11*c^8 + 28*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^5*b^9*c^9 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b
^2 - 4*a*c)*c)*a^4*b^10*c^9 - 288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^6*b^7*c^10 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^5*b^8*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^4*b^9*c^10 + 1408*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^7*b^5*c^11 + 384*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^6*b^6*c^11 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^5*b^7*c^11 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*

c + sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^12 - 1280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^12 - 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^5*c^12 + 3072*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b*c^13 + 1536*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^13 + 640*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^3*c^13 - 768*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b*c^14 - 2*(b^2 - 4*a*c)*a^4*b^9*
c^10 + 48*(b^2 - 4*a*c)*a^5*b^7*c^11 - 384*(b^2 - 4*a*c)*a^6*b^5*c^12 + 128
0*(b^2 - 4*a*c)*a^7*b^3*c^13 - 1536*(b^2 - 4*a*c)*a^8*b*c^14)*d + 4*(2*a^5*
b^10*c^10 - 32*a^6*b^8*c^11 + 192*a^7*b^6*c^12 - 512*a^8*b^4*c^13 + 512*a^9
*b^2*c^14 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b
^10*c^8 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*
b^8*c^9 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b
^9*c^9 - 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b
^6*c^10 - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*
b^7*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^
8*c^10 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*
b^4*c^11 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7
*b^5*c^11 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
6*b^6*c^11 - 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^9*b^2*c^12 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^8*b^3*c^12 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^7*b^4*c^12 + 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^8*b^2*c^13 - 2*(b^2 - 4*a*c)*a^5*b^8*c^10 + 24*(b^2 - 4*a*c)*a^6*b^6*
c^11 - 96*(b^2 - 4*a*c)*a^7*b^4*c^12 + 128*(b^2 - 4*a*c)*a^8*b^2*c^13)*f -
(2*a^5*b^11*c^9 - 24*a^6*b^9*c^10 + 64*a^7*b^7*c^11 + 256*a^8*b^5*c^12 - 15
36*a^9*b^3*c^13 + 2048*a^10*b*c^14 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^5*b^11*c^7 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^8 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^5*b^10*c^8 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^9 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^9 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^5*b^9*c^9 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^8*b^5*c^10 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^6*b^7*c^10 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^11 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^11 - 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^12 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^12 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^12 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b*c^13 - 2*(b^2 - 4*a*c)*a^5*b^9*c^9 + 1
6*(b^2 - 4*a*c)*a^6*b^7*c^10 - 256*(b^2 - 4*a*c)*a^8*b^3*c^12 + 512*(b^2 -
4*a*c)*a^9*b*c^13)*h - (2*a^5*b^12*c^8 - 48*a^6*b^10*c^9 + 448*a^7*b^8*c^10
 - 2048*a^8*b^6*c^11 + 4608*a^9*b^4*c^12 - 4096*a^10*b^2*c^13 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^6 + 24*sqrt(2)*sq

rt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^7 + 2*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^7 - 224*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^8 - 40*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^8 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^8 + 1024*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^9 + 288*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^9 + 20*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^9 - 2304*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^10 - 896*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^10 - 144*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^10 + 204
8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^11 +
 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^1
1 + 448*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c
^11 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2
c^12 - 2(b^2 - 4*a*c)*a^5*b^10*c^8 + 40*(b^2 - 4*a*c)*a^6*b^8*c^9 - 288*(
b^2 - 4*a*c)*a^7*b^6*c^10 + 896*(b^2 - 4*a*c)*a^8*b^4*c^11 - 1024*(b^2 - 4*
a*c)*a^9*b^2*c^12)*j)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5*c^3 - 8*a^3*b^3*c^
4 + 16*a^4*b*c^5 + sqrt((a^2*b^5*c^3 - 8*a^3*b^3*c^4 + 16*a^4*b*c^5)^2 - 4*
(a^3*b^4*c^3 - 8*a^4*b^2*c^4 + 16*a^5*c^5)*(a^2*b^4*c^4 - 8*a^3*b^2*c^5 + 1
6*a^4*c^6)))/(a^2*b^4*c^4 - 8*a^3*b^2*c^5 + 16*a^4*c^6)))/((a^4*b^8*c^5 - 1
6*a^5*b^6*c^6 - 2*a^4*b^7*c^6 + 96*a^6*b^4*c^7 + 24*a^5*b^5*c^7 + a^4*b^6*c
^7 - 256*a^7*b^2*c^8 - 96*a^6*b^3*c^8 - 12*a^5*b^4*c^8 + 256*a^8*c^9 + 128*
a^7*b*c^9 + 48*a^6*b^2*c^9 - 64*a^7*c^10)*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 +
 16*a^4*c^5)*abs(c)) - 1/16*((a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5)^2*(
2*b^3*c^4 - 8*a*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2
*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^4 - 2*
(b^2 - 4*a*c)*b*c^4)*d - 2*(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5)^2*(2*
a*b^2*c^4 - 8*a^2*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a*b^2*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^4 -
2*(b^2 - 4*a*c)*a*c^4)*f + (a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5)^2*(2*
a*b^3*c^3 - 8*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a*b^3*c + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^2*b*c^2 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*
c^3 - 2*(b^2 - 4*a*c)*a*b*c^3)*h + (a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^
5)^2*(2*a*b^4*c^2 - 20*a^2*b^2*c^3 + 48*a^3*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4 + 10*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 - sqrt(b^2 - 4*a*c)*c)*a*b^3*c - 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^3*c^2 - 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(

b^2 - 4*a*c)*c)*a^2*b*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a*b^2*c^2 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^2*c^3 - 2*(b^2 - 4*a*c)*a*b^2*c^2 + 12*(b^2 - 4*a*c)*a^2*c^3)*j -
2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^8*c^5 - 18*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a^2*b^7*c^6 + 2*a^2*b^8*c^6 + 120*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^4*b^4*c^7 + 28*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^7
+ sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c^7 - 36*a^3*b^6*c^7 - 35
2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 128*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^8 - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a^3*b^4*c^8 + 240*a^4*b^4*c^8 + 384*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^6*c^9 + 192*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^9 + 6
4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^9 - 704*a^5*b^2*c^9 - 9
6*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*c^10 + 768*a^6*c^10 - 2*(b^2
- 4*a*c)*a^2*b^6*c^6 + 28*(b^2 - 4*a*c)*a^3*b^4*c^7 - 128*(b^2 - 4*a*c)*a^4
*b^2*c^8 + 192*(b^2 - 4*a*c)*a^5*c^9)*d*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 1
6*a^4*c^5) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^7*c^5 - 12*sq
rt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 - 2*sqrt(2)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^3*b^6*c^6 + 2*a^3*b^7*c^6 + 48*sqrt(2)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^4*b^4*c^7 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c^7 - 24*a^4*b
^5*c^7 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 - 32*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^4*b^3*c^8 + 96*a^5*b^3*c^8 + 16*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^5*b*c^9 - 128*a^6*b*c^9 - 2*(b^2 - 4*a*c)*a^3*b^5*c^6 + 16*(b
^2 - 4*a*c)*a^4*b^3*c^7 - 32*(b^2 - 4*a*c)*a^5*b*c^8)*f*abs(a^2*b^4*c^3 - 8
*a^3*b^2*c^4 + 16*a^4*c^5) + 4*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4
*b^6*c^5 - 12*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 - 2*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^5*c^6 + 2*a^4*b^6*c^6 + 48*sqrt(2)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 + 16*sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^5*b^3*c^7 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^
4*c^7 - 24*a^5*b^4*c^7 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*c^8
- 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 - 8*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^8 + 96*a^6*b^2*c^8 + 16*sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^6*c^9 - 128*a^7*c^9 - 2*(b^2 - 4*a*c)*a^4*b^4*c^6
+ 16*(b^2 - 4*a*c)*a^5*b^2*c^7 - 32*(b^2 - 4*a*c)*a^6*c^8)*h*abs(a^2*b^4*c
^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^4*b^7*c^4 - 12*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^5*c^5 - 2
*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^6*c^5 + 2*a^4*b^7*c^5 + 48*s
qrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^3*c^6 + 16*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^5*b^4*c^6 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^4*b^5*c^6 - 24*a^5*b^5*c^6 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^7*b*c^7 - 32*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^2*c^7 - 8*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^3*c^7 + 96*a^6*b^3*c^7 + 16*sqrt(2
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b*c^8 - 128*a^7*b*c^8 - 2*(b^2 - 4*a*

c)*a^4*b^5*c^5 + 16*(b^2 - 4*a*c)*a^5*b^3*c^6 - 32*(b^2 - 4*a*c)*a^6*b*c^7)
*j*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + (2*a^4*b^11*c^10 - 56*a^
5*b^9*c^11 + 576*a^6*b^7*c^12 - 2816*a^7*b^5*c^13 + 6656*a^8*b^3*c^14 - 614
4*a^9*b*c^15 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
4*b^11*c^8 + 28*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^5*b^9*c^9 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
4*b^10*c^9 - 288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^6*b^7*c^10 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)
*a^5*b^8*c^10 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a
^4*b^9*c^10 + 1408*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^7*b^5*c^11 + 384*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^6*b^6*c^11 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^5*b^7*c^11 - 3328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^8*b^3*c^12 - 1280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^7*b^4*c^12 - 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^6*b^5*c^12 + 3072*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^9*b*c^13 + 1536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a^8*b^2*c^13 + 640*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^7*b^3*c^13 - 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^8*b*c^14 - 2*(b^2 - 4*a*c)*a^4*b^9*c^10 + 48*(b^2 -
4*a*c)*a^5*b^7*c^11 - 384*(b^2 - 4*a*c)*a^6*b^5*c^12 + 1280*(b^2 - 4*a*c)*a
^7*b^3*c^13 - 1536*(b^2 - 4*a*c)*a^8*b*c^14)*d + 4*(2*a^5*b^10*c^10 - 32*a^
6*b^8*c^11 + 192*a^7*b^6*c^12 - 512*a^8*b^4*c^13 + 512*a^9*b^2*c^14 - sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^8 + 16*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^9 + 2*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^9*c^9 - 96*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^10 - 24*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^7*c^10 - sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^8*c^10 + 256*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^11 + 96*sqr
t(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^5*c^11 + 12*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^6*c^11 - 256*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^12 - 12
8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^3*c^12 -
48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^4*c^12 +
64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^2*c^13
- 2*(b^2 - 4*a*c)*a^5*b^8*c^10 + 24*(b^2 - 4*a*c)*a^6*b^6*c^11 - 96*(b^2 -
4*a*c)*a^7*b^4*c^12 + 128*(b^2 - 4*a*c)*a^8*b^2*c^13)*f - (2*a^5*b^11*c^9 -
24*a^6*b^9*c^10 + 64*a^7*b^7*c^11 + 256*a^8*b^5*c^12 - 1536*a^9*b^3*c^13 +
2048*a^10*b*c^14 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^5*b^11*c^7 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^6*b^9*c^8 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^5*b^10*c^8 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^7*b^7*c^9 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^6*b^8*c^9 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)

*a^5*b^9*c^9 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^8*b^5*c^10 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^6*b^7*c^10 + 768*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)
*c)*a^9*b^3*c^11 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^8*b^4*c^11 - 1024*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^10*b*c^12 - 512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^9*b^2*c^12 - 128*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^8*b^3*c^12 + 256*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^9*b*c^13 - 2*(b^2 - 4*a*c)*a^5*b^9*c^9 + 16*(b^2 - 4*a*c)*a
^6*b^7*c^10 - 256*(b^2 - 4*a*c)*a^8*b^3*c^12 + 512*(b^2 - 4*a*c)*a^9*b*c^13
)*h - (2*a^5*b^12*c^8 - 48*a^6*b^10*c^9 + 448*a^7*b^8*c^10 - 2048*a^8*b^6*c
^11 + 4608*a^9*b^4*c^12 - 4096*a^10*b^2*c^13 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^6 + 24*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^7 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^7 - 224*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^8 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^8 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^8 + 1024*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^9 + 288*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^9 + 20*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^9 - 2304*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^10 - 896*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^10 - 144*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^10 + 2048*sqrt(2)*sqrt(b^
2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^11 + 1024*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^11 + 448*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^11 - 512*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^12 - 2*(b^2 -
4*a*c)*a^5*b^10*c^8 + 40*(b^2 - 4*a*c)*a^6*b^8*c^9 - 288*(b^2 - 4*a*c)*a^7*
b^6*c^10 + 896*(b^2 - 4*a*c)*a^8*b^4*c^11 - 1024*(b^2 - 4*a*c)*a^9*b^2*c^12
)*j)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5*c^3 - 8*a^3*b^3*c^4 + 16*a^4*b*c^5
- sqrt((a^2*b^5*c^3 - 8*a^3*b^3*c^4 + 16*a^4*b*c^5)^2 - 4*(a^3*b^4*c^3 - 8*
a^4*b^2*c^4 + 16*a^5*c^5)*(a^2*b^4*c^4 - 8*a^3*b^2*c^5 + 16*a^4*c^6)))/(a^2
*b^4*c^4 - 8*a^3*b^2*c^5 + 16*a^4*c^6)))/((a^4*b^8*c^5 - 16*a^5*b^6*c^6 - 2
*a^4*b^7*c^6 + 96*a^6*b^4*c^7 + 24*a^5*b^5*c^7 + a^4*b^6*c^7 - 256*a^7*b^2*
c^8 - 96*a^6*b^3*c^8 - 12*a^5*b^4*c^8 + 256*a^8*c^9 + 128*a^7*b*c^9 + 48*a^
6*b^2*c^9 - 64*a^7*c^10)*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5)*abs(
c)) - 1/16*(4*(b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5 + (b^2*c^3 - 4*a*c^4
- 2*b*c^4 + c^5)*sqrt(b^2 - 4*a*c))*e*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16
*a^4*c^5) - 2*(b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4 + (b^3*c^2 - 4*a
*b*c^3 - 2*b^2*c^3 + b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(a^2*b^4*c^3 - 8*a^3*b^
2*c^4 + 16*a^4*c^5) + 4*(a*b^3*c^2 - 4*a^2*b*c^3 - 2*a*b^2*c^3 + a*b*c^4 +
(a*b^2*c^2 - 4*a^2*c^3 - 2*a*b*c^3 + a*c^4)*sqrt(b^2 - 4*a*c))*i*abs(a^2*b^
4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + (b^6 - 10*a*b^4*c - 2*b^5*c + 24*a^2*
b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 6*a*b^2*c^3 + (b^5 - 10*a*b^3*c - 2*b^4*

c + 24*a^2*b*c^2 + 12*a*b^2*c^2 + b^3*c^2 - 6*a*b*c^3)*sqrt(b^2 - 4*a*c))*k
*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) - 4*(a^2*b^7*c^6 - 12*a^3*b^
5*c^7 - 2*a^2*b^6*c^7 + 48*a^4*b^3*c^8 + 16*a^3*b^4*c^8 + a^2*b^5*c^8 - 64*
a^5*b*c^9 - 32*a^4*b^2*c^9 - 8*a^3*b^3*c^9 + 16*a^4*b*c^10 - (a^2*b^6*c^6 -
 8*a^3*b^4*c^7 - 2*a^2*b^5*c^7 + 16*a^4*b^2*c^8 + 8*a^3*b^3*c^8 + a^2*b^4*c
^8 - 4*a^3*b^2*c^9)*sqrt(b^2 - 4*a*c))*e + 2*(a^2*b^8*c^5 - 12*a^3*b^6*c^6
- 2*a^2*b^7*c^6 + 48*a^4*b^4*c^7 + 16*a^3*b^5*c^7 + a^2*b^6*c^7 - 64*a^5*b^
2*c^8 - 32*a^4*b^3*c^8 - 8*a^3*b^4*c^8 + 16*a^4*b^2*c^9 + (a^2*b^7*c^5 - 8*
a^3*b^5*c^6 - 2*a^2*b^6*c^6 + 16*a^4*b^3*c^7 + 8*a^3*b^4*c^7 + a^2*b^5*c^7
- 4*a^3*b^3*c^8)*sqrt(b^2 - 4*a*c))*g - 4*(a^3*b^7*c^5 - 12*a^4*b^5*c^6 - 2
*a^3*b^6*c^6 + 48*a^5*b^3*c^7 + 16*a^4*b^4*c^7 + a^3*b^5*c^7 - 64*a^6*b*c^8
 - 32*a^5*b^2*c^8 - 8*a^4*b^3*c^8 + 16*a^5*b*c^9 - (a^3*b^6*c^5 - 8*a^4*b^4
*c^6 - 2*a^3*b^5*c^6 + 16*a^5*b^2*c^7 + 8*a^4*b^3*c^7 + a^3*b^4*c^7 - 4*a^4
*b^2*c^8)*sqrt(b^2 - 4*a*c))*i - (a^2*b^10*c^3 - 18*a^3*b^8*c^4 - 2*a^2*b^9
*c^4 + 120*a^4*b^6*c^5 + 28*a^3*b^7*c^5 + a^2*b^8*c^5 - 352*a^5*b^4*c^6 - 1
28*a^4*b^5*c^6 - 14*a^3*b^6*c^6 + 384*a^6*b^2*c^7 + 192*a^5*b^3*c^7 + 64*a^
4*b^4*c^7 - 96*a^5*b^2*c^8 - (a^2*b^9*c^3 - 14*a^3*b^7*c^4 - 2*a^2*b^8*c^4
+ 64*a^4*b^5*c^5 + 20*a^3*b^6*c^5 + a^2*b^7*c^5 - 96*a^5*b^3*c^6 - 48*a^4*b
^4*c^6 - 10*a^3*b^5*c^6 + 24*a^4*b^3*c^7)*sqrt(b^2 - 4*a*c))*k)*log(x^2 + 1
/2*(a^2*b^5*c^3 - 8*a^3*b^3*c^4 + 16*a^4*b*c^5 + sqrt((a^2*b^5*c^3 - 8*a^3*
b^3*c^4 + 16*a^4*b*c^5)^2 - 4*(a^3*b^4*c^3 - 8*a^4*b^2*c^4 + 16*a^5*c^5)*(a
^2*b^4*c^4 - 8*a^3*b^2*c^5 + 16*a^4*c^6)))/(a^2*b^4*c^4 - 8*a^3*b^2*c^5 + 1
6*a^4*c^6))/((a*b^4*c - 8*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^2*b*
c^3 + a*b^2*c^3 - 4*a^2*c^4)*c^2*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c
^5)) - 1/16*(4*(b^3*c^3 - 4*a*b*c^4 - 2*b^2*c^4 + b*c^5 + (b^2*c^3 - 4*a*c^
4 - 2*b*c^4 + c^5)*sqrt(b^2 - 4*a*c))*e*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 1
6*a^4*c^5) - 2*(b^4*c^2 - 4*a*b^2*c^3 - 2*b^3*c^3 + b^2*c^4 - (b^3*c^2 - 4*
a*b*c^3 - 2*b^2*c^3 + b*c^4)*sqrt(b^2 - 4*a*c))*g*abs(a^2*b^4*c^3 - 8*a^3*b
^2*c^4 + 16*a^4*c^5) + 4*(a*b^3*c^2 - 4*a^2*b*c^3 - 2*a*b^2*c^3 + a*b*c^4 -
 (a*b^2*c^2 - 4*a^2*c^3 - 2*a*b*c^3 + a*c^4)*sqrt(b^2 - 4*a*c))*i*abs(a^2*b
^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + (b^6 - 10*a*b^4*c - 2*b^5*c + 24*a^2
*b^2*c^2 + 12*a*b^3*c^2 + b^4*c^2 - 6*a*b^2*c^3 - (b^5 - 10*a*b^3*c - 2*b^4
*c + 24*a^2*b*c^2 + 12*a*b^2*c^2 + b^3*c^2 - 6*a*b*c^3)*sqrt(b^2 - 4*a*c))*
k*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*c^5) + 4*(a^2*b^7*c^6 - 12*a^3*b
^5*c^7 - 2*a^2*b^6*c^7 + 48*a^4*b^3*c^8 + 16*a^3*b^4*c^8 + a^2*b^5*c^8 - 64
*a^5*b*c^9 - 32*a^4*b^2*c^9 - 8*a^3*b^3*c^9 + 16*a^4*b*c^10 + (a^2*b^6*c^6
- 8*a^3*b^4*c^7 - 2*a^2*b^5*c^7 + 16*a^4*b^2*c^8 + 8*a^3*b^3*c^8 + a^2*b^4*
c^8 - 4*a^3*b^2*c^9)*sqrt(b^2 - 4*a*c))*e - 2*(a^2*b^8*c^5 - 12*a^3*b^6*c^6
 - 2*a^2*b^7*c^6 + 48*a^4*b^4*c^7 + 16*a^3*b^5*c^7 + a^2*b^6*c^7 - 64*a^5*b
^2*c^8 - 32*a^4*b^3*c^8 - 8*a^3*b^4*c^8 + 16*a^4*b^2*c^9 - (a^2*b^7*c^5 - 8
*a^3*b^5*c^6 - 2*a^2*b^6*c^6 + 16*a^4*b^3*c^7 + 8*a^3*b^4*c^7 + a^2*b^5*c^7
 - 4*a^3*b^3*c^8)*sqrt(b^2 - 4*a*c))*g + 4*(a^3*b^7*c^5 - 12*a^4*b^5*c^6 -
2*a^3*b^6*c^6 + 48*a^5*b^3*c^7 + 16*a^4*b^4*c^7 + a^3*b^5*c^7 - 64*a^6*b*c^
8 - 32*a^5*b^2*c^8 - 8*a^4*b^3*c^8 + 16*a^5*b*c^9 - (a^3*b^6*c^5 - 8*a^4*b^
4*c^6 - 2*a^3*b^5*c^6 + 16*a^5*b^2*c^7 + 8*a^4*b^3*c^7 + a^3*b^4*c^7 - 4*a^

4*b^2*c^8)*sqrt(b^2 - 4*a*c))*i + (a^2*b^10*c^3 - 18*a^3*b^8*c^4 - 2*a^2*b^
9*c^4 + 120*a^4*b^6*c^5 + 28*a^3*b^7*c^5 + a^2*b^8*c^5 - 352*a^5*b^4*c^6 -
128*a^4*b^5*c^6 - 14*a^3*b^6*c^6 + 384*a^6*b^2*c^7 + 192*a^5*b^3*c^7 + 64*a
^4*b^4*c^7 - 96*a^5*b^2*c^8 - (a^2*b^9*c^3 - 14*a^3*b^7*c^4 - 2*a^2*b^8*c^4
 + 64*a^4*b^5*c^5 + 20*a^3*b^6*c^5 + a^2*b^7*c^5 - 96*a^5*b^3*c^6 - 48*a^4*
b^4*c^6 - 10*a^3*b^5*c^6 + 24*a^4*b^3*c^7)*sqrt(b^2 - 4*a*c))*k)*log(x^2 +
1/2*(a^2*b^5*c^3 - 8*a^3*b^3*c^4 + 16*a^4*b*c^5 - sqrt((a^2*b^5*c^3 - 8*a^3
*b^3*c^4 + 16*a^4*b*c^5)^2 - 4*(a^3*b^4*c^3 - 8*a^4*b^2*c^4 + 16*a^5*c^5)*(
a^2*b^4*c^4 - 8*a^3*b^2*c^5 + 16*a^4*c^6)))/(a^2*b^4*c^4 - 8*a^3*b^2*c^5 +
16*a^4*c^6))/((a*b^4*c - 8*a^2*b^2*c^2 - 2*a*b^3*c^2 + 16*a^3*c^3 + 8*a^2*b
*c^3 + a*b^2*c^3 - 4*a^2*c^4)*c^2*abs(a^2*b^4*c^3 - 8*a^3*b^2*c^4 + 16*a^4*
c^5)) - 1/2*(a*b*c^2*e - 2*a^2*c^2*g + a^2*b*c*i - a^2*b^2*k + 2*a^3*c*k -
(b*c^3*d - 2*a*c^3*f + a*b*c^2*h - a*b^2*c*j + 2*a^2*c^2*j)*x^3 + (2*a*c^3*
e - a*b*c^2*g + a*b^2*c*i - 2*a^2*c^2*i - a*b^3*k + 3*a^2*b*c*k)*x^2 - (b^2
*c^2*d - 2*a*c^3*d - a*b*c^2*f + 2*a^2*c^2*h - a^2*b*c*j)*x)/((c*x^4 + b*x^
2 + a)*(b^2 - 4*a*c)*a*c^2)

Mupad [B] (verification not implemented)

Time = 13.37 (sec) , antiderivative size = 53538, normalized size of antiderivative = 83.00

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^6+kx^7}{(a+bx^2+cx^4)^2}\,dx = \text{Too large to display}$$

[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^6 + k*x^7)/(a + b*x^2 +
c*x^4)^2,x)

[Out] ((b*c^2*e - 2*a*c^2*g - a*b^2*k + 2*a^2*c*k + a*b*c*i)/(2*c^2*(4*a*c - b^2)
) + (x^2*(2*c^3*e - b^3*k - b*c^2*g - 2*a*c^2*i + b^2*c*i + 3*a*b*c*k))/(2*
c^2*(4*a*c - b^2)) + (x*(2*a*c^2*d - b^2*c*d - 2*a^2*c*h + a^2*b*j + a*b*c*
f))/(2*a*c*(4*a*c - b^2)) - (x^3*(b*c^2*d - 2*a*c^2*f - a*b^2*j + 2*a^2*c*j
 + a*b*c*h))/(2*a*c*(4*a*c - b^2)))/(a + b*x^2 + c*x^4) + symsum(log(root(1
572864*a^8*b^2*c^9*z^4 - 983040*a^7*b^4*c^8*z^4 + 327680*a^6*b^6*c^7*z^4 -
61440*a^5*b^8*c^6*z^4 + 6144*a^4*b^10*c^5*z^4 - 256*a^3*b^12*c^4*z^4 - 1048
576*a^9*c^10*z^4 - 1572864*a^8*b^2*c^7*k*z^3 + 983040*a^7*b^4*c^6*k*z^3 - 3
27680*a^6*b^6*c^5*k*z^3 + 61440*a^5*b^8*c^4*k*z^3 - 6144*a^4*b^10*c^3*k*z^3
 + 256*a^3*b^12*c^2*k*z^3 + 1048576*a^9*c^8*k*z^3 + 98304*a^8*b*c^6*i*k*z^2
 + 98304*a^7*b*c^7*e*k*z^2 + 57344*a^7*b*c^7*f*j*z^2 + 32768*a^7*b*c^7*g*i*
z^2 + 57344*a^6*b*c^8*d*h*z^2 + 32768*a^6*b*c^8*e*g*z^2 - 32*a*b^10*c^4*d*f
*z^2 - 90112*a^7*b^3*c^5*i*k*z^2 + 30720*a^6*b^5*c^4*i*k*z^2 - 4608*a^5*b^7
*c^3*i*k*z^2 + 256*a^4*b^9*c^2*i*k*z^2 - 49152*a^7*b^2*c^6*g*k*z^2 + 45056*
a^6*b^4*c^5*g*k*z^2 + 24576*a^7*b^2*c^6*h*j*z^2 - 15360*a^5*b^6*c^4*g*k*z^2
 - 3072*a^5*b^6*c^4*h*j*z^2 + 2304*a^4*b^8*c^3*g*k*z^2 + 2048*a^6*b^4*c^5*h
*j*z^2 + 576*a^4*b^8*c^3*h*j*z^2 - 128*a^3*b^10*c^2*g*k*z^2 - 32*a^3*b^10*c
^2*h*j*z^2 - 90112*a^6*b^3*c^6*e*k*z^2 - 49152*a^6*b^3*c^6*f*j*z^2 + 30720*

a^5*b^5*c^5*e*k*z^2 - 24576*a^6*b^3*c^6*g*i*z^2 + 15360*a^5*b^5*c^5*f*j*z^2
 + 6144*a^5*b^5*c^5*g*i*z^2 - 4608*a^4*b^7*c^4*e*k*z^2 - 2048*a^4*b^7*c^4*f
*j*z^2 - 512*a^4*b^7*c^4*g*i*z^2 + 256*a^3*b^9*c^3*e*k*z^2 + 96*a^3*b^9*c^3
*f*j*z^2 + 131072*a^6*b^2*c^7*d*j*z^2 + 49152*a^6*b^2*c^7*e*i*z^2 - 43008*a
^5*b^4*c^6*d*j*z^2 - 12288*a^5*b^4*c^6*e*i*z^2 + 6144*a^5*b^4*c^6*f*h*z^2 +
 6144*a^4*b^6*c^5*d*j*z^2 - 2048*a^4*b^6*c^5*f*h*z^2 + 1024*a^4*b^6*c^5*e*i
*z^2 - 320*a^3*b^8*c^4*d*j*z^2 + 192*a^3*b^8*c^4*f*h*z^2 - 49152*a^5*b^3*c^
7*d*h*z^2 - 24576*a^5*b^3*c^7*e*g*z^2 + 15360*a^4*b^5*c^6*d*h*z^2 + 6144*a^
4*b^5*c^6*e*g*z^2 - 2048*a^3*b^7*c^5*d*h*z^2 - 512*a^3*b^7*c^5*e*g*z^2 + 96
*a^2*b^9*c^4*d*h*z^2 + 24576*a^5*b^2*c^8*d*f*z^2 - 3072*a^3*b^6*c^6*d*f*z^2
 + 2048*a^4*b^4*c^7*d*f*z^2 + 576*a^2*b^8*c^5*d*f*z^2 + 1536*a^4*b^10*c*k^2
*z^2 + 61440*a^8*b*c^6*j^2*z^2 - 16*a^3*b^11*c*j^2*z^2 + 12288*a^7*b*c^7*h^
2*z^2 + 12288*a^6*b*c^8*f^2*z^2 + 61440*a^5*b*c^9*d^2*z^2 + 432*a*b^9*c^5*d
^2*z^2 - 49152*a^8*c^7*h*j*z^2 - 147456*a^7*c^8*d*j*z^2 - 65536*a^7*c^8*e*i
*z^2 - 16384*a^7*c^8*f*h*z^2 - 49152*a^6*c^9*d*f*z^2 + 516096*a^8*b^2*c^5*k
^2*z^2 - 288768*a^7*b^4*c^4*k^2*z^2 + 88576*a^6*b^6*c^3*k^2*z^2 - 15744*a^5
*b^8*c^2*k^2*z^2 - 61440*a^7*b^3*c^5*j^2*z^2 + 24064*a^6*b^5*c^4*j^2*z^2 -
4608*a^5*b^7*c^3*j^2*z^2 + 432*a^4*b^9*c^2*j^2*z^2 + 24576*a^7*b^2*c^6*i^2*
z^2 - 6144*a^6*b^4*c^5*i^2*z^2 + 512*a^5*b^6*c^4*i^2*z^2 - 8192*a^6*b^3*c^6
*h^2*z^2 + 1536*a^5*b^5*c^5*h^2*z^2 - 16*a^3*b^9*c^3*h^2*z^2 - 8192*a^6*b^2
*c^7*g^2*z^2 + 6144*a^5*b^4*c^6*g^2*z^2 - 1536*a^4*b^6*c^5*g^2*z^2 + 128*a^
3*b^8*c^4*g^2*z^2 - 8192*a^5*b^3*c^7*f^2*z^2 + 1536*a^4*b^5*c^6*f^2*z^2 - 1
6*a^2*b^9*c^4*f^2*z^2 + 24576*a^5*b^2*c^8*e^2*z^2 - 6144*a^4*b^4*c^7*e^2*z^
2 + 512*a^3*b^6*c^6*e^2*z^2 - 61440*a^4*b^3*c^8*d^2*z^2 + 24064*a^3*b^5*c^7
*d^2*z^2 - 4608*a^2*b^7*c^6*d^2*z^2 - 393216*a^9*c^6*k^2*z^2 - 64*a^3*b^12*
k^2*z^2 - 32768*a^8*c^7*i^2*z^2 - 32768*a^6*c^9*e^2*z^2 - 16*b^11*c^4*d^2*z
^2 - 16384*a^7*b*c^5*g*i*k*z - 10240*a^7*b*c^5*f*j*k*z + 4096*a^7*b*c^5*h*i
*j*z - 47104*a^6*b*c^6*d*h*k*z - 16384*a^6*b*c^6*e*g*k*z + 6144*a^6*b*c^6*f
*g*j*z + 4096*a^6*b*c^6*e*h*j*z + 32*a*b^10*c^2*d*f*k*z - 6144*a^5*b*c^7*d*
g*h*z - 4096*a^5*b*c^7*d*f*i*z - 32*a*b^8*c^4*d*f*g*z - 4096*a^4*b*c^8*d*e*
f*z + 64*a*b^7*c^5*d*e*f*z - 18432*a^7*b^2*c^4*h*j*k*z + 4608*a^6*b^4*c^3*h
*j*k*z - 384*a^5*b^6*c^2*h*j*k*z + 12288*a^6*b^3*c^4*g*i*k*z + 7680*a^6*b^3
*c^4*f*j*k*z - 3072*a^6*b^3*c^4*h*i*j*z - 3072*a^5*b^5*c^3*g*i*k*z - 1920*a
^5*b^5*c^3*f*j*k*z + 768*a^5*b^5*c^3*h*i*j*z + 256*a^4*b^7*c^2*g*i*k*z + 16
0*a^4*b^7*c^2*f*j*k*z - 64*a^4*b^7*c^2*h*i*j*z - 65536*a^6*b^2*c^5*d*j*k*z
- 24576*a^6*b^2*c^5*e*i*k*z + 21504*a^5*b^4*c^4*d*j*k*z + 9216*a^6*b^2*c^5*
f*i*j*z + 6144*a^5*b^4*c^4*e*i*k*z - 3072*a^5*b^4*c^4*f*h*k*z - 3072*a^4*b^
6*c^3*d*j*k*z - 2304*a^5*b^4*c^4*f*i*j*z - 2048*a^6*b^2*c^5*g*h*j*z + 1536*
a^5*b^4*c^4*g*h*j*z + 1024*a^4*b^6*c^3*f*h*k*z - 512*a^4*b^6*c^3*e*i*k*z -
384*a^4*b^6*c^3*g*h*j*z + 192*a^4*b^6*c^3*f*i*j*z + 160*a^3*b^8*c^2*d*j*k*z
 - 96*a^3*b^8*c^2*f*h*k*z + 32*a^3*b^8*c^2*g*h*j*z + 41472*a^5*b^3*c^5*d*h*
k*z - 13440*a^4*b^5*c^4*d*h*k*z + 12288*a^5*b^3*c^5*e*g*k*z - 4608*a^5*b^3*
c^5*f*g*j*z - 3072*a^5*b^3*c^5*e*h*j*z - 3072*a^4*b^5*c^4*e*g*k*z + 1888*a^
3*b^7*c^3*d*h*k*z + 1152*a^4*b^5*c^4*f*g*j*z + 768*a^4*b^5*c^4*e*h*j*z + 25
6*a^3*b^7*c^3*e*g*k*z - 96*a^3*b^7*c^3*f*g*j*z - 96*a^2*b^9*c^2*d*h*k*z - 6

4*a^3*b^7*c^3*e*h*j*z + 9216*a^5*b^2*c^6*e*f*j*z - 9216*a^5*b^2*c^6*d*h*i*z
 - 6656*a^4*b^4*c^5*d*f*k*z - 6144*a^5*b^2*c^6*d*f*k*z + 3456*a^3*b^6*c^4*d
*f*k*z - 2304*a^4*b^4*c^5*e*f*j*z + 2304*a^4*b^4*c^5*d*h*i*z - 576*a^2*b^8*
c^3*d*f*k*z + 192*a^3*b^6*c^4*e*f*j*z - 192*a^3*b^6*c^4*d*h*i*z + 4608*a^4*
b^3*c^6*d*g*h*z + 3072*a^4*b^3*c^6*d*f*i*z - 1152*a^3*b^5*c^5*d*g*h*z - 768
*a^3*b^5*c^5*d*f*i*z + 96*a^2*b^7*c^4*d*g*h*z + 64*a^2*b^7*c^4*d*f*i*z - 92
16*a^4*b^2*c^7*d*e*h*z + 2304*a^3*b^4*c^6*d*e*h*z + 2048*a^4*b^2*c^7*d*f*g*
z - 1536*a^3*b^4*c^6*d*f*g*z + 384*a^2*b^6*c^5*d*f*g*z - 192*a^2*b^6*c^5*d*
e*h*z + 3072*a^3*b^3*c^7*d*e*f*z - 768*a^2*b^5*c^6*d*e*f*z - 3072*a^8*b*c^4
*j^2*k*z + 48*a^5*b^7*c*j^2*k*z - 49152*a^8*b*c^4*i*k^2*z + 2304*a^5*b^7*c*
i*k^2*z - 9216*a^7*b*c^5*h^2*k*z - 32*a^4*b^8*c*i*j^2*z - 1152*a^4*b^8*c*g*
k^2*z + 9216*a^7*b*c^5*g*j^2*z - 3072*a^6*b*c^6*f^2*k*z + 16*a^3*b^9*c*g*j^
2*z - 49152*a^7*b*c^5*e*k^2*z - 128*a^3*b^9*c*e*k^2*z - 58368*a^5*b*c^7*d^2
*k*z - 1024*a^6*b*c^6*g*h^2*z - 432*a*b^9*c^3*d^2*k*z + 1024*a^5*b*c^7*f^2*
g*z + 32*a*b^8*c^4*d^2*i*z - 9216*a^4*b*c^8*d^2*g*z + 336*a*b^7*c^5*d^2*g*z
 - 672*a*b^6*c^6*d^2*e*z + 24576*a^8*c^5*h*j*k*z + 73728*a^7*c^6*d*j*k*z +
32768*a^7*c^6*e*i*k*z - 12288*a^7*c^6*f*i*j*z + 8192*a^7*c^6*f*h*k*z + 2457
6*a^6*c^7*d*f*k*z - 12288*a^6*c^7*e*f*j*z + 12288*a^6*c^7*d*h*i*z + 12288*a
^5*c^8*d*e*h*z + 2304*a^7*b^3*c^3*j^2*k*z - 576*a^6*b^5*c^2*j^2*k*z + 45056
*a^7*b^3*c^3*i*k^2*z - 15360*a^6*b^5*c^2*i*k^2*z - 12288*a^7*b^2*c^4*i^2*k*
z + 3072*a^6*b^4*c^3*i^2*k*z - 256*a^5*b^6*c^2*i^2*k*z + 15872*a^7*b^2*c^4*
i*j^2*z + 6912*a^6*b^3*c^4*h^2*k*z - 4992*a^6*b^4*c^3*i*j^2*z - 1728*a^5*b^
5*c^3*h^2*k*z + 672*a^5*b^6*c^2*i*j^2*z + 144*a^4*b^7*c^2*h^2*k*z + 24576*a
^7*b^2*c^4*g*k^2*z - 22528*a^6*b^4*c^3*g*k^2*z + 7680*a^5*b^6*c^2*g*k^2*z +
 4096*a^6*b^2*c^5*g^2*k*z - 3072*a^5*b^4*c^4*g^2*k*z + 768*a^4*b^6*c^3*g^2*
k*z - 64*a^3*b^8*c^2*g^2*k*z - 7936*a^6*b^3*c^4*g*j^2*z + 2496*a^5*b^5*c^3*
g*j^2*z - 1536*a^6*b^2*c^5*h^2*i*z + 1280*a^5*b^3*c^5*f^2*k*z + 384*a^5*b^4
*c^4*h^2*i*z - 336*a^4*b^7*c^2*g*j^2*z + 192*a^4*b^5*c^4*f^2*k*z - 144*a^3*
b^7*c^3*f^2*k*z - 32*a^4*b^6*c^3*h^2*i*z + 16*a^2*b^9*c^2*f^2*k*z + 45056*a
^6*b^3*c^4*e*k^2*z - 15360*a^5*b^5*c^3*e*k^2*z - 12288*a^5*b^2*c^6*e^2*k*z
+ 3072*a^4*b^4*c^5*e^2*k*z + 2304*a^4*b^7*c^2*e*k^2*z - 256*a^3*b^6*c^4*e^2
*k*z + 59136*a^4*b^3*c^6*d^2*k*z - 23488*a^3*b^5*c^5*d^2*k*z + 15872*a^6*b^
2*c^5*e*j^2*z - 4992*a^5*b^4*c^4*e*j^2*z + 4560*a^2*b^7*c^4*d^2*k*z + 1536*
a^5*b^2*c^6*f^2*i*z + 768*a^5*b^3*c^5*g*h^2*z + 672*a^4*b^6*c^3*e*j^2*z - 3
84*a^4*b^4*c^5*f^2*i*z - 192*a^4*b^5*c^4*g*h^2*z - 32*a^3*b^8*c^2*e*j^2*z +
 32*a^3*b^6*c^4*f^2*i*z + 16*a^3*b^7*c^3*g*h^2*z - 15872*a^4*b^2*c^7*d^2*i*
z + 4992*a^3*b^4*c^6*d^2*i*z - 1536*a^5*b^2*c^6*e*h^2*z - 768*a^4*b^3*c^6*f
^2*g*z - 672*a^2*b^6*c^5*d^2*i*z + 384*a^4*b^4*c^5*e*h^2*z + 192*a^3*b^5*c^
5*f^2*g*z - 32*a^3*b^6*c^4*e*h^2*z - 16*a^2*b^7*c^4*f^2*g*z + 7936*a^3*b^3*
c^7*d^2*g*z - 2496*a^2*b^5*c^6*d^2*g*z + 1536*a^4*b^2*c^7*e*f^2*z - 384*a^3
*b^4*c^6*e*f^2*z + 32*a^2*b^6*c^5*e*f^2*z - 15872*a^3*b^2*c^8*d^2*e*z + 499
2*a^2*b^4*c^7*d^2*e*z - 61440*a^8*b^2*c^3*k^3*z + 21504*a^7*b^4*c^2*k^3*z +
 16384*a^8*c^5*i^2*k*z - 18432*a^8*c^5*i*j^2*z - 128*a^4*b^9*i*k^2*z + 2048
*a^7*c^6*h^2*i*z + 64*a^3*b^10*g*k^2*z + 16384*a^6*c^7*e^2*k*z + 16*b^11*c^
2*d^2*k*z - 18432*a^7*c^6*e*j^2*z - 2048*a^6*c^7*f^2*i*z + 18432*a^5*c^8*d^

2*i*z - 3328*a^6*b^6*c*k^3*z + 2048*a^6*c^7*e*h^2*z - 16*b^9*c^4*d^2*g*z -
2048*a^5*c^8*e*f^2*z + 32*b^8*c^5*d^2*e*z + 18432*a^4*c^9*d^2*e*z + 65536*a
^9*c^4*k^3*z + 192*a^5*b^8*k^3*z - 3328*a^7*b*c^3*h*i*j*k - 6912*a^6*b*c^4*
d*i*j*k - 3328*a^6*b*c^4*e*h*j*k - 1536*a^6*b*c^4*f*g*j*k - 768*a^6*b*c^4*g
*h*i*j - 768*a^6*b*c^4*f*h*i*k - 6912*a^5*b*c^5*d*e*j*k - 2304*a^5*b*c^5*d*
g*i*j - 1792*a^5*b*c^5*e*f*i*j + 1536*a^5*b*c^5*d*g*h*k - 1280*a^5*b*c^5*d*
f*i*k - 768*a^5*b*c^5*e*g*h*j - 768*a^5*b*c^5*e*f*h*k - 256*a^5*b*c^5*f*g*h
*i + 16*a*b^8*c^2*d*f*g*k - 4*a*b^8*c^2*d*f*h*j - 2304*a^4*b*c^6*d*e*g*j -
1792*a^4*b*c^6*d*e*h*i - 1280*a^4*b*c^6*d*e*f*k - 768*a^4*b*c^6*d*f*g*i - 2
56*a^4*b*c^6*e*f*g*h - 32*a*b^7*c^3*d*e*f*k - 768*a^3*b*c^7*d*e*f*g + 32*a*
b^5*c^5*d*e*f*g + 576*a^6*b^3*c^2*h*i*j*k + 1664*a^6*b^2*c^3*g*h*j*k + 384*
a^6*b^2*c^3*f*i*j*k - 288*a^5*b^4*c^2*g*h*j*k - 160*a^5*b^4*c^2*f*i*j*k + 2
112*a^5*b^3*c^3*d*i*j*k + 576*a^5*b^3*c^3*e*h*j*k - 448*a^5*b^3*c^3*f*h*i*k
 - 192*a^5*b^3*c^3*g*h*i*j - 192*a^5*b^3*c^3*f*g*j*k - 160*a^4*b^5*c^2*d*i*
j*k + 96*a^4*b^5*c^2*f*h*i*k + 80*a^4*b^5*c^2*f*g*j*k + 32*a^4*b^5*c^2*g*h*
i*j + 4992*a^5*b^2*c^4*d*h*i*k - 4608*a^5*b^2*c^4*e*g*i*k + 3456*a^5*b^2*c^
4*d*g*j*k - 1312*a^4*b^4*c^3*d*h*i*k - 1056*a^4*b^4*c^3*d*g*j*k + 896*a^5*b
^2*c^4*f*g*i*j + 768*a^4*b^4*c^3*e*g*i*k + 384*a^5*b^2*c^4*f*g*h*k + 384*a^
5*b^2*c^4*e*h*i*j + 384*a^5*b^2*c^4*e*f*j*k + 224*a^4*b^4*c^3*f*g*h*k - 160
*a^4*b^4*c^3*e*f*j*k - 96*a^4*b^4*c^3*f*g*i*j + 96*a^3*b^6*c^2*d*h*i*k + 80
*a^3*b^6*c^2*d*g*j*k - 64*a^4*b^4*c^3*e*h*i*j - 48*a^3*b^6*c^2*f*g*h*k - 24
96*a^4*b^3*c^4*d*g*h*k + 2112*a^4*b^3*c^4*d*e*j*k - 960*a^4*b^3*c^4*d*f*i*k
 + 656*a^3*b^5*c^3*d*g*h*k - 448*a^4*b^3*c^4*e*f*h*k + 384*a^3*b^5*c^3*d*f*
i*k + 320*a^4*b^3*c^4*d*g*i*j - 192*a^4*b^3*c^4*f*g*h*i - 192*a^4*b^3*c^4*e
*g*h*j + 192*a^4*b^3*c^4*e*f*i*j - 160*a^3*b^5*c^3*d*e*j*k + 96*a^3*b^5*c^3
*e*f*h*k - 48*a^2*b^7*c^2*d*g*h*k + 32*a^3*b^5*c^3*e*g*h*j - 32*a^2*b^7*c^2
*d*f*i*k + 4992*a^4*b^2*c^5*d*e*h*k - 3584*a^4*b^2*c^5*d*f*h*j - 1312*a^3*b
^4*c^4*d*e*h*k + 896*a^4*b^2*c^5*e*f*g*j + 896*a^4*b^2*c^5*d*g*h*i + 640*a^
4*b^2*c^5*d*f*g*k - 640*a^4*b^2*c^5*d*e*i*j + 600*a^3*b^4*c^4*d*f*h*j + 480
*a^3*b^4*c^4*d*f*g*k + 384*a^4*b^2*c^5*e*f*h*i - 192*a^2*b^6*c^3*d*f*g*k -
96*a^3*b^4*c^4*e*f*g*j - 96*a^3*b^4*c^4*d*g*h*i + 96*a^2*b^6*c^3*d*e*h*k +
12*a^2*b^6*c^3*d*f*h*j - 960*a^3*b^3*c^5*d*e*f*k + 384*a^2*b^5*c^4*d*e*f*k
+ 320*a^3*b^3*c^5*d*e*g*j - 192*a^3*b^3*c^5*e*f*g*h - 192*a^3*b^3*c^5*d*f*g
*i + 192*a^3*b^3*c^5*d*e*h*i + 32*a^2*b^5*c^4*d*f*g*i + 896*a^3*b^2*c^6*d*e
*g*h + 384*a^3*b^2*c^6*d*e*f*i - 96*a^2*b^4*c^5*d*e*g*h - 64*a^2*b^4*c^5*d*
e*f*i - 192*a^2*b^3*c^6*d*e*f*g + 48*a^6*b^4*c*i*j^2*k - 1424*a^6*b^4*c*h*j
*k^2 - 2304*a^7*b*c^3*g*j^2*k - 24*a^5*b^5*c*g*j^2*k + 2048*a^7*b*c^3*g*i*k
^2 - 1024*a^7*b*c^3*f*j*k^2 - 768*a^5*b^5*c*g*i*k^2 + 408*a^5*b^5*c*f*j*k^2
 + 256*a^6*b*c^4*g*h^2*k + 16*a^4*b^6*c*g*i*j^2 + 4608*a^6*b*c^4*e*i^2*k +
4608*a^5*b*c^5*e^2*i*k - 896*a^6*b*c^4*f*i^2*j + 768*a^4*b^6*c*d*j*k^2 - 25
6*a^4*b^6*c*f*h*k^2 - 128*a^4*b^6*c*e*i*k^2 + 2208*a^6*b*c^4*f*h*j^2 - 1920
*a^6*b*c^4*e*i*j^2 + 800*a^5*b*c^5*f^2*h*j - 256*a^5*b*c^5*f^2*g*k - 16*a*b
^8*c^2*d^2*i*k + 6*a^3*b^7*c*f*h*j^2 + 8192*a^6*b*c^4*d*h*k^2 + 2048*a^6*b*
c^4*e*g*k^2 - 472*a^3*b^7*c*d*h*k^2 + 64*a^3*b^7*c*e*g*k^2 + 4896*a^4*b*c^6
*d^2*h*j + 2304*a^4*b*c^6*d^2*g*k + 1824*a^5*b*c^5*d*h^2*j - 384*a^5*b*c^5*

e*h^2*i - 168*a*b^7*c^3*d^2*g*k + 42*a*b^7*c^3*d^2*h*j + 6*a^2*b^8*c*d*h*j^
2 + 1536*a^5*b*c^5*e*g*i^2 + 1536*a^4*b*c^6*e^2*g*i - 896*a^5*b*c^5*d*h*i^2
 - 896*a^4*b*c^6*e^2*f*j + 144*a^2*b^8*c*d*f*k^2 + 4896*a^5*b*c^5*d*f*j^2 +
 1824*a^4*b*c^6*d*f^2*j - 384*a^4*b*c^6*e*f^2*i + 336*a*b^6*c^4*d^2*e*k - 1
56*a*b^6*c^4*d^2*f*j + 16*a*b^6*c^4*d^2*g*i + 12*a*b^7*c^3*d*f^2*j + 2208*a
^3*b*c^7*d^2*f*h - 1920*a^3*b*c^7*d^2*e*i + 800*a^4*b*c^6*d*f*h^2 - 102*a*b
^5*c^5*d^2*f*h - 32*a*b^5*c^5*d^2*e*i + 12*a*b^6*c^4*d*f^2*h - 2*a*b^7*c^3*
d*f*h^2 - 896*a^3*b*c^7*d*e^2*h - 8*a*b^6*c^4*d*f*g^2 - 240*a*b^4*c^6*d^2*e
*g - 32*a*b^4*c^6*d*e^2*f + 3072*a^7*c^4*f*i*j*k + 3072*a^6*c^5*e*f*j*k - 3
072*a^6*c^5*d*h*i*k + 1536*a^6*c^5*e*h*i*j + 4608*a^5*c^6*d*e*i*j - 3072*a^
5*c^6*d*e*h*k - 1152*a^5*c^6*d*f*h*j + 512*a^5*c^6*e*f*h*i + 1536*a^4*c^7*d
*e*f*i - 2*a*b^9*c*d*f*j^2 - 1088*a^7*b^2*c^2*i*j^2*k + 4800*a^7*b^2*c^2*h*
j*k^2 + 960*a^6*b^2*c^3*h^2*i*k + 544*a^6*b^3*c^2*g*j^2*k - 144*a^5*b^4*c^2
*h^2*i*k - 2304*a^6*b^2*c^3*g*i^2*k + 1920*a^6*b^3*c^2*g*i*k^2 + 1152*a^5*b
^3*c^3*g^2*i*k - 864*a^6*b^3*c^2*f*j*k^2 + 384*a^5*b^4*c^2*g*i^2*k + 192*a^
6*b^2*c^3*h*i^2*j - 192*a^4*b^5*c^2*g^2*i*k - 32*a^5*b^4*c^2*h*i^2*j - 1088
*a^6*b^2*c^3*e*j^2*k + 960*a^6*b^2*c^3*g*i*j^2 - 480*a^5*b^3*c^3*g*h^2*k -
240*a^5*b^4*c^2*g*i*j^2 + 192*a^5*b^2*c^4*f^2*i*k + 72*a^4*b^5*c^2*g*h^2*k
+ 48*a^5*b^4*c^2*e*j^2*k + 48*a^4*b^4*c^3*f^2*i*k - 16*a^3*b^6*c^2*f^2*i*k
+ 13376*a^6*b^2*c^3*d*j*k^2 - 5136*a^5*b^4*c^2*d*j*k^2 - 3840*a^6*b^2*c^3*e
*i*k^2 + 1536*a^5*b^4*c^2*e*i*k^2 - 768*a^5*b^3*c^3*e*i^2*k - 768*a^4*b^3*c
^4*e^2*i*k + 624*a^5*b^4*c^2*f*h*k^2 + 576*a^6*b^2*c^3*f*h*k^2 + 192*a^5*b^
2*c^4*g^2*h*j + 96*a^5*b^3*c^3*f*i^2*j + 48*a^4*b^4*c^3*g^2*h*j - 8*a^3*b^6
*c^2*g^2*h*j + 6848*a^4*b^2*c^5*d^2*i*k - 2448*a^3*b^4*c^4*d^2*i*k + 960*a^
5*b^2*c^4*e*h^2*k - 864*a^5*b^2*c^4*f*h^2*j + 480*a^5*b^3*c^3*e*i*j^2 + 336
*a^4*b^3*c^4*f^2*h*j + 336*a^2*b^6*c^3*d^2*i*k + 192*a^5*b^2*c^4*g*h^2*i +
144*a^5*b^3*c^3*f*h*j^2 - 144*a^4*b^4*c^3*e*h^2*k - 102*a^4*b^5*c^2*f*h*j^2
 - 96*a^4*b^3*c^4*f^2*g*k - 32*a^4*b^5*c^2*e*i*j^2 - 30*a^3*b^5*c^3*f^2*h*j
 - 24*a^3*b^5*c^3*f^2*g*k + 16*a^4*b^4*c^3*g*h^2*i - 12*a^4*b^4*c^3*f*h^2*j
 + 12*a^3*b^6*c^2*f*h^2*j + 8*a^2*b^7*c^2*f^2*g*k - 2*a^2*b^7*c^2*f^2*h*j -
 9312*a^5*b^3*c^3*d*h*k^2 + 3288*a^4*b^5*c^2*d*h*k^2 - 2304*a^4*b^2*c^5*e^2
*g*k + 1920*a^5*b^3*c^3*e*g*k^2 + 1152*a^4*b^3*c^4*e*g^2*k - 768*a^4*b^5*c^
2*e*g*k^2 + 384*a^3*b^4*c^4*e^2*g*k - 320*a^5*b^2*c^4*d*i^2*j - 224*a^4*b^3
*c^4*f*g^2*j + 192*a^5*b^2*c^4*f*h*i^2 + 192*a^4*b^2*c^5*e^2*h*j - 192*a^3*
b^5*c^3*e*g^2*k - 32*a^3*b^4*c^4*e^2*h*j + 24*a^3*b^5*c^3*f*g^2*j - 3552*a^
5*b^2*c^4*d*h*j^2 - 3424*a^3*b^3*c^5*d^2*g*k + 1332*a^4*b^4*c^3*d*h*j^2 + 1
224*a^2*b^5*c^4*d^2*g*k + 960*a^5*b^2*c^4*e*g*j^2 - 496*a^3*b^3*c^5*d^2*h*j
 + 432*a^4*b^3*c^4*d*h^2*j - 240*a^4*b^4*c^3*e*g*j^2 - 222*a^2*b^5*c^4*d^2*
h*j + 192*a^4*b^2*c^5*f^2*g*i + 192*a^4*b^2*c^5*e*f^2*k - 174*a^3*b^5*c^3*d
*h^2*j - 156*a^3*b^6*c^2*d*h*j^2 + 48*a^3*b^4*c^4*e*f^2*k - 32*a^4*b^3*c^4*
e*h^2*i + 16*a^3*b^6*c^2*e*g*j^2 + 16*a^3*b^4*c^4*f^2*g*i - 16*a^2*b^6*c^3*
e*f^2*k + 12*a^2*b^7*c^2*d*h^2*j + 1728*a^5*b^2*c^4*d*f*k^2 + 1392*a^4*b^4*
c^3*d*f*k^2 - 840*a^3*b^6*c^2*d*f*k^2 - 768*a^4*b^2*c^5*e*g^2*i + 576*a^4*b
^2*c^5*d*g^2*j + 96*a^4*b^3*c^4*d*h*i^2 + 96*a^3*b^3*c^5*e^2*f*j - 80*a^3*b
^4*c^4*d*g^2*j + 64*a^4*b^2*c^5*f*g^2*h + 48*a^3*b^4*c^4*f*g^2*h + 6848*a^3

*b^2*c^6*d^2*e*k - 3552*a^3*b^2*c^6*d^2*f*j - 2448*a^2*b^4*c^5*d^2*e*k + 13
32*a^2*b^4*c^5*d^2*f*j + 960*a^3*b^2*c^6*d^2*g*i - 496*a^4*b^3*c^4*d*f*j^2
+ 432*a^3*b^3*c^5*d*f^2*j - 240*a^2*b^4*c^5*d^2*g*i - 222*a^3*b^5*c^3*d*f*j
^2 + 192*a^4*b^2*c^5*e*g*h^2 - 174*a^2*b^5*c^4*d*f^2*j + 42*a^2*b^7*c^2*d*f
*j^2 - 32*a^3*b^3*c^5*e*f^2*i + 16*a^3*b^4*c^4*e*g*h^2 - 320*a^3*b^2*c^6*d*
e^2*j - 224*a^3*b^3*c^5*d*g^2*h + 192*a^4*b^2*c^5*d*f*i^2 + 192*a^3*b^2*c^6
*e^2*f*h - 32*a^3*b^4*c^4*d*f*i^2 + 24*a^2*b^5*c^4*d*g^2*h - 864*a^3*b^2*c^
6*d*f^2*h + 480*a^2*b^3*c^6*d^2*e*i + 336*a^3*b^3*c^5*d*f*h^2 + 192*a^3*b^2
*c^6*e*f^2*g + 144*a^2*b^3*c^6*d^2*f*h - 30*a^2*b^5*c^4*d*f*h^2 + 16*a^2*b^
4*c^5*e*f^2*g - 12*a^2*b^4*c^5*d*f^2*h + 192*a^3*b^2*c^6*d*f*g^2 + 96*a^2*b
^3*c^6*d*e^2*h + 48*a^2*b^4*c^5*d*f*g^2 + 960*a^2*b^2*c^7*d^2*e*g + 192*a^2
*b^2*c^7*d*e^2*f - 3072*a^8*b*c^2*j^2*k^2 + 1104*a^7*b^3*c*j^2*k^2 + 768*a^
6*b^4*c*i^2*k^2 - 256*a^6*b^3*c^2*i^3*k + 1536*a^7*b*c^3*h^2*k^2 - 960*a^7*
b*c^3*i^2*j^2 + 444*a^5*b^5*c*h^2*k^2 - 16*a^5*b^5*c*i^2*j^2 - 3072*a^7*b^2
*c^2*g*k^3 - 496*a^6*b^3*c^2*h*j^3 + 192*a^4*b^6*c*g^2*k^2 - 192*a^4*b^4*c^
3*g^3*k + 144*a^5*b^3*c^3*h^3*j + 32*a^3*b^6*c^2*g^3*k - 18*a^4*b^5*c^2*h^3
*j - 9*a^4*b^6*c*h^2*j^2 - 192*a^6*b*c^4*h^2*i^2 + 36*a^3*b^7*c*f^2*k^2 - 4
*a^3*b^7*c*g^2*j^2 - 2176*a^6*b^3*c^2*e*k^3 - 256*a^3*b^3*c^5*e^3*k - 192*a
^6*b^2*c^3*f*j^3 - 192*a^4*b^2*c^5*f^3*j + 132*a^5*b^4*c^2*f*j^3 + 128*a^4*
b^3*c^4*g^3*i - 28*a^3*b^4*c^4*f^3*j + 6*a^2*b^6*c^3*f^3*j + 10752*a^5*b*c^
5*d^2*k^2 - 960*a^5*b*c^5*e^2*j^2 - 192*a^5*b*c^5*f^2*i^2 - 1680*a^5*b^3*c^
3*d*j^3 - 1680*a^2*b^3*c^6*d^3*j + 222*a^4*b^5*c^2*d*j^3 + 80*a^4*b^3*c^4*f
*h^3 + 80*a^3*b^3*c^5*f^3*h + 30*a*b^8*c^2*d^2*j^2 + 6*a^3*b^5*c^3*f*h^3 +
6*a^2*b^5*c^4*f^3*h - 960*a^4*b*c^6*d^2*i^2 - 192*a^4*b*c^6*e^2*h^2 - 192*a
^4*b^2*c^5*d*h^3 - 192*a^2*b^2*c^7*d^3*h + 128*a^3*b^3*c^5*e*g^3 - 28*a^3*b
^4*c^4*d*h^3 + 12*a*b^6*c^4*d^2*h^2 + 6*a^2*b^6*c^3*d*h^3 - 192*a^3*b*c^7*e
^2*f^2 + 60*a*b^5*c^5*d^2*g^2 + 198*a*b^4*c^6*d^2*f^2 + 144*a^2*b^3*c^6*d*f
^3 - 960*a^2*b*c^8*d^2*e^2 + 240*a*b^3*c^7*d^2*e^2 + 4608*a^8*c^3*i*j^2*k -
 3072*a^8*c^3*h*j*k^2 - 512*a^7*c^4*h^2*i*k + 120*a^5*b^6*h*j*k^2 + 768*a^7
*c^4*h*i^2*j + 4608*a^7*c^4*e*j^2*k + 512*a^6*c^5*f^2*i*k + 64*a^4*b^7*g*i*
k^2 - 40*a^4*b^7*f*j*k^2 - 9216*a^7*c^4*d*j*k^2 - 4096*a^7*c^4*e*i*k^2 - 10
24*a^7*c^4*f*h*k^2 - 4608*a^5*c^6*d^2*i*k - 512*a^6*c^5*e*h^2*k - 192*a^6*c
^5*f*h^2*j - 40*a^3*b^8*d*j*k^2 + 24*a^3*b^8*f*h*k^2 + 2304*a^6*c^5*d*i^2*j
 + 768*a^5*c^6*e^2*h*j + 256*a^6*c^5*f*h*i^2 + 8*b^9*c^2*d^2*g*k - 2*b^9*c^
2*d^2*h*j + 6144*a^8*b*c^2*i*k^3 - 2176*a^7*b^3*c*i*k^3 - 1728*a^6*c^5*d*h*
j^2 + 1536*a^7*b*c^3*i^3*k + 512*a^5*c^6*e*f^2*k + 24*a^2*b^9*d*h*k^2 - 307
2*a^6*c^5*d*f*k^2 - 16*b^8*c^3*d^2*e*k + 6*b^8*c^3*d^2*f*j - 4608*a^4*c^7*d
^2*e*k + 2016*a^7*b*c^3*h*j^3 - 1728*a^4*c^7*d^2*f*j + 1088*a^6*b^4*c*g*k^3
 + 224*a^6*b*c^4*h^3*j + 30*a^5*b^5*c*h*j^3 + 2304*a^4*c^7*d*e^2*j + 768*a^
5*c^6*d*f*i^2 + 256*a^4*c^7*e^2*f*h + 6*b^7*c^4*d^2*f*h + 6144*a^7*b*c^3*e*
k^3 + 1536*a^4*b*c^6*e^3*k + 512*a^6*b*c^4*g*i^3 + 192*a^5*b^5*c*e*k^3 - 19
2*a^4*c^7*d*f^2*h - 10*a^4*b^6*c*f*j^3 + 108*a*b^9*c*d^2*k^2 + 16*b^6*c^5*d
^2*e*g + 4320*a^6*b*c^4*d*j^3 + 4320*a^3*b*c^7*d^3*j + 222*a*b^5*c^5*d^3*j
+ 96*a^5*b*c^5*f*h^3 + 96*a^4*b*c^6*f^3*h - 10*a^3*b^7*c*d*j^3 + 768*a^3*c^
8*d*e^2*f + 512*a^3*b*c^7*e^3*g + 132*a*b^4*c^6*d^3*h + 2016*a^2*b*c^8*d^3*

f - 496*a*b^3*c^7*d^3*f + 224*a^3*b*c^7*d*f^3 - 18*a*b^5*c^5*d*f^3 - 1920*a
^7*b^2*c^2*i^2*k^2 - 1648*a^6*b^3*c^2*h^2*k^2 + 240*a^6*b^3*c^2*i^2*j^2 - 9
60*a^6*b^2*c^3*h^2*j^2 - 512*a^6*b^2*c^3*g^2*k^2 - 480*a^5*b^4*c^2*g^2*k^2
+ 198*a^5*b^4*c^2*h^2*j^2 - 240*a^5*b^3*c^3*g^2*j^2 - 240*a^5*b^3*c^3*f^2*k
^2 + 60*a^4*b^5*c^2*g^2*j^2 - 36*a^4*b^5*c^2*f^2*k^2 - 16*a^5*b^3*c^3*h^2*i
^2 - 1920*a^5*b^2*c^4*e^2*k^2 + 768*a^4*b^4*c^3*e^2*k^2 - 464*a^5*b^2*c^4*f
^2*j^2 - 384*a^5*b^2*c^4*g^2*i^2 - 64*a^3*b^6*c^2*e^2*k^2 + 42*a^4*b^4*c^3*
f^2*j^2 + 12*a^3*b^6*c^2*f^2*j^2 - 13104*a^4*b^3*c^4*d^2*k^2 + 5628*a^3*b^5
*c^3*d^2*k^2 - 1128*a^2*b^7*c^2*d^2*k^2 + 240*a^4*b^3*c^4*e^2*j^2 - 48*a^4*
b^3*c^4*g^2*h^2 - 16*a^4*b^3*c^4*f^2*i^2 - 16*a^3*b^5*c^3*e^2*j^2 - 4*a^3*b
^5*c^3*g^2*h^2 - 2880*a^4*b^2*c^5*d^2*j^2 + 1750*a^3*b^4*c^4*d^2*j^2 - 345*
a^2*b^6*c^3*d^2*j^2 - 192*a^4*b^2*c^5*f^2*h^2 - 42*a^3*b^4*c^4*f^2*h^2 + 24
0*a^3*b^3*c^5*d^2*i^2 - 48*a^3*b^3*c^5*f^2*g^2 - 16*a^3*b^3*c^5*e^2*h^2 - 1
6*a^2*b^5*c^4*d^2*i^2 - 4*a^2*b^5*c^4*f^2*g^2 - 464*a^3*b^2*c^6*d^2*h^2 - 3
84*a^3*b^2*c^6*e^2*g^2 + 42*a^2*b^4*c^5*d^2*h^2 - 240*a^2*b^3*c^6*d^2*g^2 -
16*a^2*b^3*c^6*e^2*f^2 - 960*a^2*b^2*c^7*d^2*f^2 - 8*a*b^10*d*f*k^2 - a^2*
b^8*c*f^2*j^2 - 2048*a^8*c^3*i^2*k^2 - 100*a^6*b^5*j^2*k^2 - 64*a^5*b^6*i^2
*k^2 - 288*a^7*c^4*h^2*j^2 - 36*a^4*b^7*h^2*k^2 - 16*a^3*b^8*g^2*k^2 - 2048
*a^6*c^5*e^2*k^2 - 864*a^6*c^5*f^2*j^2 - 4*a^2*b^9*f^2*k^2 - 2592*a^5*c^6*d
^2*j^2 - 1536*a^5*c^6*e^2*i^2 - 32*a^5*c^6*f^2*h^2 - 864*a^4*c^7*d^2*h^2 +
360*a^7*b^2*c^2*j^4 - 4*b^7*c^4*d^2*g^2 - 9*b^6*c^5*d^2*f^2 - 288*a^3*c^8*d
^2*f^2 - 24*a^5*b^2*c^4*h^4 - 16*b^5*c^6*d^2*e^2 - 9*a^4*b^4*c^3*h^4 - 16*a
^3*b^4*c^4*g^4 - 24*a^3*b^2*c^6*f^4 - 9*a^2*b^4*c^5*f^4 - a^2*b^6*c^3*f^2*h
^2 + 192*a^6*b^5*i*k^3 - 96*a^5*b^6*g*k^3 - 1728*a^7*c^4*f*j^3 - 192*a^5*c^
6*f^3*j - 10*b^7*c^4*d^3*j - 1024*a^6*c^5*e*i^3 - 1024*a^4*c^7*e^3*i + 1536
*a^8*b^2*c*k^4 - 10*b^6*c^5*d^3*h - 1728*a^3*c^8*d^3*h - 192*a^5*c^6*d*h^3
- 25*a^6*b^4*c*j^4 + 30*b^5*c^6*d^3*f + 360*a*b^2*c^8*d^4 - 4*b^11*d^2*k^2
- 4096*a^9*c^2*k^4 - 1296*a^8*c^3*j^4 - 144*a^7*b^4*k^4 - 256*a^7*c^4*i^4 -
16*a^6*c^5*h^4 - 16*a^4*c^7*f^4 - 256*a^3*c^8*e^4 - 25*b^4*c^7*d^4 - 1296*
a^2*c^9*d^4 - b^8*c^3*d^2*h^2 - b^10*c*d^2*j^2, z, n)*((3072*a^5*c^6*d*k -
512*a^4*c^7*e*f - 1536*a^5*c^6*e*j - 512*a^5*c^6*f*i + 1024*a^6*c^5*h*k - 1
536*a^6*c^5*i*j + 32*a*b^5*c^5*d*e + 1024*a^3*b*c^7*d*e - 16*a*b^6*c^4*d*g
+ 1024*a^4*b*c^6*d*i + 512*a^4*b*c^6*e*h + 256*a^4*b*c^6*f*g + 16*a*b^8*c^2
*d*k + 256*a^5*b*c^5*f*k + 768*a^5*b*c^5*g*j + 512*a^5*b*c^5*h*i + 1792*a^6
*b*c^4*j*k - 384*a^2*b^3*c^6*d*e + 192*a^2*b^4*c^5*d*g + 32*a^2*b^4*c^5*e*f
- 512*a^3*b^2*c^6*d*g + 32*a^2*b^5*c^4*d*i - 16*a^2*b^5*c^4*f*g - 384*a^3*
b^3*c^5*d*i - 128*a^3*b^3*c^5*e*h - 288*a^2*b^6*c^3*d*k + 1792*a^3*b^4*c^4*
d*k - 32*a^3*b^4*c^4*e*j + 32*a^3*b^4*c^4*f*i + 64*a^3*b^4*c^4*g*h - 4352*a
^4*b^2*c^5*d*k + 512*a^4*b^2*c^5*e*j - 256*a^4*b^2*c^5*g*h + 16*a^2*b^7*c^2
*f*k - 144*a^3*b^5*c^3*f*k + 16*a^3*b^5*c^3*g*j + 256*a^4*b^3*c^4*f*k - 256
*a^4*b^3*c^4*g*j - 128*a^4*b^3*c^4*h*i - 48*a^3*b^6*c^2*h*k + 512*a^4*b^4*c
^3*h*k - 32*a^4*b^4*c^3*i*j - 1536*a^5*b^2*c^4*h*k + 512*a^5*b^2*c^4*i*j +
80*a^4*b^5*c^2*j*k - 768*a^5*b^3*c^3*j*k)/(8*(64*a^5*c^5 - a^2*b^6*c^2 + 12
*a^3*b^4*c^3 - 48*a^4*b^2*c^4)) - root(1572864*a^8*b^2*c^9*z^4 - 983040*a^7
*b^4*c^8*z^4 + 327680*a^6*b^6*c^7*z^4 - 61440*a^5*b^8*c^6*z^4 + 6144*a^4*b^

10*c^5*z^4 - 256*a^3*b^12*c^4*z^4 - 1048576*a^9*c^10*z^4 - 1572864*a^8*b^2*
c^7*k*z^3 + 983040*a^7*b^4*c^6*k*z^3 - 327680*a^6*b^6*c^5*k*z^3 + 61440*a^5
*b^8*c^4*k*z^3 - 6144*a^4*b^10*c^3*k*z^3 + 256*a^3*b^12*c^2*k*z^3 + 1048576
*a^9*c^8*k*z^3 + 98304*a^8*b*c^6*i*k*z^2 + 98304*a^7*b*c^7*e*k*z^2 + 57344*
a^7*b*c^7*f*j*z^2 + 32768*a^7*b*c^7*g*i*z^2 + 57344*a^6*b*c^8*d*h*z^2 + 327
68*a^6*b*c^8*e*g*z^2 - 32*a*b^10*c^4*d*f*z^2 - 90112*a^7*b^3*c^5*i*k*z^2 +
30720*a^6*b^5*c^4*i*k*z^2 - 4608*a^5*b^7*c^3*i*k*z^2 + 256*a^4*b^9*c^2*i*k*
z^2 - 49152*a^7*b^2*c^6*g*k*z^2 + 45056*a^6*b^4*c^5*g*k*z^2 + 24576*a^7*b^2
*c^6*h*j*z^2 - 15360*a^5*b^6*c^4*g*k*z^2 - 3072*a^5*b^6*c^4*h*j*z^2 + 2304*
a^4*b^8*c^3*g*k*z^2 + 2048*a^6*b^4*c^5*h*j*z^2 + 576*a^4*b^8*c^3*h*j*z^2 -
128*a^3*b^10*c^2*g*k*z^2 - 32*a^3*b^10*c^2*h*j*z^2 - 90112*a^6*b^3*c^6*e*k*
z^2 - 49152*a^6*b^3*c^6*f*j*z^2 + 30720*a^5*b^5*c^5*e*k*z^2 - 24576*a^6*b^3
*c^6*g*i*z^2 + 15360*a^5*b^5*c^5*f*j*z^2 + 6144*a^5*b^5*c^5*g*i*z^2 - 4608*
a^4*b^7*c^4*e*k*z^2 - 2048*a^4*b^7*c^4*f*j*z^2 - 512*a^4*b^7*c^4*g*i*z^2 +
256*a^3*b^9*c^3*e*k*z^2 + 96*a^3*b^9*c^3*f*j*z^2 + 131072*a^6*b^2*c^7*d*j*z
^2 + 49152*a^6*b^2*c^7*e*i*z^2 - 43008*a^5*b^4*c^6*d*j*z^2 - 12288*a^5*b^4*
c^6*e*i*z^2 + 6144*a^5*b^4*c^6*f*h*z^2 + 6144*a^4*b^6*c^5*d*j*z^2 - 2048*a^
4*b^6*c^5*f*h*z^2 + 1024*a^4*b^6*c^5*e*i*z^2 - 320*a^3*b^8*c^4*d*j*z^2 + 19
2*a^3*b^8*c^4*f*h*z^2 - 49152*a^5*b^3*c^7*d*h*z^2 - 24576*a^5*b^3*c^7*e*g*z
^2 + 15360*a^4*b^5*c^6*d*h*z^2 + 6144*a^4*b^5*c^6*e*g*z^2 - 2048*a^3*b^7*c^
5*d*h*z^2 - 512*a^3*b^7*c^5*e*g*z^2 + 96*a^2*b^9*c^4*d*h*z^2 + 24576*a^5*b^
2*c^8*d*f*z^2 - 3072*a^3*b^6*c^6*d*f*z^2 + 2048*a^4*b^4*c^7*d*f*z^2 + 576*a
^2*b^8*c^5*d*f*z^2 + 1536*a^4*b^10*c*k^2*z^2 + 61440*a^8*b*c^6*j^2*z^2 - 16
*a^3*b^11*c*j^2*z^2 + 12288*a^7*b*c^7*h^2*z^2 + 12288*a^6*b*c^8*f^2*z^2 + 6
1440*a^5*b*c^9*d^2*z^2 + 432*a*b^9*c^5*d^2*z^2 - 49152*a^8*c^7*h*j*z^2 - 14
7456*a^7*c^8*d*j*z^2 - 65536*a^7*c^8*e*i*z^2 - 16384*a^7*c^8*f*h*z^2 - 4915
2*a^6*c^9*d*f*z^2 + 516096*a^8*b^2*c^5*k^2*z^2 - 288768*a^7*b^4*c^4*k^2*z^2
 + 88576*a^6*b^6*c^3*k^2*z^2 - 15744*a^5*b^8*c^2*k^2*z^2 - 61440*a^7*b^3*c^
5*j^2*z^2 + 24064*a^6*b^5*c^4*j^2*z^2 - 4608*a^5*b^7*c^3*j^2*z^2 + 432*a^4*
b^9*c^2*j^2*z^2 + 24576*a^7*b^2*c^6*i^2*z^2 - 6144*a^6*b^4*c^5*i^2*z^2 + 51
2*a^5*b^6*c^4*i^2*z^2 - 8192*a^6*b^3*c^6*h^2*z^2 + 1536*a^5*b^5*c^5*h^2*z^2
 - 16*a^3*b^9*c^3*h^2*z^2 - 8192*a^6*b^2*c^7*g^2*z^2 + 6144*a^5*b^4*c^6*g^2
*z^2 - 1536*a^4*b^6*c^5*g^2*z^2 + 128*a^3*b^8*c^4*g^2*z^2 - 8192*a^5*b^3*c^
7*f^2*z^2 + 1536*a^4*b^5*c^6*f^2*z^2 - 16*a^2*b^9*c^4*f^2*z^2 + 24576*a^5*b
^2*c^8*e^2*z^2 - 6144*a^4*b^4*c^7*e^2*z^2 + 512*a^3*b^6*c^6*e^2*z^2 - 61440
*a^4*b^3*c^8*d^2*z^2 + 24064*a^3*b^5*c^7*d^2*z^2 - 4608*a^2*b^7*c^6*d^2*z^2
 - 393216*a^9*c^6*k^2*z^2 - 64*a^3*b^12*k^2*z^2 - 32768*a^8*c^7*i^2*z^2 - 3
2768*a^6*c^9*e^2*z^2 - 16*b^11*c^4*d^2*z^2 - 16384*a^7*b*c^5*g*i*k*z - 1024
0*a^7*b*c^5*f*j*k*z + 4096*a^7*b*c^5*h*i*j*z - 47104*a^6*b*c^6*d*h*k*z - 16
384*a^6*b*c^6*e*g*k*z + 6144*a^6*b*c^6*f*g*j*z + 4096*a^6*b*c^6*e*h*j*z + 3
2*a*b^10*c^2*d*f*k*z - 6144*a^5*b*c^7*d*g*h*z - 4096*a^5*b*c^7*d*f*i*z - 32
*a*b^8*c^4*d*f*g*z - 4096*a^4*b*c^8*d*e*f*z + 64*a*b^7*c^5*d*e*f*z - 18432*
a^7*b^2*c^4*h*j*k*z + 4608*a^6*b^4*c^3*h*j*k*z - 384*a^5*b^6*c^2*h*j*k*z +
12288*a^6*b^3*c^4*g*i*k*z + 7680*a^6*b^3*c^4*f*j*k*z - 3072*a^6*b^3*c^4*h*i
*j*z - 3072*a^5*b^5*c^3*g*i*k*z - 1920*a^5*b^5*c^3*f*j*k*z + 768*a^5*b^5*c^

3*h*i*j*z + 256*a^4*b^7*c^2*g*i*k*z + 160*a^4*b^7*c^2*f*j*k*z - 64*a^4*b^7*
c^2*h*i*j*z - 65536*a^6*b^2*c^5*d*j*k*z - 24576*a^6*b^2*c^5*e*i*k*z + 21504
*a^5*b^4*c^4*d*j*k*z + 9216*a^6*b^2*c^5*f*i*j*z + 6144*a^5*b^4*c^4*e*i*k*z
- 3072*a^5*b^4*c^4*f*h*k*z - 3072*a^4*b^6*c^3*d*j*k*z - 2304*a^5*b^4*c^4*f*
i*j*z - 2048*a^6*b^2*c^5*g*h*j*z + 1536*a^5*b^4*c^4*g*h*j*z + 1024*a^4*b^6*
c^3*f*h*k*z - 512*a^4*b^6*c^3*e*i*k*z - 384*a^4*b^6*c^3*g*h*j*z + 192*a^4*b
^6*c^3*f*i*j*z + 160*a^3*b^8*c^2*d*j*k*z - 96*a^3*b^8*c^2*f*h*k*z + 32*a^3*
b^8*c^2*g*h*j*z + 41472*a^5*b^3*c^5*d*h*k*z - 13440*a^4*b^5*c^4*d*h*k*z + 1
2288*a^5*b^3*c^5*e*g*k*z - 4608*a^5*b^3*c^5*f*g*j*z - 3072*a^5*b^3*c^5*e*h*
j*z - 3072*a^4*b^5*c^4*e*g*k*z + 1888*a^3*b^7*c^3*d*h*k*z + 1152*a^4*b^5*c^
4*f*g*j*z + 768*a^4*b^5*c^4*e*h*j*z + 256*a^3*b^7*c^3*e*g*k*z - 96*a^3*b^7*
c^3*f*g*j*z - 96*a^2*b^9*c^2*d*h*k*z - 64*a^3*b^7*c^3*e*h*j*z + 9216*a^5*b^
2*c^6*e*f*j*z - 9216*a^5*b^2*c^6*d*h*i*z - 6656*a^4*b^4*c^5*d*f*k*z - 6144*
a^5*b^2*c^6*d*f*k*z + 3456*a^3*b^6*c^4*d*f*k*z - 2304*a^4*b^4*c^5*e*f*j*z +
 2304*a^4*b^4*c^5*d*h*i*z - 576*a^2*b^8*c^3*d*f*k*z + 192*a^3*b^6*c^4*e*f*j
*z - 192*a^3*b^6*c^4*d*h*i*z + 4608*a^4*b^3*c^6*d*g*h*z + 3072*a^4*b^3*c^6*
d*f*i*z - 1152*a^3*b^5*c^5*d*g*h*z - 768*a^3*b^5*c^5*d*f*i*z + 96*a^2*b^7*c
^4*d*g*h*z + 64*a^2*b^7*c^4*d*f*i*z - 9216*a^4*b^2*c^7*d*e*h*z + 2304*a^3*b
^4*c^6*d*e*h*z + 2048*a^4*b^2*c^7*d*f*g*z - 1536*a^3*b^4*c^6*d*f*g*z + 384*
a^2*b^6*c^5*d*f*g*z - 192*a^2*b^6*c^5*d*e*h*z + 3072*a^3*b^3*c^7*d*e*f*z -
768*a^2*b^5*c^6*d*e*f*z - 3072*a^8*b*c^4*j^2*k*z + 48*a^5*b^7*c*j^2*k*z - 4
9152*a^8*b*c^4*i*k^2*z + 2304*a^5*b^7*c*i*k^2*z - 9216*a^7*b*c^5*h^2*k*z -
32*a^4*b^8*c*i*j^2*z - 1152*a^4*b^8*c*g*k^2*z + 9216*a^7*b*c^5*g*j^2*z - 30
72*a^6*b*c^6*f^2*k*z + 16*a^3*b^9*c*g*j^2*z - 49152*a^7*b*c^5*e*k^2*z - 128
*a^3*b^9*c*e*k^2*z - 58368*a^5*b*c^7*d^2*k*z - 1024*a^6*b*c^6*g*h^2*z - 432
*a*b^9*c^3*d^2*k*z + 1024*a^5*b*c^7*f^2*g*z + 32*a*b^8*c^4*d^2*i*z - 9216*a
^4*b*c^8*d^2*g*z + 336*a*b^7*c^5*d^2*g*z - 672*a*b^6*c^6*d^2*e*z + 24576*a^
8*c^5*h*j*k*z + 73728*a^7*c^6*d*j*k*z + 32768*a^7*c^6*e*i*k*z - 12288*a^7*c
^6*f*i*j*z + 8192*a^7*c^6*f*h*k*z + 24576*a^6*c^7*d*f*k*z - 12288*a^6*c^7*e
*f*j*z + 12288*a^6*c^7*d*h*i*z + 12288*a^5*c^8*d*e*h*z + 2304*a^7*b^3*c^3*j
^2*k*z - 576*a^6*b^5*c^2*j^2*k*z + 45056*a^7*b^3*c^3*i*k^2*z - 15360*a^6*b^
5*c^2*i*k^2*z - 12288*a^7*b^2*c^4*i^2*k*z + 3072*a^6*b^4*c^3*i^2*k*z - 256*
a^5*b^6*c^2*i^2*k*z + 15872*a^7*b^2*c^4*i*j^2*z + 6912*a^6*b^3*c^4*h^2*k*z
- 4992*a^6*b^4*c^3*i*j^2*z - 1728*a^5*b^5*c^3*h^2*k*z + 672*a^5*b^6*c^2*i*j
^2*z + 144*a^4*b^7*c^2*h^2*k*z + 24576*a^7*b^2*c^4*g*k^2*z - 22528*a^6*b^4*
c^3*g*k^2*z + 7680*a^5*b^6*c^2*g*k^2*z + 4096*a^6*b^2*c^5*g^2*k*z - 3072*a^
5*b^4*c^4*g^2*k*z + 768*a^4*b^6*c^3*g^2*k*z - 64*a^3*b^8*c^2*g^2*k*z - 7936
*a^6*b^3*c^4*g*j^2*z + 2496*a^5*b^5*c^3*g*j^2*z - 1536*a^6*b^2*c^5*h^2*i*z
+ 1280*a^5*b^3*c^5*f^2*k*z + 384*a^5*b^4*c^4*h^2*i*z - 336*a^4*b^7*c^2*g*j^
2*z + 192*a^4*b^5*c^4*f^2*k*z - 144*a^3*b^7*c^3*f^2*k*z - 32*a^4*b^6*c^3*h^
2*i*z + 16*a^2*b^9*c^2*f^2*k*z + 45056*a^6*b^3*c^4*e*k^2*z - 15360*a^5*b^5*
c^3*e*k^2*z - 12288*a^5*b^2*c^6*e^2*k*z + 3072*a^4*b^4*c^5*e^2*k*z + 2304*a
^4*b^7*c^2*e*k^2*z - 256*a^3*b^6*c^4*e^2*k*z + 59136*a^4*b^3*c^6*d^2*k*z -
23488*a^3*b^5*c^5*d^2*k*z + 15872*a^6*b^2*c^5*e*j^2*z - 4992*a^5*b^4*c^4*e*
j^2*z + 4560*a^2*b^7*c^4*d^2*k*z + 1536*a^5*b^2*c^6*f^2*i*z + 768*a^5*b^3*c

^5*g*h^2*z + 672*a^4*b^6*c^3*e*j^2*z - 384*a^4*b^4*c^5*f^2*i*z - 192*a^4*b^
5*c^4*g*h^2*z - 32*a^3*b^8*c^2*e*j^2*z + 32*a^3*b^6*c^4*f^2*i*z + 16*a^3*b^
7*c^3*g*h^2*z - 15872*a^4*b^2*c^7*d^2*i*z + 4992*a^3*b^4*c^6*d^2*i*z - 1536
*a^5*b^2*c^6*e*h^2*z - 768*a^4*b^3*c^6*f^2*g*z - 672*a^2*b^6*c^5*d^2*i*z +
384*a^4*b^4*c^5*e*h^2*z + 192*a^3*b^5*c^5*f^2*g*z - 32*a^3*b^6*c^4*e*h^2*z
- 16*a^2*b^7*c^4*f^2*g*z + 7936*a^3*b^3*c^7*d^2*g*z - 2496*a^2*b^5*c^6*d^2*
g*z + 1536*a^4*b^2*c^7*e*f^2*z - 384*a^3*b^4*c^6*e*f^2*z + 32*a^2*b^6*c^5*e
*f^2*z - 15872*a^3*b^2*c^8*d^2*e*z + 4992*a^2*b^4*c^7*d^2*e*z - 61440*a^8*b
^2*c^3*k^3*z + 21504*a^7*b^4*c^2*k^3*z + 16384*a^8*c^5*i^2*k*z - 18432*a^8*
c^5*i*j^2*z - 128*a^4*b^9*i*k^2*z + 2048*a^7*c^6*h^2*i*z + 64*a^3*b^10*g*k^
2*z + 16384*a^6*c^7*e^2*k*z + 16*b^11*c^2*d^2*k*z - 18432*a^7*c^6*e*j^2*z -
 2048*a^6*c^7*f^2*i*z + 18432*a^5*c^8*d^2*i*z - 3328*a^6*b^6*c*k^3*z + 2048
*a^6*c^7*e*h^2*z - 16*b^9*c^4*d^2*g*z - 2048*a^5*c^8*e*f^2*z + 32*b^8*c^5*d
^2*e*z + 18432*a^4*c^9*d^2*e*z + 65536*a^9*c^4*k^3*z + 192*a^5*b^8*k^3*z -
3328*a^7*b*c^3*h*i*j*k - 6912*a^6*b*c^4*d*i*j*k - 3328*a^6*b*c^4*e*h*j*k -
1536*a^6*b*c^4*f*g*j*k - 768*a^6*b*c^4*g*h*i*j - 768*a^6*b*c^4*f*h*i*k - 69
12*a^5*b*c^5*d*e*j*k - 2304*a^5*b*c^5*d*g*i*j - 1792*a^5*b*c^5*e*f*i*j + 15
36*a^5*b*c^5*d*g*h*k - 1280*a^5*b*c^5*d*f*i*k - 768*a^5*b*c^5*e*g*h*j - 768
*a^5*b*c^5*e*f*h*k - 256*a^5*b*c^5*f*g*h*i + 16*a*b^8*c^2*d*f*g*k - 4*a*b^8
*c^2*d*f*h*j - 2304*a^4*b*c^6*d*e*g*j - 1792*a^4*b*c^6*d*e*h*i - 1280*a^4*b
*c^6*d*e*f*k - 768*a^4*b*c^6*d*f*g*i - 256*a^4*b*c^6*e*f*g*h - 32*a*b^7*c^3
*d*e*f*k - 768*a^3*b*c^7*d*e*f*g + 32*a*b^5*c^5*d*e*f*g + 576*a^6*b^3*c^2*h
*i*j*k + 1664*a^6*b^2*c^3*g*h*j*k + 384*a^6*b^2*c^3*f*i*j*k - 288*a^5*b^4*c
^2*g*h*j*k - 160*a^5*b^4*c^2*f*i*j*k + 2112*a^5*b^3*c^3*d*i*j*k + 576*a^5*b
^3*c^3*e*h*j*k - 448*a^5*b^3*c^3*f*h*i*k - 192*a^5*b^3*c^3*g*h*i*j - 192*a^
5*b^3*c^3*f*g*j*k - 160*a^4*b^5*c^2*d*i*j*k + 96*a^4*b^5*c^2*f*h*i*k + 80*a
^4*b^5*c^2*f*g*j*k + 32*a^4*b^5*c^2*g*h*i*j + 4992*a^5*b^2*c^4*d*h*i*k - 46
08*a^5*b^2*c^4*e*g*i*k + 3456*a^5*b^2*c^4*d*g*j*k - 1312*a^4*b^4*c^3*d*h*i*
k - 1056*a^4*b^4*c^3*d*g*j*k + 896*a^5*b^2*c^4*f*g*i*j + 768*a^4*b^4*c^3*e*
g*i*k + 384*a^5*b^2*c^4*f*g*h*k + 384*a^5*b^2*c^4*e*h*i*j + 384*a^5*b^2*c^4
*e*f*j*k + 224*a^4*b^4*c^3*f*g*h*k - 160*a^4*b^4*c^3*e*f*j*k - 96*a^4*b^4*c
^3*f*g*i*j + 96*a^3*b^6*c^2*d*h*i*k + 80*a^3*b^6*c^2*d*g*j*k - 64*a^4*b^4*c
^3*e*h*i*j - 48*a^3*b^6*c^2*f*g*h*k - 2496*a^4*b^3*c^4*d*g*h*k + 2112*a^4*b
^3*c^4*d*e*j*k - 960*a^4*b^3*c^4*d*f*i*k + 656*a^3*b^5*c^3*d*g*h*k - 448*a^
4*b^3*c^4*e*f*h*k + 384*a^3*b^5*c^3*d*f*i*k + 320*a^4*b^3*c^4*d*g*i*j - 192
*a^4*b^3*c^4*f*g*h*i - 192*a^4*b^3*c^4*e*g*h*j + 192*a^4*b^3*c^4*e*f*i*j -
160*a^3*b^5*c^3*d*e*j*k + 96*a^3*b^5*c^3*e*f*h*k - 48*a^2*b^7*c^2*d*g*h*k +
 32*a^3*b^5*c^3*e*g*h*j - 32*a^2*b^7*c^2*d*f*i*k + 4992*a^4*b^2*c^5*d*e*h*k
 - 3584*a^4*b^2*c^5*d*f*h*j - 1312*a^3*b^4*c^4*d*e*h*k + 896*a^4*b^2*c^5*e*
f*g*j + 896*a^4*b^2*c^5*d*g*h*i + 640*a^4*b^2*c^5*d*f*g*k - 640*a^4*b^2*c^5
*d*e*i*j + 600*a^3*b^4*c^4*d*f*h*j + 480*a^3*b^4*c^4*d*f*g*k + 384*a^4*b^2*
c^5*e*f*h*i - 192*a^2*b^6*c^3*d*f*g*k - 96*a^3*b^4*c^4*e*f*g*j - 96*a^3*b^4
*c^4*d*g*h*i + 96*a^2*b^6*c^3*d*e*h*k + 12*a^2*b^6*c^3*d*f*h*j - 960*a^3*b^
3*c^5*d*e*f*k + 384*a^2*b^5*c^4*d*e*f*k + 320*a^3*b^3*c^5*d*e*g*j - 192*a^3
*b^3*c^5*e*f*g*h - 192*a^3*b^3*c^5*d*f*g*i + 192*a^3*b^3*c^5*d*e*h*i + 32*a

^2*b^5*c^4*d*f*g*i + 896*a^3*b^2*c^6*d*e*g*h + 384*a^3*b^2*c^6*d*e*f*i - 96
*a^2*b^4*c^5*d*e*g*h - 64*a^2*b^4*c^5*d*e*f*i - 192*a^2*b^3*c^6*d*e*f*g + 4
8*a^6*b^4*c*i*j^2*k - 1424*a^6*b^4*c*h*j*k^2 - 2304*a^7*b*c^3*g*j^2*k - 24*
a^5*b^5*c*g*j^2*k + 2048*a^7*b*c^3*g*i*k^2 - 1024*a^7*b*c^3*f*j*k^2 - 768*a
^5*b^5*c*g*i*k^2 + 408*a^5*b^5*c*f*j*k^2 + 256*a^6*b*c^4*g*h^2*k + 16*a^4*b
^6*c*g*i*j^2 + 4608*a^6*b*c^4*e*i^2*k + 4608*a^5*b*c^5*e^2*i*k - 896*a^6*b*
c^4*f*i^2*j + 768*a^4*b^6*c*d*j*k^2 - 256*a^4*b^6*c*f*h*k^2 - 128*a^4*b^6*c
*e*i*k^2 + 2208*a^6*b*c^4*f*h*j^2 - 1920*a^6*b*c^4*e*i*j^2 + 800*a^5*b*c^5*
f^2*h*j - 256*a^5*b*c^5*f^2*g*k - 16*a*b^8*c^2*d^2*i*k + 6*a^3*b^7*c*f*h*j^
2 + 8192*a^6*b*c^4*d*h*k^2 + 2048*a^6*b*c^4*e*g*k^2 - 472*a^3*b^7*c*d*h*k^2
 + 64*a^3*b^7*c*e*g*k^2 + 4896*a^4*b*c^6*d^2*h*j + 2304*a^4*b*c^6*d^2*g*k +
 1824*a^5*b*c^5*d*h^2*j - 384*a^5*b*c^5*e*h^2*i - 168*a*b^7*c^3*d^2*g*k + 4
2*a*b^7*c^3*d^2*h*j + 6*a^2*b^8*c*d*h*j^2 + 1536*a^5*b*c^5*e*g*i^2 + 1536*a
^4*b*c^6*e^2*g*i - 896*a^5*b*c^5*d*h*i^2 - 896*a^4*b*c^6*e^2*f*j + 144*a^2*
b^8*c*d*f*k^2 + 4896*a^5*b*c^5*d*f*j^2 + 1824*a^4*b*c^6*d*f^2*j - 384*a^4*b
*c^6*e*f^2*i + 336*a*b^6*c^4*d^2*e*k - 156*a*b^6*c^4*d^2*f*j + 16*a*b^6*c^4
*d^2*g*i + 12*a*b^7*c^3*d*f^2*j + 2208*a^3*b*c^7*d^2*f*h - 1920*a^3*b*c^7*d
^2*e*i + 800*a^4*b*c^6*d*f*h^2 - 102*a*b^5*c^5*d^2*f*h - 32*a*b^5*c^5*d^2*e
*i + 12*a*b^6*c^4*d*f^2*h - 2*a*b^7*c^3*d*f*h^2 - 896*a^3*b*c^7*d*e^2*h - 8
*a*b^6*c^4*d*f*g^2 - 240*a*b^4*c^6*d^2*e*g - 32*a*b^4*c^6*d*e^2*f + 3072*a^
7*c^4*f*i*j*k + 3072*a^6*c^5*e*f*j*k - 3072*a^6*c^5*d*h*i*k + 1536*a^6*c^5*
e*h*i*j + 4608*a^5*c^6*d*e*i*j - 3072*a^5*c^6*d*e*h*k - 1152*a^5*c^6*d*f*h*
j + 512*a^5*c^6*e*f*h*i + 1536*a^4*c^7*d*e*f*i - 2*a*b^9*c*d*f*j^2 - 1088*a
^7*b^2*c^2*i*j^2*k + 4800*a^7*b^2*c^2*h*j*k^2 + 960*a^6*b^2*c^3*h^2*i*k + 5
44*a^6*b^3*c^2*g*j^2*k - 144*a^5*b^4*c^2*h^2*i*k - 2304*a^6*b^2*c^3*g*i^2*k
 + 1920*a^6*b^3*c^2*g*i*k^2 + 1152*a^5*b^3*c^3*g^2*i*k - 864*a^6*b^3*c^2*f*
j*k^2 + 384*a^5*b^4*c^2*g*i^2*k + 192*a^6*b^2*c^3*h*i^2*j - 192*a^4*b^5*c^2
*g^2*i*k - 32*a^5*b^4*c^2*h*i^2*j - 1088*a^6*b^2*c^3*e*j^2*k + 960*a^6*b^2*
c^3*g*i*j^2 - 480*a^5*b^3*c^3*g*h^2*k - 240*a^5*b^4*c^2*g*i*j^2 + 192*a^5*b
^2*c^4*f^2*i*k + 72*a^4*b^5*c^2*g*h^2*k + 48*a^5*b^4*c^2*e*j^2*k + 48*a^4*b
^4*c^3*f^2*i*k - 16*a^3*b^6*c^2*f^2*i*k + 13376*a^6*b^2*c^3*d*j*k^2 - 5136*
a^5*b^4*c^2*d*j*k^2 - 3840*a^6*b^2*c^3*e*i*k^2 + 1536*a^5*b^4*c^2*e*i*k^2 -
 768*a^5*b^3*c^3*e*i^2*k - 768*a^4*b^3*c^4*e^2*i*k + 624*a^5*b^4*c^2*f*h*k^
2 + 576*a^6*b^2*c^3*f*h*k^2 + 192*a^5*b^2*c^4*g^2*h*j + 96*a^5*b^3*c^3*f*i^
2*j + 48*a^4*b^4*c^3*g^2*h*j - 8*a^3*b^6*c^2*g^2*h*j + 6848*a^4*b^2*c^5*d^2
*i*k - 2448*a^3*b^4*c^4*d^2*i*k + 960*a^5*b^2*c^4*e*h^2*k - 864*a^5*b^2*c^4
*f*h^2*j + 480*a^5*b^3*c^3*e*i*j^2 + 336*a^4*b^3*c^4*f^2*h*j + 336*a^2*b^6*
c^3*d^2*i*k + 192*a^5*b^2*c^4*g*h^2*i + 144*a^5*b^3*c^3*f*h*j^2 - 144*a^4*b
^4*c^3*e*h^2*k - 102*a^4*b^5*c^2*f*h*j^2 - 96*a^4*b^3*c^4*f^2*g*k - 32*a^4*
b^5*c^2*e*i*j^2 - 30*a^3*b^5*c^3*f^2*h*j - 24*a^3*b^5*c^3*f^2*g*k + 16*a^4*
b^4*c^3*g*h^2*i - 12*a^4*b^4*c^3*f*h^2*j + 12*a^3*b^6*c^2*f*h^2*j + 8*a^2*b
^7*c^2*f^2*g*k - 2*a^2*b^7*c^2*f^2*h*j - 9312*a^5*b^3*c^3*d*h*k^2 + 3288*a^
4*b^5*c^2*d*h*k^2 - 2304*a^4*b^2*c^5*e^2*g*k + 1920*a^5*b^3*c^3*e*g*k^2 + 1
152*a^4*b^3*c^4*e*g^2*k - 768*a^4*b^5*c^2*e*g*k^2 + 384*a^3*b^4*c^4*e^2*g*k
 - 320*a^5*b^2*c^4*d*i^2*j - 224*a^4*b^3*c^4*f*g^2*j + 192*a^5*b^2*c^4*f*h*

i^2 + 192*a^4*b^2*c^5*e^2*h*j - 192*a^3*b^5*c^3*e*g^2*k - 32*a^3*b^4*c^4*e^
2*h*j + 24*a^3*b^5*c^3*f*g^2*j - 3552*a^5*b^2*c^4*d*h*j^2 - 3424*a^3*b^3*c^
5*d^2*g*k + 1332*a^4*b^4*c^3*d*h*j^2 + 1224*a^2*b^5*c^4*d^2*g*k + 960*a^5*b
^2*c^4*e*g*j^2 - 496*a^3*b^3*c^5*d^2*h*j + 432*a^4*b^3*c^4*d*h^2*j - 240*a^
4*b^4*c^3*e*g*j^2 - 222*a^2*b^5*c^4*d^2*h*j + 192*a^4*b^2*c^5*f^2*g*i + 192
*a^4*b^2*c^5*e*f^2*k - 174*a^3*b^5*c^3*d*h^2*j - 156*a^3*b^6*c^2*d*h*j^2 +
48*a^3*b^4*c^4*e*f^2*k - 32*a^4*b^3*c^4*e*h^2*i + 16*a^3*b^6*c^2*e*g*j^2 +
16*a^3*b^4*c^4*f^2*g*i - 16*a^2*b^6*c^3*e*f^2*k + 12*a^2*b^7*c^2*d*h^2*j +
1728*a^5*b^2*c^4*d*f*k^2 + 1392*a^4*b^4*c^3*d*f*k^2 - 840*a^3*b^6*c^2*d*f*k
^2 - 768*a^4*b^2*c^5*e*g^2*i + 576*a^4*b^2*c^5*d*g^2*j + 96*a^4*b^3*c^4*d*h
*i^2 + 96*a^3*b^3*c^5*e^2*f*j - 80*a^3*b^4*c^4*d*g^2*j + 64*a^4*b^2*c^5*f*g
^2*h + 48*a^3*b^4*c^4*f*g^2*h + 6848*a^3*b^2*c^6*d^2*e*k - 3552*a^3*b^2*c^6
*d^2*f*j - 2448*a^2*b^4*c^5*d^2*e*k + 1332*a^2*b^4*c^5*d^2*f*j + 960*a^3*b^
2*c^6*d^2*g*i - 496*a^4*b^3*c^4*d*f*j^2 + 432*a^3*b^3*c^5*d*f^2*j - 240*a^2
*b^4*c^5*d^2*g*i - 222*a^3*b^5*c^3*d*f*j^2 + 192*a^4*b^2*c^5*e*g*h^2 - 174*
a^2*b^5*c^4*d*f^2*j + 42*a^2*b^7*c^2*d*f*j^2 - 32*a^3*b^3*c^5*e*f^2*i + 16*
a^3*b^4*c^4*e*g*h^2 - 320*a^3*b^2*c^6*d*e^2*j - 224*a^3*b^3*c^5*d*g^2*h + 1
92*a^4*b^2*c^5*d*f*i^2 + 192*a^3*b^2*c^6*e^2*f*h - 32*a^3*b^4*c^4*d*f*i^2 +
 24*a^2*b^5*c^4*d*g^2*h - 864*a^3*b^2*c^6*d*f^2*h + 480*a^2*b^3*c^6*d^2*e*i
 + 336*a^3*b^3*c^5*d*f*h^2 + 192*a^3*b^2*c^6*e*f^2*g + 144*a^2*b^3*c^6*d^2*
f*h - 30*a^2*b^5*c^4*d*f*h^2 + 16*a^2*b^4*c^5*e*f^2*g - 12*a^2*b^4*c^5*d*f^
2*h + 192*a^3*b^2*c^6*d*f*g^2 + 96*a^2*b^3*c^6*d*e^2*h + 48*a^2*b^4*c^5*d*f
*g^2 + 960*a^2*b^2*c^7*d^2*e*g + 192*a^2*b^2*c^7*d*e^2*f - 3072*a^8*b*c^2*j
^2*k^2 + 1104*a^7*b^3*c*j^2*k^2 + 768*a^6*b^4*c*i^2*k^2 - 256*a^6*b^3*c^2*i
^3*k + 1536*a^7*b*c^3*h^2*k^2 - 960*a^7*b*c^3*i^2*j^2 + 444*a^5*b^5*c*h^2*k
^2 - 16*a^5*b^5*c*i^2*j^2 - 3072*a^7*b^2*c^2*g*k^3 - 496*a^6*b^3*c^2*h*j^3
+ 192*a^4*b^6*c*g^2*k^2 - 192*a^4*b^4*c^3*g^3*k + 144*a^5*b^3*c^3*h^3*j + 3
2*a^3*b^6*c^2*g^3*k - 18*a^4*b^5*c^2*h^3*j - 9*a^4*b^6*c*h^2*j^2 - 192*a^6*
b*c^4*h^2*i^2 + 36*a^3*b^7*c*f^2*k^2 - 4*a^3*b^7*c*g^2*j^2 - 2176*a^6*b^3*c
^2*e*k^3 - 256*a^3*b^3*c^5*e^3*k - 192*a^6*b^2*c^3*f*j^3 - 192*a^4*b^2*c^5*
f^3*j + 132*a^5*b^4*c^2*f*j^3 + 128*a^4*b^3*c^4*g^3*i - 28*a^3*b^4*c^4*f^3*
j + 6*a^2*b^6*c^3*f^3*j + 10752*a^5*b*c^5*d^2*k^2 - 960*a^5*b*c^5*e^2*j^2 -
 192*a^5*b*c^5*f^2*i^2 - 1680*a^5*b^3*c^3*d*j^3 - 1680*a^2*b^3*c^6*d^3*j +
222*a^4*b^5*c^2*d*j^3 + 80*a^4*b^3*c^4*f*h^3 + 80*a^3*b^3*c^5*f^3*h + 30*a*
b^8*c^2*d^2*j^2 + 6*a^3*b^5*c^3*f*h^3 + 6*a^2*b^5*c^4*f^3*h - 960*a^4*b*c^6
*d^2*i^2 - 192*a^4*b*c^6*e^2*h^2 - 192*a^4*b^2*c^5*d*h^3 - 192*a^2*b^2*c^7*
d^3*h + 128*a^3*b^3*c^5*e*g^3 - 28*a^3*b^4*c^4*d*h^3 + 12*a*b^6*c^4*d^2*h^2
 + 6*a^2*b^6*c^3*d*h^3 - 192*a^3*b*c^7*e^2*f^2 + 60*a*b^5*c^5*d^2*g^2 + 198
*a*b^4*c^6*d^2*f^2 + 144*a^2*b^3*c^6*d*f^3 - 960*a^2*b*c^8*d^2*e^2 + 240*a*
b^3*c^7*d^2*e^2 + 4608*a^8*c^3*i*j^2*k - 3072*a^8*c^3*h*j*k^2 - 512*a^7*c^4
*h^2*i*k + 120*a^5*b^6*h*j*k^2 + 768*a^7*c^4*h*i^2*j + 4608*a^7*c^4*e*j^2*k
 + 512*a^6*c^5*f^2*i*k + 64*a^4*b^7*g*i*k^2 - 40*a^4*b^7*f*j*k^2 - 9216*a^7
*c^4*d*j*k^2 - 4096*a^7*c^4*e*i*k^2 - 1024*a^7*c^4*f*h*k^2 - 4608*a^5*c^6*d
^2*i*k - 512*a^6*c^5*e*h^2*k - 192*a^6*c^5*f*h^2*j - 40*a^3*b^8*d*j*k^2 + 2
4*a^3*b^8*f*h*k^2 + 2304*a^6*c^5*d*i^2*j + 768*a^5*c^6*e^2*h*j + 256*a^6*c^

5*f*h*i^2 + 8*b^9*c^2*d^2*g*k - 2*b^9*c^2*d^2*h*j + 6144*a^8*b*c^2*i*k^3 -
2176*a^7*b^3*c*i*k^3 - 1728*a^6*c^5*d*h*j^2 + 1536*a^7*b*c^3*i^3*k + 512*a^
5*c^6*e*f^2*k + 24*a^2*b^9*d*h*k^2 - 3072*a^6*c^5*d*f*k^2 - 16*b^8*c^3*d^2*
e*k + 6*b^8*c^3*d^2*f*j - 4608*a^4*c^7*d^2*e*k + 2016*a^7*b*c^3*h*j^3 - 172
8*a^4*c^7*d^2*f*j + 1088*a^6*b^4*c*g*k^3 + 224*a^6*b*c^4*h^3*j + 30*a^5*b^5
*c*h*j^3 + 2304*a^4*c^7*d*e^2*j + 768*a^5*c^6*d*f*i^2 + 256*a^4*c^7*e^2*f*h
 + 6*b^7*c^4*d^2*f*h + 6144*a^7*b*c^3*e*k^3 + 1536*a^4*b*c^6*e^3*k + 512*a^
6*b*c^4*g*i^3 + 192*a^5*b^5*c*e*k^3 - 192*a^4*c^7*d*f^2*h - 10*a^4*b^6*c*f*
j^3 + 108*a*b^9*c*d^2*k^2 + 16*b^6*c^5*d^2*e*g + 4320*a^6*b*c^4*d*j^3 + 432
0*a^3*b*c^7*d^3*j + 222*a*b^5*c^5*d^3*j + 96*a^5*b*c^5*f*h^3 + 96*a^4*b*c^6
*f^3*h - 10*a^3*b^7*c*d*j^3 + 768*a^3*c^8*d*e^2*f + 512*a^3*b*c^7*e^3*g + 1
32*a*b^4*c^6*d^3*h + 2016*a^2*b*c^8*d^3*f - 496*a*b^3*c^7*d^3*f + 224*a^3*b
*c^7*d*f^3 - 18*a*b^5*c^5*d*f^3 - 1920*a^7*b^2*c^2*i^2*k^2 - 1648*a^6*b^3*c
^2*h^2*k^2 + 240*a^6*b^3*c^2*i^2*j^2 - 960*a^6*b^2*c^3*h^2*j^2 - 512*a^6*b^
2*c^3*g^2*k^2 - 480*a^5*b^4*c^2*g^2*k^2 + 198*a^5*b^4*c^2*h^2*j^2 - 240*a^5
*b^3*c^3*g^2*j^2 - 240*a^5*b^3*c^3*f^2*k^2 + 60*a^4*b^5*c^2*g^2*j^2 - 36*a^
4*b^5*c^2*f^2*k^2 - 16*a^5*b^3*c^3*h^2*i^2 - 1920*a^5*b^2*c^4*e^2*k^2 + 768
*a^4*b^4*c^3*e^2*k^2 - 464*a^5*b^2*c^4*f^2*j^2 - 384*a^5*b^2*c^4*g^2*i^2 -
64*a^3*b^6*c^2*e^2*k^2 + 42*a^4*b^4*c^3*f^2*j^2 + 12*a^3*b^6*c^2*f^2*j^2 -
13104*a^4*b^3*c^4*d^2*k^2 + 5628*a^3*b^5*c^3*d^2*k^2 - 1128*a^2*b^7*c^2*d^2
*k^2 + 240*a^4*b^3*c^4*e^2*j^2 - 48*a^4*b^3*c^4*g^2*h^2 - 16*a^4*b^3*c^4*f^
2*i^2 - 16*a^3*b^5*c^3*e^2*j^2 - 4*a^3*b^5*c^3*g^2*h^2 - 2880*a^4*b^2*c^5*d
^2*j^2 + 1750*a^3*b^4*c^4*d^2*j^2 - 345*a^2*b^6*c^3*d^2*j^2 - 192*a^4*b^2*c
^5*f^2*h^2 - 42*a^3*b^4*c^4*f^2*h^2 + 240*a^3*b^3*c^5*d^2*i^2 - 48*a^3*b^3*
c^5*f^2*g^2 - 16*a^3*b^3*c^5*e^2*h^2 - 16*a^2*b^5*c^4*d^2*i^2 - 4*a^2*b^5*c
^4*f^2*g^2 - 464*a^3*b^2*c^6*d^2*h^2 - 384*a^3*b^2*c^6*e^2*g^2 + 42*a^2*b^4
*c^5*d^2*h^2 - 240*a^2*b^3*c^6*d^2*g^2 - 16*a^2*b^3*c^6*e^2*f^2 - 960*a^2*b
^2*c^7*d^2*f^2 - 8*a*b^10*d*f*k^2 - a^2*b^8*c*f^2*j^2 - 2048*a^8*c^3*i^2*k^
2 - 100*a^6*b^5*j^2*k^2 - 64*a^5*b^6*i^2*k^2 - 288*a^7*c^4*h^2*j^2 - 36*a^4
*b^7*h^2*k^2 - 16*a^3*b^8*g^2*k^2 - 2048*a^6*c^5*e^2*k^2 - 864*a^6*c^5*f^2*
j^2 - 4*a^2*b^9*f^2*k^2 - 2592*a^5*c^6*d^2*j^2 - 1536*a^5*c^6*e^2*i^2 - 32*
a^5*c^6*f^2*h^2 - 864*a^4*c^7*d^2*h^2 + 360*a^7*b^2*c^2*j^4 - 4*b^7*c^4*d^2
*g^2 - 9*b^6*c^5*d^2*f^2 - 288*a^3*c^8*d^2*f^2 - 24*a^5*b^2*c^4*h^4 - 16*b^
5*c^6*d^2*e^2 - 9*a^4*b^4*c^3*h^4 - 16*a^3*b^4*c^4*g^4 - 24*a^3*b^2*c^6*f^4
 - 9*a^2*b^4*c^5*f^4 - a^2*b^6*c^3*f^2*h^2 + 192*a^6*b^5*i*k^3 - 96*a^5*b^6
*g*k^3 - 1728*a^7*c^4*f*j^3 - 192*a^5*c^6*f^3*j - 10*b^7*c^4*d^3*j - 1024*a
^6*c^5*e*i^3 - 1024*a^4*c^7*e^3*i + 1536*a^8*b^2*c*k^4 - 10*b^6*c^5*d^3*h -
 1728*a^3*c^8*d^3*h - 192*a^5*c^6*d*h^3 - 25*a^6*b^4*c*j^4 + 30*b^5*c^6*d^3
*f + 360*a*b^2*c^8*d^4 - 4*b^11*d^2*k^2 - 4096*a^9*c^2*k^4 - 1296*a^8*c^3*j
^4 - 144*a^7*b^4*k^4 - 256*a^7*c^4*i^4 - 16*a^6*c^5*h^4 - 16*a^4*c^7*f^4 -
256*a^3*c^8*e^4 - 25*b^4*c^7*d^4 - 1296*a^2*c^9*d^4 - b^8*c^3*d^2*h^2 - b^1
0*c*d^2*j^2, z, n)*((6144*a^5*c^8*d + 2048*a^6*c^7*h - 288*a^2*b^6*c^5*d +
1920*a^3*b^4*c^6*d - 5632*a^4*b^2*c^7*d + 16*a^2*b^7*c^4*f - 192*a^3*b^5*c^
5*f + 768*a^4*b^3*c^6*f - 32*a^3*b^6*c^4*h + 384*a^4*b^4*c^5*h - 1536*a^5*b
^2*c^6*h + 16*a^3*b^7*c^3*j - 192*a^4*b^5*c^4*j + 768*a^5*b^3*c^5*j + 16*a*

b^8*c^4*d - 1024*a^5*b*c^7*f - 1024*a^6*b*c^6*j)/(8*(64*a^5*c^5 - a^2*b^6*c
^2 + 12*a^3*b^4*c^3 - 48*a^4*b^2*c^4)) + (x*(32*a^2*b^6*c^5*e - 2048*a^6*c^
7*i - 2048*a^5*c^8*e - 384*a^3*b^4*c^6*e + 1536*a^4*b^2*c^7*e - 16*a^2*b^7*
c^4*g + 192*a^3*b^5*c^5*g - 768*a^4*b^3*c^6*g + 32*a^3*b^6*c^4*i - 384*a^4*
b^4*c^5*i + 1536*a^5*b^2*c^6*i + 32*a^2*b^9*c^2*k - 528*a^3*b^7*c^3*k + 326
4*a^4*b^5*c^4*k - 8960*a^5*b^3*c^5*k + 1024*a^5*b*c^7*g + 9216*a^6*b*c^6*k)
)/(4*(64*a^5*c^5 - a^2*b^6*c^2 + 12*a^3*b^4*c^3 - 48*a^4*b^2*c^4)) - (root(
1572864*a^8*b^2*c^9*z^4 - 983040*a^7*b^4*c^8*z^4 + 327680*a^6*b^6*c^7*z^4 -
 61440*a^5*b^8*c^6*z^4 + 6144*a^4*b^10*c^5*z^4 - 256*a^3*b^12*c^4*z^4 - 104
8576*a^9*c^10*z^4 - 1572864*a^8*b^2*c^7*k*z^3 + 983040*a^7*b^4*c^6*k*z^3 -
327680*a^6*b^6*c^5*k*z^3 + 61440*a^5*b^8*c^4*k*z^3 - 6144*a^4*b^10*c^3*k*z^
3 + 256*a^3*b^12*c^2*k*z^3 + 1048576*a^9*c^8*k*z^3 + 98304*a^8*b*c^6*i*k*z^
2 + 98304*a^7*b*c^7*e*k*z^2 + 57344*a^7*b*c^7*f*j*z^2 + 32768*a^7*b*c^7*g*i
*z^2 + 57344*a^6*b*c^8*d*h*z^2 + 32768*a^6*b*c^8*e*g*z^2 - 32*a*b^10*c^4*d*
f*z^2 - 90112*a^7*b^3*c^5*i*k*z^2 + 30720*a^6*b^5*c^4*i*k*z^2 - 4608*a^5*b^
7*c^3*i*k*z^2 + 256*a^4*b^9*c^2*i*k*z^2 - 49152*a^7*b^2*c^6*g*k*z^2 + 45056
*a^6*b^4*c^5*g*k*z^2 + 24576*a^7*b^2*c^6*h*j*z^2 - 15360*a^5*b^6*c^4*g*k*z^
2 - 3072*a^5*b^6*c^4*h*j*z^2 + 2304*a^4*b^8*c^3*g*k*z^2 + 2048*a^6*b^4*c^5*
h*j*z^2 + 576*a^4*b^8*c^3*h*j*z^2 - 128*a^3*b^10*c^2*g*k*z^2 - 32*a^3*b^10*
c^2*h*j*z^2 - 90112*a^6*b^3*c^6*e*k*z^2 - 49152*a^6*b^3*c^6*f*j*z^2 + 30720
*a^5*b^5*c^5*e*k*z^2 - 24576*a^6*b^3*c^6*g*i*z^2 + 15360*a^5*b^5*c^5*f*j*z^
2 + 6144*a^5*b^5*c^5*g*i*z^2 - 4608*a^4*b^7*c^4*e*k*z^2 - 2048*a^4*b^7*c^4*
f*j*z^2 - 512*a^4*b^7*c^4*g*i*z^2 + 256*a^3*b^9*c^3*e*k*z^2 + 96*a^3*b^9*c^
3*f*j*z^2 + 131072*a^6*b^2*c^7*d*j*z^2 + 49152*a^6*b^2*c^7*e*i*z^2 - 43008*
a^5*b^4*c^6*d*j*z^2 - 12288*a^5*b^4*c^6*e*i*z^2 + 6144*a^5*b^4*c^6*f*h*z^2
+ 6144*a^4*b^6*c^5*d*j*z^2 - 2048*a^4*b^6*c^5*f*h*z^2 + 1024*a^4*b^6*c^5*e*
i*z^2 - 320*a^3*b^8*c^4*d*j*z^2 + 192*a^3*b^8*c^4*f*h*z^2 - 49152*a^5*b^3*c
^7*d*h*z^2 - 24576*a^5*b^3*c^7*e*g*z^2 + 15360*a^4*b^5*c^6*d*h*z^2 + 6144*a
^4*b^5*c^6*e*g*z^2 - 2048*a^3*b^7*c^5*d*h*z^2 - 512*a^3*b^7*c^5*e*g*z^2 + 9
6*a^2*b^9*c^4*d*h*z^2 + 24576*a^5*b^2*c^8*d*f*z^2 - 3072*a^3*b^6*c^6*d*f*z^
2 + 2048*a^4*b^4*c^7*d*f*z^2 + 576*a^2*b^8*c^5*d*f*z^2 + 1536*a^4*b^10*c*k^
2*z^2 + 61440*a^8*b*c^6*j^2*z^2 - 16*a^3*b^11*c*j^2*z^2 + 12288*a^7*b*c^7*h
^2*z^2 + 12288*a^6*b*c^8*f^2*z^2 + 61440*a^5*b*c^9*d^2*z^2 + 432*a*b^9*c^5*
d^2*z^2 - 49152*a^8*c^7*h*j*z^2 - 147456*a^7*c^8*d*j*z^2 - 65536*a^7*c^8*e*
i*z^2 - 16384*a^7*c^8*f*h*z^2 - 49152*a^6*c^9*d*f*z^2 + 516096*a^8*b^2*c^5*
k^2*z^2 - 288768*a^7*b^4*c^4*k^2*z^2 + 88576*a^6*b^6*c^3*k^2*z^2 - 15744*a^
5*b^8*c^2*k^2*z^2 - 61440*a^7*b^3*c^5*j^2*z^2 + 24064*a^6*b^5*c^4*j^2*z^2 -
 4608*a^5*b^7*c^3*j^2*z^2 + 432*a^4*b^9*c^2*j^2*z^2 + 24576*a^7*b^2*c^6*i^2
*z^2 - 6144*a^6*b^4*c^5*i^2*z^2 + 512*a^5*b^6*c^4*i^2*z^2 - 8192*a^6*b^3*c^
6*h^2*z^2 + 1536*a^5*b^5*c^5*h^2*z^2 - 16*a^3*b^9*c^3*h^2*z^2 - 8192*a^6*b^
2*c^7*g^2*z^2 + 6144*a^5*b^4*c^6*g^2*z^2 - 1536*a^4*b^6*c^5*g^2*z^2 + 128*a
^3*b^8*c^4*g^2*z^2 - 8192*a^5*b^3*c^7*f^2*z^2 + 1536*a^4*b^5*c^6*f^2*z^2 -
16*a^2*b^9*c^4*f^2*z^2 + 24576*a^5*b^2*c^8*e^2*z^2 - 6144*a^4*b^4*c^7*e^2*z
^2 + 512*a^3*b^6*c^6*e^2*z^2 - 61440*a^4*b^3*c^8*d^2*z^2 + 24064*a^3*b^5*c^
7*d^2*z^2 - 4608*a^2*b^7*c^6*d^2*z^2 - 393216*a^9*c^6*k^2*z^2 - 64*a^3*b^12

*k^2*z^2 - 32768*a^8*c^7*i^2*z^2 - 32768*a^6*c^9*e^2*z^2 - 16*b^11*c^4*d^2*
z^2 - 16384*a^7*b*c^5*g*i*k*z - 10240*a^7*b*c^5*f*j*k*z + 4096*a^7*b*c^5*h*
i*j*z - 47104*a^6*b*c^6*d*h*k*z - 16384*a^6*b*c^6*e*g*k*z + 6144*a^6*b*c^6*
f*g*j*z + 4096*a^6*b*c^6*e*h*j*z + 32*a*b^10*c^2*d*f*k*z - 6144*a^5*b*c^7*d
*g*h*z - 4096*a^5*b*c^7*d*f*i*z - 32*a*b^8*c^4*d*f*g*z - 4096*a^4*b*c^8*d*e
*f*z + 64*a*b^7*c^5*d*e*f*z - 18432*a^7*b^2*c^4*h*j*k*z + 4608*a^6*b^4*c^3*
h*j*k*z - 384*a^5*b^6*c^2*h*j*k*z + 12288*a^6*b^3*c^4*g*i*k*z + 7680*a^6*b^
3*c^4*f*j*k*z - 3072*a^6*b^3*c^4*h*i*j*z - 3072*a^5*b^5*c^3*g*i*k*z - 1920*
a^5*b^5*c^3*f*j*k*z + 768*a^5*b^5*c^3*h*i*j*z + 256*a^4*b^7*c^2*g*i*k*z + 1
60*a^4*b^7*c^2*f*j*k*z - 64*a^4*b^7*c^2*h*i*j*z - 65536*a^6*b^2*c^5*d*j*k*z
 - 24576*a^6*b^2*c^5*e*i*k*z + 21504*a^5*b^4*c^4*d*j*k*z + 9216*a^6*b^2*c^5
*f*i*j*z + 6144*a^5*b^4*c^4*e*i*k*z - 3072*a^5*b^4*c^4*f*h*k*z - 3072*a^4*b
^6*c^3*d*j*k*z - 2304*a^5*b^4*c^4*f*i*j*z - 2048*a^6*b^2*c^5*g*h*j*z + 1536
*a^5*b^4*c^4*g*h*j*z + 1024*a^4*b^6*c^3*f*h*k*z - 512*a^4*b^6*c^3*e*i*k*z -
 384*a^4*b^6*c^3*g*h*j*z + 192*a^4*b^6*c^3*f*i*j*z + 160*a^3*b^8*c^2*d*j*k*
z - 96*a^3*b^8*c^2*f*h*k*z + 32*a^3*b^8*c^2*g*h*j*z + 41472*a^5*b^3*c^5*d*h
*k*z - 13440*a^4*b^5*c^4*d*h*k*z + 12288*a^5*b^3*c^5*e*g*k*z - 4608*a^5*b^3
*c^5*f*g*j*z - 3072*a^5*b^3*c^5*e*h*j*z - 3072*a^4*b^5*c^4*e*g*k*z + 1888*a
^3*b^7*c^3*d*h*k*z + 1152*a^4*b^5*c^4*f*g*j*z + 768*a^4*b^5*c^4*e*h*j*z + 2
56*a^3*b^7*c^3*e*g*k*z - 96*a^3*b^7*c^3*f*g*j*z - 96*a^2*b^9*c^2*d*h*k*z -
64*a^3*b^7*c^3*e*h*j*z + 9216*a^5*b^2*c^6*e*f*j*z - 9216*a^5*b^2*c^6*d*h*i*
z - 6656*a^4*b^4*c^5*d*f*k*z - 6144*a^5*b^2*c^6*d*f*k*z + 3456*a^3*b^6*c^4*
d*f*k*z - 2304*a^4*b^4*c^5*e*f*j*z + 2304*a^4*b^4*c^5*d*h*i*z - 576*a^2*b^8
*c^3*d*f*k*z + 192*a^3*b^6*c^4*e*f*j*z - 192*a^3*b^6*c^4*d*h*i*z + 4608*a^4
*b^3*c^6*d*g*h*z + 3072*a^4*b^3*c^6*d*f*i*z - 1152*a^3*b^5*c^5*d*g*h*z - 76
8*a^3*b^5*c^5*d*f*i*z + 96*a^2*b^7*c^4*d*g*h*z + 64*a^2*b^7*c^4*d*f*i*z - 9
216*a^4*b^2*c^7*d*e*h*z + 2304*a^3*b^4*c^6*d*e*h*z + 2048*a^4*b^2*c^7*d*f*g
*z - 1536*a^3*b^4*c^6*d*f*g*z + 384*a^2*b^6*c^5*d*f*g*z - 192*a^2*b^6*c^5*d
*e*h*z + 3072*a^3*b^3*c^7*d*e*f*z - 768*a^2*b^5*c^6*d*e*f*z - 3072*a^8*b*c^
4*j^2*k*z + 48*a^5*b^7*c*j^2*k*z - 49152*a^8*b*c^4*i*k^2*z + 2304*a^5*b^7*c
*i*k^2*z - 9216*a^7*b*c^5*h^2*k*z - 32*a^4*b^8*c*i*j^2*z - 1152*a^4*b^8*c*g
*k^2*z + 9216*a^7*b*c^5*g*j^2*z - 3072*a^6*b*c^6*f^2*k*z + 16*a^3*b^9*c*g*j
^2*z - 49152*a^7*b*c^5*e*k^2*z - 128*a^3*b^9*c*e*k^2*z - 58368*a^5*b*c^7*d^
2*k*z - 1024*a^6*b*c^6*g*h^2*z - 432*a*b^9*c^3*d^2*k*z + 1024*a^5*b*c^7*f^2
*g*z + 32*a*b^8*c^4*d^2*i*z - 9216*a^4*b*c^8*d^2*g*z + 336*a*b^7*c^5*d^2*g*
z - 672*a*b^6*c^6*d^2*e*z + 24576*a^8*c^5*h*j*k*z + 73728*a^7*c^6*d*j*k*z +
 32768*a^7*c^6*e*i*k*z - 12288*a^7*c^6*f*i*j*z + 8192*a^7*c^6*f*h*k*z + 245
76*a^6*c^7*d*f*k*z - 12288*a^6*c^7*e*f*j*z + 12288*a^6*c^7*d*h*i*z + 12288*
a^5*c^8*d*e*h*z + 2304*a^7*b^3*c^3*j^2*k*z - 576*a^6*b^5*c^2*j^2*k*z + 4505
6*a^7*b^3*c^3*i*k^2*z - 15360*a^6*b^5*c^2*i*k^2*z - 12288*a^7*b^2*c^4*i^2*k
*z + 3072*a^6*b^4*c^3*i^2*k*z - 256*a^5*b^6*c^2*i^2*k*z + 15872*a^7*b^2*c^4
*i*j^2*z + 6912*a^6*b^3*c^4*h^2*k*z - 4992*a^6*b^4*c^3*i*j^2*z - 1728*a^5*b
^5*c^3*h^2*k*z + 672*a^5*b^6*c^2*i*j^2*z + 144*a^4*b^7*c^2*h^2*k*z + 24576*
a^7*b^2*c^4*g*k^2*z - 22528*a^6*b^4*c^3*g*k^2*z + 7680*a^5*b^6*c^2*g*k^2*z
+ 4096*a^6*b^2*c^5*g^2*k*z - 3072*a^5*b^4*c^4*g^2*k*z + 768*a^4*b^6*c^3*g^2

*k*z - 64*a^3*b^8*c^2*g^2*k*z - 7936*a^6*b^3*c^4*g*j^2*z + 2496*a^5*b^5*c^3
*g*j^2*z - 1536*a^6*b^2*c^5*h^2*i*z + 1280*a^5*b^3*c^5*f^2*k*z + 384*a^5*b^
4*c^4*h^2*i*z - 336*a^4*b^7*c^2*g*j^2*z + 192*a^4*b^5*c^4*f^2*k*z - 144*a^3
*b^7*c^3*f^2*k*z - 32*a^4*b^6*c^3*h^2*i*z + 16*a^2*b^9*c^2*f^2*k*z + 45056*
a^6*b^3*c^4*e*k^2*z - 15360*a^5*b^5*c^3*e*k^2*z - 12288*a^5*b^2*c^6*e^2*k*z
 + 3072*a^4*b^4*c^5*e^2*k*z + 2304*a^4*b^7*c^2*e*k^2*z - 256*a^3*b^6*c^4*e^
2*k*z + 59136*a^4*b^3*c^6*d^2*k*z - 23488*a^3*b^5*c^5*d^2*k*z + 15872*a^6*b
^2*c^5*e*j^2*z - 4992*a^5*b^4*c^4*e*j^2*z + 4560*a^2*b^7*c^4*d^2*k*z + 1536
*a^5*b^2*c^6*f^2*i*z + 768*a^5*b^3*c^5*g*h^2*z + 672*a^4*b^6*c^3*e*j^2*z -
384*a^4*b^4*c^5*f^2*i*z - 192*a^4*b^5*c^4*g*h^2*z - 32*a^3*b^8*c^2*e*j^2*z
+ 32*a^3*b^6*c^4*f^2*i*z + 16*a^3*b^7*c^3*g*h^2*z - 15872*a^4*b^2*c^7*d^2*i
*z + 4992*a^3*b^4*c^6*d^2*i*z - 1536*a^5*b^2*c^6*e*h^2*z - 768*a^4*b^3*c^6*
f^2*g*z - 672*a^2*b^6*c^5*d^2*i*z + 384*a^4*b^4*c^5*e*h^2*z + 192*a^3*b^5*c
^5*f^2*g*z - 32*a^3*b^6*c^4*e*h^2*z - 16*a^2*b^7*c^4*f^2*g*z + 7936*a^3*b^3
*c^7*d^2*g*z - 2496*a^2*b^5*c^6*d^2*g*z + 1536*a^4*b^2*c^7*e*f^2*z - 384*a^
3*b^4*c^6*e*f^2*z + 32*a^2*b^6*c^5*e*f^2*z - 15872*a^3*b^2*c^8*d^2*e*z + 49
92*a^2*b^4*c^7*d^2*e*z - 61440*a^8*b^2*c^3*k^3*z + 21504*a^7*b^4*c^2*k^3*z
+ 16384*a^8*c^5*i^2*k*z - 18432*a^8*c^5*i*j^2*z - 128*a^4*b^9*i*k^2*z + 204
8*a^7*c^6*h^2*i*z + 64*a^3*b^10*g*k^2*z + 16384*a^6*c^7*e^2*k*z + 16*b^11*c
^2*d^2*k*z - 18432*a^7*c^6*e*j^2*z - 2048*a^6*c^7*f^2*i*z + 18432*a^5*c^8*d
^2*i*z - 3328*a^6*b^6*c*k^3*z + 2048*a^6*c^7*e*h^2*z - 16*b^9*c^4*d^2*g*z -
 2048*a^5*c^8*e*f^2*z + 32*b^8*c^5*d^2*e*z + 18432*a^4*c^9*d^2*e*z + 65536*
a^9*c^4*k^3*z + 192*a^5*b^8*k^3*z - 3328*a^7*b*c^3*h*i*j*k - 6912*a^6*b*c^4
*d*i*j*k - 3328*a^6*b*c^4*e*h*j*k - 1536*a^6*b*c^4*f*g*j*k - 768*a^6*b*c^4*
g*h*i*j - 768*a^6*b*c^4*f*h*i*k - 6912*a^5*b*c^5*d*e*j*k - 2304*a^5*b*c^5*d
*g*i*j - 1792*a^5*b*c^5*e*f*i*j + 1536*a^5*b*c^5*d*g*h*k - 1280*a^5*b*c^5*d
*f*i*k - 768*a^5*b*c^5*e*g*h*j - 768*a^5*b*c^5*e*f*h*k - 256*a^5*b*c^5*f*g*
h*i + 16*a*b^8*c^2*d*f*g*k - 4*a*b^8*c^2*d*f*h*j - 2304*a^4*b*c^6*d*e*g*j -
 1792*a^4*b*c^6*d*e*h*i - 1280*a^4*b*c^6*d*e*f*k - 768*a^4*b*c^6*d*f*g*i -
256*a^4*b*c^6*e*f*g*h - 32*a*b^7*c^3*d*e*f*k - 768*a^3*b*c^7*d*e*f*g + 32*a
*b^5*c^5*d*e*f*g + 576*a^6*b^3*c^2*h*i*j*k + 1664*a^6*b^2*c^3*g*h*j*k + 384
*a^6*b^2*c^3*f*i*j*k - 288*a^5*b^4*c^2*g*h*j*k - 160*a^5*b^4*c^2*f*i*j*k +
2112*a^5*b^3*c^3*d*i*j*k + 576*a^5*b^3*c^3*e*h*j*k - 448*a^5*b^3*c^3*f*h*i*
k - 192*a^5*b^3*c^3*g*h*i*j - 192*a^5*b^3*c^3*f*g*j*k - 160*a^4*b^5*c^2*d*i
*j*k + 96*a^4*b^5*c^2*f*h*i*k + 80*a^4*b^5*c^2*f*g*j*k + 32*a^4*b^5*c^2*g*h
*i*j + 4992*a^5*b^2*c^4*d*h*i*k - 4608*a^5*b^2*c^4*e*g*i*k + 3456*a^5*b^2*c
^4*d*g*j*k - 1312*a^4*b^4*c^3*d*h*i*k - 1056*a^4*b^4*c^3*d*g*j*k + 896*a^5*
b^2*c^4*f*g*i*j + 768*a^4*b^4*c^3*e*g*i*k + 384*a^5*b^2*c^4*f*g*h*k + 384*a
^5*b^2*c^4*e*h*i*j + 384*a^5*b^2*c^4*e*f*j*k + 224*a^4*b^4*c^3*f*g*h*k - 16
0*a^4*b^4*c^3*e*f*j*k - 96*a^4*b^4*c^3*f*g*i*j + 96*a^3*b^6*c^2*d*h*i*k + 8
0*a^3*b^6*c^2*d*g*j*k - 64*a^4*b^4*c^3*e*h*i*j - 48*a^3*b^6*c^2*f*g*h*k - 2
496*a^4*b^3*c^4*d*g*h*k + 2112*a^4*b^3*c^4*d*e*j*k - 960*a^4*b^3*c^4*d*f*i*
k + 656*a^3*b^5*c^3*d*g*h*k - 448*a^4*b^3*c^4*e*f*h*k + 384*a^3*b^5*c^3*d*f
*i*k + 320*a^4*b^3*c^4*d*g*i*j - 192*a^4*b^3*c^4*f*g*h*i - 192*a^4*b^3*c^4*
e*g*h*j + 192*a^4*b^3*c^4*e*f*i*j - 160*a^3*b^5*c^3*d*e*j*k + 96*a^3*b^5*c^

3*e*f*h*k - 48*a^2*b^7*c^2*d*g*h*k + 32*a^3*b^5*c^3*e*g*h*j - 32*a^2*b^7*c^2*d*f*i*k + 4992*a^4*b^2*c^5*d*e*h*k - 3584*a^4*b^2*c^5*d*f*h*j - 1312*a^3*b^4*c^4*d*e*h*k + 896*a^4*b^2*c^5*e*f*g*j + 896*a^4*b^2*c^5*d*g*h*i + 640*a^4*b^2*c^5*d*f*g*k - 640*a^4*b^2*c^5*d*e*i*j + 600*a^3*b^4*c^4*d*f*h*j + 480*a^3*b^4*c^4*d*f*g*k + 384*a^4*b^2*c^5*e*f*h*i - 192*a^2*b^6*c^3*d*f*g*k - 96*a^3*b^4*c^4*e*f*g*j - 96*a^3*b^4*c^4*d*g*h*i + 96*a^2*b^6*c^3*d*e*h*k + 12*a^2*b^6*c^3*d*f*h*j - 960*a^3*b^3*c^5*d*e*f*k + 384*a^2*b^5*c^4*d*e*f*k + 320*a^3*b^3*c^5*d*e*g*j - 192*a^3*b^3*c^5*e*f*g*h - 192*a^3*b^3*c^5*d*f*g*i + 192*a^3*b^3*c^5*d*e*h*i + 32*a^2*b^5*c^4*d*f*g*i + 896*a^3*b^2*c^6*d*e*g*h + 384*a^3*b^2*c^6*d*e*f*i - 96*a^2*b^4*c^5*d*e*g*h - 64*a^2*b^4*c^5*d*e*f*i - 192*a^2*b^3*c^6*d*e*f*g + 48*a^6*b^4*c*i*j^2*k - 1424*a^6*b^4*c*h*j*k^2 - 2304*a^7*b*c^3*g*j^2*k - 24*a^5*b^5*c*g*j^2*k + 2048*a^7*b*c^3*g*i*k^2 - 1024*a^7*b*c^3*f*j*k^2 - 768*a^5*b^5*c*g*i*k^2 + 408*a^5*b^5*c*f*j*k^2 + 256*a^6*b*c^4*g*h^2*k + 16*a^4*b^6*c*g*i*j^2 + 4608*a^6*b*c^4*e*i^2*k + 4608*a^5*b*c^5*e^2*i*k - 896*a^6*b*c^4*f*i^2*j + 768*a^4*b^6*c*d*j*k^2 - 256*a^4*b^6*c*f*h*k^2 - 128*a^4*b^6*c*e*i*k^2 + 2208*a^6*b*c^4*f*h*j^2 - 1920*a^6*b*c^4*e*i*j^2 + 800*a^5*b*c^5*f^2*h*j - 256*a^5*b*c^5*f^2*g*k - 16*a*b^8*c^2*d^2*i*k + 6*a^3*b^7*c*f*h*j^2 + 8192*a^6*b*c^4*d*h*k^2 + 2048*a^6*b*c^4*e*g*k^2 - 472*a^3*b^7*c*d*h*k^2 + 64*a^3*b^7*c*e*g*k^2 + 4896*a^4*b*c^6*d^2*h*j + 2304*a^4*b*c^6*d^2*g*k + 1824*a^5*b*c^5*d*h^2*j - 384*a^5*b*c^5*e*h^2*i - 168*a*b^7*c^3*d^2*g*k + 42*a*b^7*c^3*d^2*h*j + 6*a^2*b^8*c*d*h*j^2 + 1536*a^5*b*c^5*e*g*i^2 + 1536*a^4*b*c^6*e^2*g*i - 896*a^5*b*c^5*d*h*i^2 - 896*a^4*b*c^6*e^2*f*j + 144*a^2*b^8*c*d*f*k^2 + 4896*a^5*b*c^5*d*f*j^2 + 1824*a^4*b*c^6*d*f^2*j - 384*a^4*b*c^6*e*f^2*i + 336*a*b^6*c^4*d^2*e*k - 156*a*b^6*c^4*d^2*f*j + 16*a*b^6*c^4*d^2*g*i + 12*a*b^7*c^3*d*f^2*j + 2208*a^3*b*c^7*d^2*f*h - 1920*a^3*b*c^7*d^2*e*i + 800*a^4*b*c^6*d*f*h^2 - 102*a*b^5*c^5*d^2*f*h - 32*a*b^5*c^5*d^2*e*i + 12*a*b^6*c^4*d*f^2*h - 2*a*b^7*c^3*d*f*h^2 - 896*a^3*b*c^7*d*e^2*h - 8*a*b^6*c^4*d*f*g^2 - 240*a*b^4*c^6*d^2*e*g - 32*a*b^4*c^6*d*e^2*f + 3072*a^7*c^4*f*i*j*k + 3072*a^6*c^5*e*f*j*k - 3072*a^6*c^5*d*h*i*k + 1536*a^6*c^5*e*h*i*j + 4608*a^5*c^6*d*e*i*j - 3072*a^5*c^6*d*e*h*k - 1152*a^5*c^6*d*f*h*j + 512*a^5*c^6*e*f*h*i + 1536*a^4*c^7*d*e*f*i - 2*a*b^9*c*d*f*j^2 - 1088*a^7*b^2*c^2*i*j^2*k + 4800*a^7*b^2*c^2*h*j*k^2 + 960*a^6*b^2*c^3*h^2*i*k + 544*a^6*b^3*c^2*g*j^2*k - 144*a^5*b^4*c^2*h^2*i*k - 2304*a^6*b^2*c^3*g*i^2*k + 1920*a^6*b^3*c^2*g*i*k^2 + 1152*a^5*b^3*c^3*g^2*i*k - 864*a^6*b^3*c^2*f*j*k^2 + 384*a^5*b^4*c^2*g*i^2*k + 192*a^6*b^2*c^3*h*i^2*j - 192*a^4*b^5*c^2*g^2*i*k - 32*a^5*b^4*c^2*h*i^2*j - 1088*a^6*b^2*c^3*e*j^2*k + 960*a^6*b^2*c^3*g*i*j^2 - 480*a^5*b^3*c^3*g*h^2*k - 240*a^5*b^4*c^2*g*i*j^2 + 192*a^5*b^2*c^4*f^2*i*k + 72*a^4*b^5*c^2*g*h^2*k + 48*a^5*b^4*c^2*e*j^2*k + 48*a^4*b^4*c^3*f^2*i*k - 16*a^3*b^6*c^2*f^2*i*k + 13376*a^6*b^2*c^3*d*j*k^2 - 5136*a^5*b^4*c^2*d*j*k^2 - 3840*a^6*b^2*c^3*e*i*k^2 + 1536*a^5*b^4*c^2*e*i*k^2 - 768*a^5*b^3*c^3*e*i^2*k - 768*a^4*b^3*c^4*e^2*i*k + 624*a^5*b^4*c^2*f*h*k^2 + 576*a^6*b^2*c^3*f*h*k^2 + 192*a^5*b^2*c^4*g^2*h*j + 96*a^5*b^3*c^3*f*i^2*j + 48*a^4*b^4*c^3*g^2*h*j - 8*a^3*b^6*c^2*g^2*h*j + 6848*a^4*b^2*c^5*d^2*i*k - 2448*a^3*b^4*c^4*d^2*i*k + 960*a^5*b^2*c^4*e*h^2*k - 864*a^5*b^2*c^4*f*h^2*j + 480*a^5*b^3*c^3*e*i*j^2 + 33

6*a^4*b^3*c^4*f^2*h*j + 336*a^2*b^6*c^3*d^2*i*k + 192*a^5*b^2*c^4*g*h^2*i +
 144*a^5*b^3*c^3*f*h*j^2 - 144*a^4*b^4*c^3*e*h^2*k - 102*a^4*b^5*c^2*f*h*j^
2 - 96*a^4*b^3*c^4*f^2*g*k - 32*a^4*b^5*c^2*e*i*j^2 - 30*a^3*b^5*c^3*f^2*h*
j - 24*a^3*b^5*c^3*f^2*g*k + 16*a^4*b^4*c^3*g*h^2*i - 12*a^4*b^4*c^3*f*h^2*
j + 12*a^3*b^6*c^2*f*h^2*j + 8*a^2*b^7*c^2*f^2*g*k - 2*a^2*b^7*c^2*f^2*h*j
 - 9312*a^5*b^3*c^3*d*h*k^2 + 3288*a^4*b^5*c^2*d*h*k^2 - 2304*a^4*b^2*c^5*e^
2*g*k + 1920*a^5*b^3*c^3*e*g*k^2 + 1152*a^4*b^3*c^4*e*g^2*k - 768*a^4*b^5*c
^2*e*g*k^2 + 384*a^3*b^4*c^4*e^2*g*k - 320*a^5*b^2*c^4*d*i^2*j - 224*a^4*b^
3*c^4*f*g^2*j + 192*a^5*b^2*c^4*f*h*i^2 + 192*a^4*b^2*c^5*e^2*h*j - 192*a^3
*b^5*c^3*e*g^2*k - 32*a^3*b^4*c^4*e^2*h*j + 24*a^3*b^5*c^3*f*g^2*j - 3552*a
^5*b^2*c^4*d*h*j^2 - 3424*a^3*b^3*c^5*d^2*g*k + 1332*a^4*b^4*c^3*d*h*j^2 +
1224*a^2*b^5*c^4*d^2*g*k + 960*a^5*b^2*c^4*e*g*j^2 - 496*a^3*b^3*c^5*d^2*h*
j + 432*a^4*b^3*c^4*d*h^2*j - 240*a^4*b^4*c^3*e*g*j^2 - 222*a^2*b^5*c^4*d^2
*h*j + 192*a^4*b^2*c^5*f^2*g*i + 192*a^4*b^2*c^5*e*f^2*k - 174*a^3*b^5*c^3*
d*h^2*j - 156*a^3*b^6*c^2*d*h*j^2 + 48*a^3*b^4*c^4*e*f^2*k - 32*a^4*b^3*c^4
*e*h^2*i + 16*a^3*b^6*c^2*e*g*j^2 + 16*a^3*b^4*c^4*f^2*g*i - 16*a^2*b^6*c^3
*e*f^2*k + 12*a^2*b^7*c^2*d*h^2*j + 1728*a^5*b^2*c^4*d*f*k^2 + 1392*a^4*b^4
*c^3*d*f*k^2 - 840*a^3*b^6*c^2*d*f*k^2 - 768*a^4*b^2*c^5*e*g^2*i + 576*a^4*
b^2*c^5*d*g^2*j + 96*a^4*b^3*c^4*d*h*i^2 + 96*a^3*b^3*c^5*e^2*f*j - 80*a^3*
b^4*c^4*d*g^2*j + 64*a^4*b^2*c^5*f*g^2*h + 48*a^3*b^4*c^4*f*g^2*h + 6848*a^
3*b^2*c^6*d^2*e*k - 3552*a^3*b^2*c^6*d^2*f*j - 2448*a^2*b^4*c^5*d^2*e*k + 1
332*a^2*b^4*c^5*d^2*f*j + 960*a^3*b^2*c^6*d^2*g*i - 496*a^4*b^3*c^4*d*f*j^2
 + 432*a^3*b^3*c^5*d*f^2*j - 240*a^2*b^4*c^5*d^2*g*i - 222*a^3*b^5*c^3*d*f*
j^2 + 192*a^4*b^2*c^5*e*g*h^2 - 174*a^2*b^5*c^4*d*f^2*j + 42*a^2*b^7*c^2*d*
f*j^2 - 32*a^3*b^3*c^5*e*f^2*i + 16*a^3*b^4*c^4*e*g*h^2 - 320*a^3*b^2*c^6*d
*e^2*j - 224*a^3*b^3*c^5*d*g^2*h + 192*a^4*b^2*c^5*d*f*i^2 + 192*a^3*b^2*c^
6*e^2*f*h - 32*a^3*b^4*c^4*d*f*i^2 + 24*a^2*b^5*c^4*d*g^2*h - 864*a^3*b^2*c
^6*d*f^2*h + 480*a^2*b^3*c^6*d^2*e*i + 336*a^3*b^3*c^5*d*f*h^2 + 192*a^3*b^
2*c^6*e*f^2*g + 144*a^2*b^3*c^6*d^2*f*h - 30*a^2*b^5*c^4*d*f*h^2 + 16*a^2*b
^4*c^5*e*f^2*g - 12*a^2*b^4*c^5*d*f^2*h + 192*a^3*b^2*c^6*d*f*g^2 + 96*a^2*
b^3*c^6*d*e^2*h + 48*a^2*b^4*c^5*d*f*g^2 + 960*a^2*b^2*c^7*d^2*e*g + 192*a^
2*b^2*c^7*d*e^2*f - 3072*a^8*b*c^2*j^2*k^2 + 1104*a^7*b^3*c*j^2*k^2 + 768*a
^6*b^4*c*i^2*k^2 - 256*a^6*b^3*c^2*i^3*k + 1536*a^7*b*c^3*h^2*k^2 - 960*a^7
*b*c^3*i^2*j^2 + 444*a^5*b^5*c*h^2*k^2 - 16*a^5*b^5*c*i^2*j^2 - 3072*a^7*b^
2*c^2*g*k^3 - 496*a^6*b^3*c^2*h*j^3 + 192*a^4*b^6*c*g^2*k^2 - 192*a^4*b^4*c
^3*g^3*k + 144*a^5*b^3*c^3*h^3*j + 32*a^3*b^6*c^2*g^3*k - 18*a^4*b^5*c^2*h^
3*j - 9*a^4*b^6*c*h^2*j^2 - 192*a^6*b*c^4*h^2*i^2 + 36*a^3*b^7*c*f^2*k^2 -
4*a^3*b^7*c*g^2*j^2 - 2176*a^6*b^3*c^2*e*k^3 - 256*a^3*b^3*c^5*e^3*k - 192*
a^6*b^2*c^3*f*j^3 - 192*a^4*b^2*c^5*f^3*j + 132*a^5*b^4*c^2*f*j^3 + 128*a^4
*b^3*c^4*g^3*i - 28*a^3*b^4*c^4*f^3*j + 6*a^2*b^6*c^3*f^3*j + 10752*a^5*b*c
^5*d^2*k^2 - 960*a^5*b*c^5*e^2*j^2 - 192*a^5*b*c^5*f^2*i^2 - 1680*a^5*b^3*c
^3*d*j^3 - 1680*a^2*b^3*c^6*d^3*j + 222*a^4*b^5*c^2*d*j^3 + 80*a^4*b^3*c^4*
f*h^3 + 80*a^3*b^3*c^5*f^3*h + 30*a*b^8*c^2*d^2*j^2 + 6*a^3*b^5*c^3*f*h^3 +
 6*a^2*b^5*c^4*f^3*h - 960*a^4*b*c^6*d^2*i^2 - 192*a^4*b*c^6*e^2*h^2 - 192*
a^4*b^2*c^5*d*h^3 - 192*a^2*b^2*c^7*d^3*h + 128*a^3*b^3*c^5*e*g^3 - 28*a^3*

b^4*c^4*d*h^3 + 12*a*b^6*c^4*d^2*h^2 + 6*a^2*b^6*c^3*d*h^3 - 192*a^3*b*c^7*
e^2*f^2 + 60*a*b^5*c^5*d^2*g^2 + 198*a*b^4*c^6*d^2*f^2 + 144*a^2*b^3*c^6*d*
f^3 - 960*a^2*b*c^8*d^2*e^2 + 240*a*b^3*c^7*d^2*e^2 + 4608*a^8*c^3*i*j^2*k
- 3072*a^8*c^3*h*j*k^2 - 512*a^7*c^4*h^2*i*k + 120*a^5*b^6*h*j*k^2 + 768*a^
7*c^4*h*i^2*j + 4608*a^7*c^4*e*j^2*k + 512*a^6*c^5*f^2*i*k + 64*a^4*b^7*g*i
*k^2 - 40*a^4*b^7*f*j*k^2 - 9216*a^7*c^4*d*j*k^2 - 4096*a^7*c^4*e*i*k^2 - 1
024*a^7*c^4*f*h*k^2 - 4608*a^5*c^6*d^2*i*k - 512*a^6*c^5*e*h^2*k - 192*a^6*
c^5*f*h^2*j - 40*a^3*b^8*d*j*k^2 + 24*a^3*b^8*f*h*k^2 + 2304*a^6*c^5*d*i^2*
j + 768*a^5*c^6*e^2*h*j + 256*a^6*c^5*f*h*i^2 + 8*b^9*c^2*d^2*g*k - 2*b^9*c
^2*d^2*h*j + 6144*a^8*b*c^2*i*k^3 - 2176*a^7*b^3*c*i*k^3 - 1728*a^6*c^5*d*h
*j^2 + 1536*a^7*b*c^3*i^3*k + 512*a^5*c^6*e*f^2*k + 24*a^2*b^9*d*h*k^2 - 30
72*a^6*c^5*d*f*k^2 - 16*b^8*c^3*d^2*e*k + 6*b^8*c^3*d^2*f*j - 4608*a^4*c^7*
d^2*e*k + 2016*a^7*b*c^3*h*j^3 - 1728*a^4*c^7*d^2*f*j + 1088*a^6*b^4*c*g*k^
3 + 224*a^6*b*c^4*h^3*j + 30*a^5*b^5*c*h*j^3 + 2304*a^4*c^7*d*e^2*j + 768*a
^5*c^6*d*f*i^2 + 256*a^4*c^7*e^2*f*h + 6*b^7*c^4*d^2*f*h + 6144*a^7*b*c^3*e
*k^3 + 1536*a^4*b*c^6*e^3*k + 512*a^6*b*c^4*g*i^3 + 192*a^5*b^5*c*e*k^3 - 1
92*a^4*c^7*d*f^2*h - 10*a^4*b^6*c*f*j^3 + 108*a*b^9*c*d^2*k^2 + 16*b^6*c^5*
d^2*e*g + 4320*a^6*b*c^4*d*j^3 + 4320*a^3*b*c^7*d^3*j + 222*a*b^5*c^5*d^3*j
 + 96*a^5*b*c^5*f*h^3 + 96*a^4*b*c^6*f^3*h - 10*a^3*b^7*c*d*j^3 + 768*a^3*c
^8*d*e^2*f + 512*a^3*b*c^7*e^3*g + 132*a*b^4*c^6*d^3*h + 2016*a^2*b*c^8*d^3
*f - 496*a*b^3*c^7*d^3*f + 224*a^3*b*c^7*d*f^3 - 18*a*b^5*c^5*d*f^3 - 1920*
a^7*b^2*c^2*i^2*k^2 - 1648*a^6*b^3*c^2*h^2*k^2 + 240*a^6*b^3*c^2*i^2*j^2 -
960*a^6*b^2*c^3*h^2*j^2 - 512*a^6*b^2*c^3*g^2*k^2 - 480*a^5*b^4*c^2*g^2*k^2
 + 198*a^5*b^4*c^2*h^2*j^2 - 240*a^5*b^3*c^3*g^2*j^2 - 240*a^5*b^3*c^3*f^2*
k^2 + 60*a^4*b^5*c^2*g^2*j^2 - 36*a^4*b^5*c^2*f^2*k^2 - 16*a^5*b^3*c^3*h^2*
i^2 - 1920*a^5*b^2*c^4*e^2*k^2 + 768*a^4*b^4*c^3*e^2*k^2 - 464*a^5*b^2*c^4*
f^2*j^2 - 384*a^5*b^2*c^4*g^2*i^2 - 64*a^3*b^6*c^2*e^2*k^2 + 42*a^4*b^4*c^3
*f^2*j^2 + 12*a^3*b^6*c^2*f^2*j^2 - 13104*a^4*b^3*c^4*d^2*k^2 + 5628*a^3*b^
5*c^3*d^2*k^2 - 1128*a^2*b^7*c^2*d^2*k^2 + 240*a^4*b^3*c^4*e^2*j^2 - 48*a^4
*b^3*c^4*g^2*h^2 - 16*a^4*b^3*c^4*f^2*i^2 - 16*a^3*b^5*c^3*e^2*j^2 - 4*a^3*
b^5*c^3*g^2*h^2 - 2880*a^4*b^2*c^5*d^2*j^2 + 1750*a^3*b^4*c^4*d^2*j^2 - 345
*a^2*b^6*c^3*d^2*j^2 - 192*a^4*b^2*c^5*f^2*h^2 - 42*a^3*b^4*c^4*f^2*h^2 + 2
40*a^3*b^3*c^5*d^2*i^2 - 48*a^3*b^3*c^5*f^2*g^2 - 16*a^3*b^3*c^5*e^2*h^2 -
16*a^2*b^5*c^4*d^2*i^2 - 4*a^2*b^5*c^4*f^2*g^2 - 464*a^3*b^2*c^6*d^2*h^2 -
384*a^3*b^2*c^6*e^2*g^2 + 42*a^2*b^4*c^5*d^2*h^2 - 240*a^2*b^3*c^6*d^2*g^2
- 16*a^2*b^3*c^6*e^2*f^2 - 960*a^2*b^2*c^7*d^2*f^2 - 8*a*b^10*d*f*k^2 - a^2
*b^8*c*f^2*j^2 - 2048*a^8*c^3*i^2*k^2 - 100*a^6*b^5*j^2*k^2 - 64*a^5*b^6*i^
2*k^2 - 288*a^7*c^4*h^2*j^2 - 36*a^4*b^7*h^2*k^2 - 16*a^3*b^8*g^2*k^2 - 204
8*a^6*c^5*e^2*k^2 - 864*a^6*c^5*f^2*j^2 - 4*a^2*b^9*f^2*k^2 - 2592*a^5*c^6*
d^2*j^2 - 1536*a^5*c^6*e^2*i^2 - 32*a^5*c^6*f^2*h^2 - 864*a^4*c^7*d^2*h^2 +
 360*a^7*b^2*c^2*j^4 - 4*b^7*c^4*d^2*g^2 - 9*b^6*c^5*d^2*f^2 - 288*a^3*c^8*
d^2*f^2 - 24*a^5*b^2*c^4*h^4 - 16*b^5*c^6*d^2*e^2 - 9*a^4*b^4*c^3*h^4 - 16*
a^3*b^4*c^4*g^4 - 24*a^3*b^2*c^6*f^4 - 9*a^2*b^4*c^5*f^4 - a^2*b^6*c^3*f^2*
h^2 + 192*a^6*b^5*i*k^3 - 96*a^5*b^6*g*k^3 - 1728*a^7*c^4*f*j^3 - 192*a^5*c
^6*f^3*j - 10*b^7*c^4*d^3*j - 1024*a^6*c^5*e*i^3 - 1024*a^4*c^7*e^3*i + 153

6*a^8*b^2*c*k^4 - 10*b^6*c^5*d^3*h - 1728*a^3*c^8*d^3*h - 192*a^5*c^6*d*h^3
 - 25*a^6*b^4*c*j^4 + 30*b^5*c^6*d^3*f + 360*a*b^2*c^8*d^4 - 4*b^11*d^2*k^2
 - 4096*a^9*c^2*k^4 - 1296*a^8*c^3*j^4 - 144*a^7*b^4*k^4 - 256*a^7*c^4*i^4
- 16*a^6*c^5*h^4 - 16*a^4*c^7*f^4 - 256*a^3*c^8*e^4 - 25*b^4*c^7*d^4 - 1296
*a^2*c^9*d^4 - b^8*c^3*d^2*h^2 - b^10*c*d^2*j^2, z, n)*x*(8192*a^6*b*c^8 +
32*a^2*b^9*c^4 - 512*a^3*b^7*c^5 + 3072*a^4*b^5*c^6 - 8192*a^5*b^3*c^7))/(4
*(64*a^5*c^5 - a^2*b^6*c^2 + 12*a^3*b^4*c^3 - 48*a^4*b^2*c^4))) + (x*(2*b^6
*c^5*d^2 - 576*a^3*c^8*d^2 + 64*a^4*c^7*f^2 - 64*a^5*c^6*h^2 + 8*a^2*b^9*k^
2 + 576*a^6*c^5*j^2 - 36*a*b^4*c^6*d^2 + 128*a^3*b*c^7*e^2 + 128*a^5*b*c^5*
i^2 + 2*a^2*b^8*c*j^2 - 136*a^3*b^7*c*k^2 + 3072*a^6*b*c^4*k^2 + 256*a^2*b^
2*c^7*d^2 - 32*a^2*b^3*c^6*e^2 + 20*a^2*b^4*c^5*f^2 - 96*a^3*b^2*c^6*f^2 -
8*a^2*b^5*c^4*g^2 + 32*a^3*b^3*c^5*g^2 + 2*a^2*b^6*c^3*h^2 - 4*a^3*b^4*c^4*
h^2 - 32*a^4*b^3*c^4*i^2 - 40*a^3*b^6*c^2*j^2 + 276*a^4*b^4*c^3*j^2 - 736*a
^5*b^2*c^4*j^2 + 888*a^4*b^5*c^2*k^2 - 2656*a^5*b^3*c^3*k^2 - 384*a^4*c^7*d
*h - 1024*a^5*c^6*e*k + 384*a^5*c^6*f*j - 1024*a^6*c^5*i*k + 4*a*b^5*c^5*d*
f + 320*a^3*b*c^7*d*f + 576*a^4*b*c^6*d*j + 256*a^4*b*c^6*e*i + 64*a^4*b*c^
6*f*h + 512*a^5*b*c^5*g*k + 64*a^5*b*c^5*h*j - 96*a^2*b^3*c^6*d*f + 8*a^2*b
^4*c^5*d*h + 32*a^2*b^4*c^5*e*g + 64*a^3*b^2*c^6*d*h - 128*a^3*b^2*c^6*e*g
+ 20*a^2*b^5*c^4*d*j - 12*a^2*b^5*c^4*f*h - 224*a^3*b^3*c^5*d*j - 64*a^3*b^
3*c^5*e*i + 32*a^3*b^3*c^5*f*h - 12*a^2*b^6*c^3*f*j - 32*a^3*b^4*c^4*e*k +
152*a^3*b^4*c^4*f*j + 32*a^3*b^4*c^4*g*i + 384*a^4*b^2*c^5*e*k - 512*a^4*b^
2*c^5*f*j - 128*a^4*b^2*c^5*g*i + 4*a^2*b^7*c^2*h*j + 16*a^3*b^5*c^3*g*k -
44*a^3*b^5*c^3*h*j - 192*a^4*b^3*c^4*g*k + 96*a^4*b^3*c^4*h*j - 32*a^4*b^4*
c^3*i*k + 384*a^5*b^2*c^4*i*k))/(4*(64*a^5*c^5 - a^2*b^6*c^2 + 12*a^3*b^4*c
^3 - 48*a^4*b^2*c^4))) - (5*b^3*c^6*d^3 + 8*a^3*c^6*f^3 + 216*a^6*c^3*j^3 -
 96*a^2*c^7*d*e^2 + 72*a^2*c^7*d^2*f - 4*a^4*b*c^4*h^3 - 3*b^4*c^5*d^2*f +
5*a^4*b^4*c*j^3 - 32*a^3*c^6*e^2*h - 96*a^4*c^5*d*i^2 + b^5*c^4*d^2*h + 216
*a^3*c^6*d^2*j + 8*a^4*c^5*f*h^2 + 384*a^5*c^4*d*k^2 + b^6*c^3*d^2*j + 4*a^
2*b^7*f*k^2 + 72*a^4*c^5*f^2*j + 216*a^5*c^4*f*j^2 - 32*a^5*c^4*h*i^2 - 12*
a^3*b^6*h*k^2 + 24*a^5*c^4*h^2*j + 128*a^6*c^3*h*k^2 + 20*a^4*b^5*j*k^2 + 6
*a^2*b^2*c^5*f^3 - 3*a^3*b^3*c^3*h^3 - 66*a^5*b^2*c^2*j^3 - 36*a*b*c^7*d^3
+ 4*a*b^8*d*k^2 + a*b^7*c*d*j^2 - 192*a^3*c^6*d*e*i + 48*a^3*c^6*d*f*h + 14
4*a^4*c^5*d*h*j - 128*a^4*c^5*e*f*k - 64*a^4*c^5*e*h*i - 384*a^5*c^4*e*j*k
- 128*a^5*c^4*f*i*k - 384*a^6*c^3*i*j*k + 16*a*b^2*c^6*d*e^2 + 18*a*b^2*c^6
*d^2*f + 3*a*b^3*c^5*d*f^2 - 60*a^2*b*c^6*d*f^2 + 4*a*b^4*c^4*d*g^2 + 16*a^
2*b*c^6*e^2*f - a*b^3*c^5*d^2*h + a*b^5*c^3*d*h^2 - 60*a^2*b*c^6*d^2*h - 28
*a^3*b*c^5*d*h^2 - 10*a*b^4*c^4*d^2*j - 28*a^3*b*c^5*f^2*h - 396*a^4*b*c^4*
d*j^2 - 72*a^2*b^6*c*d*k^2 + 16*a^3*b*c^5*e^2*j + 16*a^4*b*c^4*f*i^2 + a^2*
b^6*c*f*j^2 - 36*a^3*b^5*c*f*k^2 + 128*a^5*b*c^3*f*k^2 - 3*a^3*b^5*c*h*j^2
- 204*a^5*b*c^3*h*j^2 + 128*a^4*b^4*c*h*k^2 + 16*a^5*b*c^3*i^2*j - 204*a^5*
b^3*c*j*k^2 + 512*a^6*b*c^2*j*k^2 - 24*a^2*b^2*c^5*d*g^2 - 9*a^2*b^3*c^4*d*
h^2 + 4*a^2*b^3*c^4*f*g^2 + 16*a^3*b^2*c^4*d*i^2 - 6*a^2*b^2*c^5*d^2*j - 5*
a^2*b^3*c^4*f^2*h + a^2*b^4*c^3*f*h^2 - 21*a^2*b^5*c^2*d*j^2 + 18*a^3*b^2*c
^4*f*h^2 + 155*a^3*b^3*c^3*d*j^2 - 8*a^3*b^2*c^4*g^2*h + 436*a^3*b^4*c^2*d*
k^2 - 952*a^4*b^2*c^3*d*k^2 - 5*a^2*b^4*c^3*f^2*j + 26*a^3*b^2*c^4*f^2*j -

12*a^3*b^4*c^2*f*j^2 + 2*a^4*b^2*c^3*f*j^2 + 4*a^3*b^3*c^3*g^2*j + 52*a^4*b
^3*c^2*f*k^2 - 6*a^3*b^4*c^2*h^2*j + 42*a^4*b^2*c^3*h^2*j + 51*a^4*b^3*c^2*
h*j^2 - 360*a^5*b^2*c^2*h*k^2 - 16*a*b^3*c^5*d*e*g + 96*a^2*b*c^6*d*e*g - 4
*a*b^4*c^4*d*f*h + 16*a*b^5*c^3*d*e*k - 4*a*b^5*c^3*d*f*j + 544*a^3*b*c^5*d
*e*k - 312*a^3*b*c^5*d*f*j + 96*a^3*b*c^5*d*g*i + 32*a^3*b*c^5*e*f*i + 32*a
^3*b*c^5*e*g*h - 8*a*b^6*c^2*d*g*k + 2*a*b^6*c^2*d*h*j + 544*a^4*b*c^4*d*i*
k + 224*a^4*b*c^4*e*h*k + 32*a^4*b*c^4*e*i*j + 64*a^4*b*c^4*f*g*k - 152*a^4
*b*c^4*f*h*j + 32*a^4*b*c^4*g*h*i + 192*a^5*b*c^3*g*j*k + 224*a^5*b*c^3*h*i
*k + 32*a^2*b^2*c^5*d*e*i + 52*a^2*b^2*c^5*d*f*h - 16*a^2*b^2*c^5*e*f*g - 1
92*a^2*b^3*c^4*d*e*k + 70*a^2*b^3*c^4*d*f*j - 16*a^2*b^3*c^4*d*g*i + 96*a^2
*b^4*c^3*d*g*k - 30*a^2*b^4*c^3*d*h*j + 16*a^2*b^4*c^3*e*f*k - 272*a^3*b^2*
c^4*d*g*k + 100*a^3*b^2*c^4*d*h*j - 48*a^3*b^2*c^4*e*f*k - 16*a^3*b^2*c^4*e
*g*j - 16*a^3*b^2*c^4*f*g*i + 16*a^2*b^5*c^2*d*i*k - 8*a^2*b^5*c^2*f*g*k +
2*a^2*b^5*c^2*f*h*j - 192*a^3*b^3*c^3*d*i*k - 48*a^3*b^3*c^3*e*h*k + 24*a^3
*b^3*c^3*f*g*k + 6*a^3*b^3*c^3*f*h*j + 16*a^3*b^4*c^2*f*i*k + 24*a^3*b^4*c^
2*g*h*k + 80*a^4*b^2*c^3*e*j*k - 48*a^4*b^2*c^3*f*i*k - 112*a^4*b^2*c^3*g*h
*k - 16*a^4*b^2*c^3*g*i*j - 40*a^4*b^3*c^2*g*j*k - 48*a^4*b^3*c^2*h*i*k + 8
0*a^5*b^2*c^2*i*j*k)/(8*(64*a^5*c^5 - a^2*b^6*c^2 + 12*a^3*b^4*c^3 - 48*a^4
*b^2*c^4)) + (x*(32*a^2*c^7*e^3 + 32*a^5*c^4*i^3 - 12*a^4*b^5*k^3 - 2*b^3*c
^6*d^2*e + b^4*c^5*d^2*g + 124*a^5*b^3*c*k^3 - 320*a^6*b*c^2*k^3 + 96*a^3*c
^6*e^2*i + 96*a^4*c^5*e*i^2 + 144*a^3*c^6*d^2*k + 128*a^5*c^4*e*k^2 - b^6*c
^3*d^2*k - 4*a^2*b^7*g*k^2 - 16*a^4*c^5*f^2*k + 8*a^3*b^6*i*k^2 + 16*a^5*c^
4*h^2*k + 128*a^6*c^3*i*k^2 - 144*a^6*c^3*j^2*k - 4*a^2*b^3*c^4*g^3 + 24*a*
b*c^7*d^2*e - 48*a^2*c^7*d*e*f - 144*a^3*c^6*d*e*j - 48*a^3*c^6*d*f*i - 16*
a^3*c^6*e*f*h + 96*a^4*c^5*d*h*k - 144*a^4*c^5*d*i*j - 48*a^4*c^5*e*h*j - 1
6*a^4*c^5*f*h*i - 96*a^5*c^4*f*j*k - 48*a^5*c^4*h*i*j - 12*a*b^2*c^6*d^2*g
+ 16*a^2*b*c^6*e*f^2 - 48*a^2*b*c^6*e^2*g - 2*a*b^3*c^5*d^2*i + 24*a^2*b*c^
6*d^2*i + 8*a^3*b*c^5*e*h^2 + 18*a*b^4*c^4*d^2*k + 16*a^3*b*c^5*f^2*i + 96*
a^4*b*c^4*e*j^2 + 8*a^2*b^6*c*e*k^2 - 176*a^3*b*c^5*e^2*k - 48*a^4*b*c^4*g*
i^2 - a^2*b^6*c*g*j^2 + 8*a^4*b*c^4*h^2*i + 44*a^3*b^5*c*g*k^2 - 64*a^5*b*c
^3*g*k^2 + 2*a^3*b^5*c*i*j^2 + 96*a^5*b*c^3*i*j^2 - 88*a^4*b^4*c*i*k^2 - 17
6*a^5*b*c^3*i^2*k - 3*a^4*b^4*c*j^2*k + 24*a^2*b^2*c^5*e*g^2 - 8*a^2*b^2*c^
5*f^2*g + 2*a^2*b^3*c^4*e*h^2 - 100*a^2*b^2*c^5*d^2*k - a^2*b^4*c^3*g*h^2 +
 2*a^2*b^5*c^2*e*j^2 - 4*a^3*b^2*c^4*g*h^2 - 28*a^3*b^3*c^3*e*j^2 + 32*a^2*
b^3*c^4*e^2*k + 24*a^3*b^2*c^4*g^2*i - 88*a^3*b^4*c^2*e*k^2 + 216*a^4*b^2*c
^3*e*k^2 - a^2*b^4*c^3*f^2*k + 2*a^3*b^3*c^3*h^2*i + 14*a^3*b^4*c^2*g*j^2 -
 48*a^4*b^2*c^3*g*j^2 + 8*a^2*b^5*c^2*g^2*k - 44*a^3*b^3*c^3*g^2*k - 108*a^
4*b^3*c^2*g*k^2 - 12*a^4*b^2*c^3*h^2*k - 28*a^4*b^3*c^2*i*j^2 + 32*a^4*b^3*
c^2*i^2*k + 216*a^5*b^2*c^2*i*k^2 + 40*a^5*b^2*c^2*j^2*k - 4*a*b^2*c^6*d*e*
f + 2*a*b^3*c^5*d*f*g + 32*a^2*b*c^6*d*e*h + 24*a^2*b*c^6*d*f*g - 2*a*b^5*c
^3*d*f*k - 8*a^3*b*c^5*d*f*k + 72*a^3*b*c^5*d*g*j + 32*a^3*b*c^5*d*h*i + 80
*a^3*b*c^5*e*f*j - 96*a^3*b*c^5*e*g*i + 8*a^3*b*c^5*f*g*h + 72*a^4*b*c^4*d*
j*k - 352*a^4*b*c^4*e*i*k + 8*a^4*b*c^4*f*h*k + 80*a^4*b*c^4*f*i*j + 24*a^4
*b*c^4*g*h*j + 56*a^5*b*c^3*h*j*k + 20*a^2*b^2*c^5*d*e*j - 4*a^2*b^2*c^5*d*
f*i - 16*a^2*b^2*c^5*d*g*h - 12*a^2*b^2*c^5*e*f*h + 18*a^2*b^3*c^4*d*f*k -

10*a^2*b^3*c^4*d*g*j - 12*a^2*b^3*c^4*e*f*j + 6*a^2*b^3*c^4*f*g*h + 6*a^2*b^4*c^3*d*h*k - 32*a^2*b^4*c^3*e*g*k + 4*a^2*b^4*c^3*e*h*j + 6*a^2*b^4*c^3*f*g*j - 64*a^3*b^2*c^4*d*h*k + 20*a^3*b^2*c^4*d*i*j + 176*a^3*b^2*c^4*e*g*k - 20*a^3*b^2*c^4*e*h*j - 40*a^3*b^2*c^4*f*g*j - 12*a^3*b^2*c^4*f*h*i - 2*a^2*b^5*c^2*g*h*j - 10*a^3*b^3*c^3*d*j*k + 64*a^3*b^3*c^3*e*i*k + 6*a^3*b^3*c^3*f*h*k - 12*a^3*b^3*c^3*f*i*j + 10*a^3*b^3*c^3*g*h*j - 32*a^3*b^4*c^2*g*i*k + 4*a^3*b^4*c^2*h*i*j + 8*a^4*b^2*c^3*f*j*k + 176*a^4*b^2*c^3*g*i*k - 20*a^4*b^2*c^3*h*i*j - 6*a^4*b^3*c^2*h*j*k))/(4*(64*a^5*c^5 - a^2*b^6*c^2 + 12*a^3*b^4*c^3 - 48*a^4*b^2*c^4)))*root(1572864*a^8*b^2*c^9*z^4 - 983040*a^7*b^4*c^8*z^4 + 327680*a^6*b^6*c^7*z^4 - 61440*a^5*b^8*c^6*z^4 + 6144*a^4*b^10*c^5*z^4 - 256*a^3*b^12*c^4*z^4 - 1048576*a^9*c^10*z^4 - 1572864*a^8*b^2*c^7*k*z^3 + 983040*a^7*b^4*c^6*k*z^3 - 327680*a^6*b^6*c^5*k*z^3 + 61440*a^5*b^8*c^4*k*z^3 - 6144*a^4*b^10*c^3*k*z^3 + 256*a^3*b^12*c^2*k*z^3 + 1048576*a^9*c^8*k*z^3 + 98304*a^8*b*c^6*i*k*z^2 + 98304*a^7*b*c^7*e*k*z^2 + 57344*a^7*b*c^7*f*j*z^2 + 32768*a^7*b*c^7*g*i*z^2 + 57344*a^6*b*c^8*d*h*z^2 + 32768*a^6*b*c^8*e*g*z^2 - 32*a*b^10*c^4*d*f*z^2 - 90112*a^7*b^3*c^5*i*k*z^2 + 30720*a^6*b^5*c^4*i*k*z^2 - 4608*a^5*b^7*c^3*i*k*z^2 + 256*a^4*b^9*c^2*i*k*z^2 - 49152*a^7*b^2*c^6*g*k*z^2 + 45056*a^6*b^4*c^5*g*k*z^2 + 24576*a^7*b^2*c^6*h*j*z^2 - 15360*a^5*b^6*c^4*g*k*z^2 - 3072*a^5*b^6*c^4*h*j*z^2 + 2304*a^4*b^8*c^3*g*k*z^2 + 2048*a^6*b^4*c^5*h*j*z^2 + 576*a^4*b^8*c^3*h*j*z^2 - 128*a^3*b^10*c^2*g*k*z^2 - 32*a^3*b^10*c^2*h*j*z^2 - 90112*a^6*b^3*c^6*e*k*z^2 - 49152*a^6*b^3*c^6*f*j*z^2 + 30720*a^5*b^5*c^5*e*k*z^2 - 24576*a^6*b^3*c^6*g*i*z^2 + 15360*a^5*b^5*c^5*f*j*z^2 + 6144*a^5*b^5*c^5*g*i*z^2 - 4608*a^4*b^7*c^4*e*k*z^2 - 2048*a^4*b^7*c^4*f*j*z^2 - 512*a^4*b^7*c^4*g*i*z^2 + 256*a^3*b^9*c^3*e*k*z^2 + 96*a^3*b^9*c^3*f*j*z^2 + 131072*a^6*b^2*c^7*d*j*z^2 + 49152*a^6*b^2*c^7*e*i*z^2 - 43008*a^5*b^4*c^6*d*j*z^2 - 12288*a^5*b^4*c^6*e*i*z^2 + 6144*a^5*b^4*c^6*f*h*z^2 + 6144*a^4*b^6*c^5*d*j*z^2 - 2048*a^4*b^6*c^5*f*h*z^2 + 1024*a^4*b^6*c^5*e*i*z^2 - 320*a^3*b^8*c^4*d*j*z^2 + 192*a^3*b^8*c^4*f*h*z^2 - 49152*a^5*b^3*c^7*d*h*z^2 - 24576*a^5*b^3*c^7*e*g*z^2 + 15360*a^4*b^5*c^6*d*h*z^2 + 6144*a^4*b^5*c^6*e*g*z^2 - 2048*a^3*b^7*c^5*d*h*z^2 - 512*a^3*b^7*c^5*e*g*z^2 + 96*a^2*b^9*c^4*d*h*z^2 + 24576*a^5*b^2*c^8*d*f*z^2 - 3072*a^3*b^6*c^6*d*f*z^2 + 2048*a^4*b^4*c^7*d*f*z^2 + 576*a^2*b^8*c^5*d*f*z^2 + 1536*a^4*b^10*c*k^2*z^2 + 61440*a^8*b*c^6*j^2*z^2 - 16*a^3*b^11*c*j^2*z^2 + 12288*a^7*b*c^7*h^2*z^2 + 12288*a^6*b*c^8*f^2*z^2 + 61440*a^5*b*c^9*d^2*z^2 + 432*a*b^9*c^5*d^2*z^2 - 49152*a^8*c^7*h*j*z^2 - 147456*a^7*c^8*d*j*z^2 - 65536*a^7*c^8*e*i*z^2 - 16384*a^7*c^8*f*h*z^2 - 49152*a^6*c^9*d*f*z^2 + 516096*a^8*b^2*c^5*k^2*z^2 - 288768*a^7*b^4*c^4*k^2*z^2 + 88576*a^6*b^6*c^3*k^2*z^2 - 15744*a^5*b^8*c^2*k^2*z^2 - 61440*a^7*b^3*c^5*j^2*z^2 + 24064*a^6*b^5*c^4*j^2*z^2 - 4608*a^5*b^7*c^3*j^2*z^2 + 432*a^4*b^9*c^2*j^2*z^2 + 24576*a^7*b^2*c^6*i^2*z^2 - 6144*a^6*b^4*c^5*i^2*z^2 + 512*a^5*b^6*c^4*i^2*z^2 - 8192*a^6*b^3*c^6*h^2*z^2 + 1536*a^5*b^5*c^5*h^2*z^2 - 16*a^3*b^9*c^3*h^2*z^2 - 8192*a^6*b^2*c^7*g^2*z^2 + 6144*a^5*b^4*c^6*g^2*z^2 - 1536*a^4*b^6*c^5*g^2*z^2 + 128*a^3*b^8*c^4*g^2*z^2 - 8192*a^5*b^3*c^7*f^2*z^2 + 1536*a^4*b^5*c^6*f^2*z^2 - 16*a^2*b^9*c^4*f^2*z^2 + 24576*a^5*b^2*c^8*e^2*z^2 - 6144*a^4*b^4*c^7*e^2*z^2 + 512*a^3*b^6*c^6*e^2*z^2 - 61440

*a^4*b^3*c^8*d^2*z^2 + 24064*a^3*b^5*c^7*d^2*z^2 - 4608*a^2*b^7*c^6*d^2*z^2
 - 393216*a^9*c^6*k^2*z^2 - 64*a^3*b^12*k^2*z^2 - 32768*a^8*c^7*i^2*z^2 - 3
2768*a^6*c^9*e^2*z^2 - 16*b^11*c^4*d^2*z^2 - 16384*a^7*b*c^5*g*i*k*z - 1024
0*a^7*b*c^5*f*j*k*z + 4096*a^7*b*c^5*h*i*j*z - 47104*a^6*b*c^6*d*h*k*z - 16
384*a^6*b*c^6*e*g*k*z + 6144*a^6*b*c^6*f*g*j*z + 4096*a^6*b*c^6*e*h*j*z + 3
2*a*b^10*c^2*d*f*k*z - 6144*a^5*b*c^7*d*g*h*z - 4096*a^5*b*c^7*d*f*i*z - 32
*a*b^8*c^4*d*f*g*z - 4096*a^4*b*c^8*d*e*f*z + 64*a*b^7*c^5*d*e*f*z - 18432*
a^7*b^2*c^4*h*j*k*z + 4608*a^6*b^4*c^3*h*j*k*z - 384*a^5*b^6*c^2*h*j*k*z +
12288*a^6*b^3*c^4*g*i*k*z + 7680*a^6*b^3*c^4*f*j*k*z - 3072*a^6*b^3*c^4*h*i
*j*z - 3072*a^5*b^5*c^3*g*i*k*z - 1920*a^5*b^5*c^3*f*j*k*z + 768*a^5*b^5*c^
3*h*i*j*z + 256*a^4*b^7*c^2*g*i*k*z + 160*a^4*b^7*c^2*f*j*k*z - 64*a^4*b^7*
c^2*h*i*j*z - 65536*a^6*b^2*c^5*d*j*k*z - 24576*a^6*b^2*c^5*e*i*k*z + 21504
*a^5*b^4*c^4*d*j*k*z + 9216*a^6*b^2*c^5*f*i*j*z + 6144*a^5*b^4*c^4*e*i*k*z
 - 3072*a^5*b^4*c^4*f*h*k*z - 3072*a^4*b^6*c^3*d*j*k*z - 2304*a^5*b^4*c^4*f*
i*j*z - 2048*a^6*b^2*c^5*g*h*j*z + 1536*a^5*b^4*c^4*g*h*j*z + 1024*a^4*b^6*
c^3*f*h*k*z - 512*a^4*b^6*c^3*e*i*k*z - 384*a^4*b^6*c^3*g*h*j*z + 192*a^4*b
^6*c^3*f*i*j*z + 160*a^3*b^8*c^2*d*j*k*z - 96*a^3*b^8*c^2*f*h*k*z + 32*a^3*
b^8*c^2*g*h*j*z + 41472*a^5*b^3*c^5*d*h*k*z - 13440*a^4*b^5*c^4*d*h*k*z + 1
2288*a^5*b^3*c^5*e*g*k*z - 4608*a^5*b^3*c^5*f*g*j*z - 3072*a^5*b^3*c^5*e*h*
j*z - 3072*a^4*b^5*c^4*e*g*k*z + 1888*a^3*b^7*c^3*d*h*k*z + 1152*a^4*b^5*c^
4*f*g*j*z + 768*a^4*b^5*c^4*e*h*j*z + 256*a^3*b^7*c^3*e*g*k*z - 96*a^3*b^7*
c^3*f*g*j*z - 96*a^2*b^9*c^2*d*h*k*z - 64*a^3*b^7*c^3*e*h*j*z + 9216*a^5*b^
2*c^6*e*f*j*z - 9216*a^5*b^2*c^6*d*h*i*z - 6656*a^4*b^4*c^5*d*f*k*z - 6144*
a^5*b^2*c^6*d*f*k*z + 3456*a^3*b^6*c^4*d*f*k*z - 2304*a^4*b^4*c^5*e*f*j*z +
 2304*a^4*b^4*c^5*d*h*i*z - 576*a^2*b^8*c^3*d*f*k*z + 192*a^3*b^6*c^4*e*f*j
*z - 192*a^3*b^6*c^4*d*h*i*z + 4608*a^4*b^3*c^6*d*g*h*z + 3072*a^4*b^3*c^6*
d*f*i*z - 1152*a^3*b^5*c^5*d*g*h*z - 768*a^3*b^5*c^5*d*f*i*z + 96*a^2*b^7*c
^4*d*g*h*z + 64*a^2*b^7*c^4*d*f*i*z - 9216*a^4*b^2*c^7*d*e*h*z + 2304*a^3*b
^4*c^6*d*e*h*z + 2048*a^4*b^2*c^7*d*f*g*z - 1536*a^3*b^4*c^6*d*f*g*z + 384*
a^2*b^6*c^5*d*f*g*z - 192*a^2*b^6*c^5*d*e*h*z + 3072*a^3*b^3*c^7*d*e*f*z -
768*a^2*b^5*c^6*d*e*f*z - 3072*a^8*b*c^4*j^2*k*z + 48*a^5*b^7*c*j^2*k*z - 4
9152*a^8*b*c^4*i*k^2*z + 2304*a^5*b^7*c*i*k^2*z - 9216*a^7*b*c^5*h^2*k*z -
32*a^4*b^8*c*i*j^2*z - 1152*a^4*b^8*c*g*k^2*z + 9216*a^7*b*c^5*g*j^2*z - 30
72*a^6*b*c^6*f^2*k*z + 16*a^3*b^9*c*g*j^2*z - 49152*a^7*b*c^5*e*k^2*z - 128
*a^3*b^9*c*e*k^2*z - 58368*a^5*b*c^7*d^2*k*z - 1024*a^6*b*c^6*g*h^2*z - 432
*a*b^9*c^3*d^2*k*z + 1024*a^5*b*c^7*f^2*g*z + 32*a*b^8*c^4*d^2*i*z - 9216*a
^4*b*c^8*d^2*g*z + 336*a*b^7*c^5*d^2*g*z - 672*a*b^6*c^6*d^2*e*z + 24576*a^
8*c^5*h*j*k*z + 73728*a^7*c^6*d*j*k*z + 32768*a^7*c^6*e*i*k*z - 12288*a^7*c
^6*f*i*j*z + 8192*a^7*c^6*f*h*k*z + 24576*a^6*c^7*d*f*k*z - 12288*a^6*c^7*e
*f*j*z + 12288*a^6*c^7*d*h*i*z + 12288*a^5*c^8*d*e*h*z + 2304*a^7*b^3*c^3*j
^2*k*z - 576*a^6*b^5*c^2*j^2*k*z + 45056*a^7*b^3*c^3*i*k^2*z - 15360*a^6*b^
5*c^2*i*k^2*z - 12288*a^7*b^2*c^4*i^2*k*z + 3072*a^6*b^4*c^3*i^2*k*z - 256*
a^5*b^6*c^2*i^2*k*z + 15872*a^7*b^2*c^4*i*j^2*z + 6912*a^6*b^3*c^4*h^2*k*z
 - 4992*a^6*b^4*c^3*i*j^2*z - 1728*a^5*b^5*c^3*h^2*k*z + 672*a^5*b^6*c^2*i*j
^2*z + 144*a^4*b^7*c^2*h^2*k*z + 24576*a^7*b^2*c^4*g*k^2*z - 22528*a^6*b^4*

c^3*g*k^2*z + 7680*a^5*b^6*c^2*g*k^2*z + 4096*a^6*b^2*c^5*g^2*k*z - 3072*a^5*b^4*c^4*g^2*k*z + 768*a^4*b^6*c^3*g^2*k*z - 64*a^3*b^8*c^2*g^2*k*z - 7936*a^6*b^3*c^4*g*j^2*z + 2496*a^5*b^5*c^3*g*j^2*z - 1536*a^6*b^2*c^5*h^2*i*z + 1280*a^5*b^3*c^5*f^2*k*z + 384*a^5*b^4*c^4*h^2*i*z - 336*a^4*b^7*c^2*g*j^2*z + 192*a^4*b^5*c^4*f^2*k*z - 144*a^3*b^7*c^3*f^2*k*z - 32*a^4*b^6*c^3*h^2*i*z + 16*a^2*b^9*c^2*f^2*k*z + 45056*a^6*b^3*c^4*e*k^2*z - 15360*a^5*b^5*c^3*e*k^2*z - 12288*a^5*b^2*c^6*e^2*k*z + 3072*a^4*b^4*c^5*e^2*k*z + 2304*a^4*b^7*c^2*e*k^2*z - 256*a^3*b^6*c^4*e^2*k*z + 59136*a^4*b^3*c^6*d^2*k*z - 23488*a^3*b^5*c^5*d^2*k*z + 15872*a^6*b^2*c^5*e*j^2*z - 4992*a^5*b^4*c^4*e*j^2*z + 4560*a^2*b^7*c^4*d^2*k*z + 1536*a^5*b^2*c^6*f^2*i*z + 768*a^5*b^3*c^5*g*h^2*z + 672*a^4*b^6*c^3*e*j^2*z - 384*a^4*b^4*c^5*f^2*i*z - 192*a^4*b^5*c^4*g*h^2*z - 32*a^3*b^8*c^2*e*j^2*z + 32*a^3*b^6*c^4*f^2*i*z + 16*a^3*b^7*c^3*g*h^2*z - 15872*a^4*b^2*c^7*d^2*i*z + 4992*a^3*b^4*c^6*d^2*i*z - 1536*a^5*b^2*c^6*e*h^2*z - 768*a^4*b^3*c^6*f^2*g*z - 672*a^2*b^6*c^5*d^2*i*z + 384*a^4*b^4*c^5*e*h^2*z + 192*a^3*b^5*c^5*f^2*g*z - 32*a^3*b^6*c^4*e*h^2*z - 16*a^2*b^7*c^4*f^2*g*z + 7936*a^3*b^3*c^7*d^2*g*z - 2496*a^2*b^5*c^6*d^2*g*z + 1536*a^4*b^2*c^7*e*f^2*z - 384*a^3*b^4*c^6*e*f^2*z + 32*a^2*b^6*c^5*e*f^2*z - 15872*a^3*b^2*c^8*d^2*e*z + 4992*a^2*b^4*c^7*d^2*e*z - 61440*a^8*b^2*c^3*k^3*z + 21504*a^7*b^4*c^2*k^3*z + 16384*a^8*c^5*i^2*k*z - 18432*a^8*c^5*i*j^2*z - 128*a^4*b^9*i*k^2*z + 2048*a^7*c^6*h^2*i*z + 64*a^3*b^10*g*k^2*z + 16384*a^6*c^7*e^2*k*z + 16*b^11*c^2*d^2*k*z - 18432*a^7*c^6*e*j^2*z - 2048*a^6*c^7*f^2*i*z + 18432*a^5*c^8*d^2*i*z - 3328*a^6*b^6*c*k^3*z + 2048*a^6*c^7*e*h^2*z - 16*b^9*c^4*d^2*g*z - 2048*a^5*c^8*e*f^2*z + 32*b^8*c^5*d^2*e*z + 18432*a^4*c^9*d^2*e*z + 65536*a^9*c^4*k^3*z + 192*a^5*b^8*k^3*z - 3328*a^7*b*c^3*h*i*j*k - 6912*a^6*b*c^4*d*i*j*k - 3328*a^6*b*c^4*e*h*j*k - 1536*a^6*b*c^4*f*g*j*k - 768*a^6*b*c^4*g*h*i*j - 768*a^6*b*c^4*f*h*i*k - 6912*a^5*b*c^5*d*e*j*k - 2304*a^5*b*c^5*d*g*i*j - 1792*a^5*b*c^5*e*f*i*j + 1536*a^5*b*c^5*d*g*h*k - 1280*a^5*b*c^5*d*f*i*k - 768*a^5*b*c^5*e*g*h*j - 768*a^5*b*c^5*e*f*h*k - 256*a^5*b*c^5*f*g*h*i + 16*a*b^8*c^2*d*f*g*k - 4*a*b^8*c^2*d*f*h*j - 2304*a^4*b*c^6*d*e*g*j - 1792*a^4*b*c^6*d*e*h*i - 1280*a^4*b*c^6*d*e*f*k - 768*a^4*b*c^6*d*f*g*i - 256*a^4*b*c^6*e*f*g*h - 32*a*b^7*c^3*d*e*f*k - 768*a^3*b*c^7*d*e*f*g + 32*a*b^5*c^5*d*e*f*g + 576*a^6*b^3*c^2*h*i*j*k + 1664*a^6*b^2*c^3*g*h*j*k + 384*a^6*b^2*c^3*f*i*j*k - 288*a^5*b^4*c^2*g*h*j*k - 160*a^5*b^4*c^2*f*i*j*k + 2112*a^5*b^3*c^3*d*i*j*k + 576*a^5*b^3*c^3*e*h*j*k - 448*a^5*b^3*c^3*f*h*i*k - 192*a^5*b^3*c^3*g*h*i*j - 192*a^5*b^3*c^3*f*g*j*k - 160*a^4*b^5*c^2*d*i*j*k + 96*a^4*b^5*c^2*f*h*i*k + 80*a^4*b^5*c^2*f*g*j*k + 32*a^4*b^5*c^2*g*h*i*j + 4992*a^5*b^2*c^4*d*h*i*k - 4608*a^5*b^2*c^4*e*g*i*k + 3456*a^5*b^2*c^4*d*g*j*k - 1312*a^4*b^4*c^3*d*h*i*k - 1056*a^4*b^4*c^3*d*g*j*k + 896*a^5*b^2*c^4*f*g*i*j + 768*a^4*b^4*c^3*e*g*i*k + 384*a^5*b^2*c^4*f*g*h*k + 384*a^5*b^2*c^4*e*h*i*j + 384*a^5*b^2*c^4*e*f*j*k + 224*a^4*b^4*c^3*f*g*h*k - 160*a^4*b^4*c^3*e*f*j*k - 96*a^4*b^4*c^3*f*g*i*j + 96*a^3*b^6*c^2*d*h*i*k + 80*a^3*b^6*c^2*d*g*j*k - 64*a^4*b^4*c^3*e*h*i*j - 48*a^3*b^6*c^2*f*g*h*k - 2496*a^4*b^3*c^4*d*g*h*k + 2112*a^4*b^3*c^4*d*e*j*k - 960*a^4*b^3*c^4*d*f*i*k + 656*a^3*b^5*c^3*d*g*h*k - 448*a^4*b^3*c^4*e*f*h*k + 384*a^3*b^5*c^3*d*f*i*k + 320*a^4*b^3*c^4*d*g*i*j - 192

*a^4*b^3*c^4*f*g*h*i - 192*a^4*b^3*c^4*e*g*h*j + 192*a^4*b^3*c^4*e*f*i*j -
160*a^3*b^5*c^3*d*e*j*k + 96*a^3*b^5*c^3*e*f*h*k - 48*a^2*b^7*c^2*d*g*h*k +
32*a^3*b^5*c^3*e*g*h*j - 32*a^2*b^7*c^2*d*f*i*k + 4992*a^4*b^2*c^5*d*e*h*k
- 3584*a^4*b^2*c^5*d*f*h*j - 1312*a^3*b^4*c^4*d*e*h*k + 896*a^4*b^2*c^5*e*
f*g*j + 896*a^4*b^2*c^5*d*g*h*i + 640*a^4*b^2*c^5*d*f*g*k - 640*a^4*b^2*c^5
*d*e*i*j + 600*a^3*b^4*c^4*d*f*h*j + 480*a^3*b^4*c^4*d*f*g*k + 384*a^4*b^2*
c^5*e*f*h*i - 192*a^2*b^6*c^3*d*f*g*k - 96*a^3*b^4*c^4*e*f*g*j - 96*a^3*b^4
*c^4*d*g*h*i + 96*a^2*b^6*c^3*d*e*h*k + 12*a^2*b^6*c^3*d*f*h*j - 960*a^3*b^
3*c^5*d*e*f*k + 384*a^2*b^5*c^4*d*e*f*k + 320*a^3*b^3*c^5*d*e*g*j - 192*a^3
*b^3*c^5*e*f*g*h - 192*a^3*b^3*c^5*d*f*g*i + 192*a^3*b^3*c^5*d*e*h*i + 32*a
^2*b^5*c^4*d*f*g*i + 896*a^3*b^2*c^6*d*e*g*h + 384*a^3*b^2*c^6*d*e*f*i - 96
*a^2*b^4*c^5*d*e*g*h - 64*a^2*b^4*c^5*d*e*f*i - 192*a^2*b^3*c^6*d*e*f*g + 4
8*a^6*b^4*c*i*j^2*k - 1424*a^6*b^4*c*h*j*k^2 - 2304*a^7*b*c^3*g*j^2*k - 24*
a^5*b^5*c*g*j^2*k + 2048*a^7*b*c^3*g*i*k^2 - 1024*a^7*b*c^3*f*j*k^2 - 768*a
^5*b^5*c*g*i*k^2 + 408*a^5*b^5*c*f*j*k^2 + 256*a^6*b*c^4*g*h^2*k + 16*a^4*b
^6*c*g*i*j^2 + 4608*a^6*b*c^4*e*i^2*k + 4608*a^5*b*c^5*e^2*i*k - 896*a^6*b*
c^4*f*i^2*j + 768*a^4*b^6*c*d*j*k^2 - 256*a^4*b^6*c*f*h*k^2 - 128*a^4*b^6*c
*e*i*k^2 + 2208*a^6*b*c^4*f*h*j^2 - 1920*a^6*b*c^4*e*i*j^2 + 800*a^5*b*c^5*
f^2*h*j - 256*a^5*b*c^5*f^2*g*k - 16*a*b^8*c^2*d^2*i*k + 6*a^3*b^7*c*f*h*j^
2 + 8192*a^6*b*c^4*d*h*k^2 + 2048*a^6*b*c^4*e*g*k^2 - 472*a^3*b^7*c*d*h*k^2
+ 64*a^3*b^7*c*e*g*k^2 + 4896*a^4*b*c^6*d^2*h*j + 2304*a^4*b*c^6*d^2*g*k +
1824*a^5*b*c^5*d*h^2*j - 384*a^5*b*c^5*e*h^2*i - 168*a*b^7*c^3*d^2*g*k + 4
2*a*b^7*c^3*d^2*h*j + 6*a^2*b^8*c*d*h*j^2 + 1536*a^5*b*c^5*e*g*i^2 + 1536*a
^4*b*c^6*e^2*g*i - 896*a^5*b*c^5*d*h*i^2 - 896*a^4*b*c^6*e^2*f*j + 144*a^2*
b^8*c*d*f*k^2 + 4896*a^5*b*c^5*d*f*j^2 + 1824*a^4*b*c^6*d*f^2*j - 384*a^4*b
*c^6*e*f^2*i + 336*a*b^6*c^4*d^2*e*k - 156*a*b^6*c^4*d^2*f*j + 16*a*b^6*c^4
*d^2*g*i + 12*a*b^7*c^3*d*f^2*j + 2208*a^3*b*c^7*d^2*f*h - 1920*a^3*b*c^7*d
^2*e*i + 800*a^4*b*c^6*d*f*h^2 - 102*a*b^5*c^5*d^2*f*h - 32*a*b^5*c^5*d^2*e
*i + 12*a*b^6*c^4*d*f^2*h - 2*a*b^7*c^3*d*f*h^2 - 896*a^3*b*c^7*d*e^2*h - 8
*a*b^6*c^4*d*f*g^2 - 240*a*b^4*c^6*d^2*e*g - 32*a*b^4*c^6*d*e^2*f + 3072*a^
7*c^4*f*i*j*k + 3072*a^6*c^5*e*f*j*k - 3072*a^6*c^5*d*h*i*k + 1536*a^6*c^5*
e*h*i*j + 4608*a^5*c^6*d*e*i*j - 3072*a^5*c^6*d*e*h*k - 1152*a^5*c^6*d*f*h*
j + 512*a^5*c^6*e*f*h*i + 1536*a^4*c^7*d*e*f*i - 2*a*b^9*c*d*f*j^2 - 1088*a
^7*b^2*c^2*i*j^2*k + 4800*a^7*b^2*c^2*h*j*k^2 + 960*a^6*b^2*c^3*h^2*i*k + 5
44*a^6*b^3*c^2*g*j^2*k - 144*a^5*b^4*c^2*h^2*i*k - 2304*a^6*b^2*c^3*g*i^2*k
+ 1920*a^6*b^3*c^2*g*i*k^2 + 1152*a^5*b^3*c^3*g^2*i*k - 864*a^6*b^3*c^2*f*
j*k^2 + 384*a^5*b^4*c^2*g*i^2*k + 192*a^6*b^2*c^3*h*i^2*j - 192*a^4*b^5*c^2
*g^2*i*k - 32*a^5*b^4*c^2*h*i^2*j - 1088*a^6*b^2*c^3*e*j^2*k + 960*a^6*b^2*
c^3*g*i*j^2 - 480*a^5*b^3*c^3*g*h^2*k - 240*a^5*b^4*c^2*g*i*j^2 + 192*a^5*b
^2*c^4*f^2*i*k + 72*a^4*b^5*c^2*g*h^2*k + 48*a^5*b^4*c^2*e*j^2*k + 48*a^4*b
^4*c^3*f^2*i*k - 16*a^3*b^6*c^2*f^2*i*k + 13376*a^6*b^2*c^3*d*j*k^2 - 5136*
a^5*b^4*c^2*d*j*k^2 - 3840*a^6*b^2*c^3*e*i*k^2 + 1536*a^5*b^4*c^2*e*i*k^2 -
768*a^5*b^3*c^3*e*i^2*k - 768*a^4*b^3*c^4*e^2*i*k + 624*a^5*b^4*c^2*f*h*k^
2 + 576*a^6*b^2*c^3*f*h*k^2 + 192*a^5*b^2*c^4*g^2*h*j + 96*a^5*b^3*c^3*f*i^
2*j + 48*a^4*b^4*c^3*g^2*h*j - 8*a^3*b^6*c^2*g^2*h*j + 6848*a^4*b^2*c^5*d^2

*i*k - 2448*a^3*b^4*c^4*d^2*i*k + 960*a^5*b^2*c^4*e*h^2*k - 864*a^5*b^2*c^4
*f*h^2*j + 480*a^5*b^3*c^3*e*i*j^2 + 336*a^4*b^3*c^4*f^2*h*j + 336*a^2*b^6*
c^3*d^2*i*k + 192*a^5*b^2*c^4*g*h^2*i + 144*a^5*b^3*c^3*f*h*j^2 - 144*a^4*b
^4*c^3*e*h^2*k - 102*a^4*b^5*c^2*f*h*j^2 - 96*a^4*b^3*c^4*f^2*g*k - 32*a^4*
b^5*c^2*e*i*j^2 - 30*a^3*b^5*c^3*f^2*h*j - 24*a^3*b^5*c^3*f^2*g*k + 16*a^4*
b^4*c^3*g*h^2*i - 12*a^4*b^4*c^3*f*h^2*j + 12*a^3*b^6*c^2*f*h^2*j + 8*a^2*b
^7*c^2*f^2*g*k - 2*a^2*b^7*c^2*f^2*h*j - 9312*a^5*b^3*c^3*d*h*k^2 + 3288*a^
4*b^5*c^2*d*h*k^2 - 2304*a^4*b^2*c^5*e^2*g*k + 1920*a^5*b^3*c^3*e*g*k^2 + 1
152*a^4*b^3*c^4*e*g^2*k - 768*a^4*b^5*c^2*e*g*k^2 + 384*a^3*b^4*c^4*e^2*g*k
 - 320*a^5*b^2*c^4*d*i^2*j - 224*a^4*b^3*c^4*f*g^2*j + 192*a^5*b^2*c^4*f*h*
i^2 + 192*a^4*b^2*c^5*e^2*h*j - 192*a^3*b^5*c^3*e*g^2*k - 32*a^3*b^4*c^4*e^
2*h*j + 24*a^3*b^5*c^3*f*g^2*j - 3552*a^5*b^2*c^4*d*h*j^2 - 3424*a^3*b^3*c^
5*d^2*g*k + 1332*a^4*b^4*c^3*d*h*j^2 + 1224*a^2*b^5*c^4*d^2*g*k + 960*a^5*b
^2*c^4*e*g*j^2 - 496*a^3*b^3*c^5*d^2*h*j + 432*a^4*b^3*c^4*d*h^2*j - 240*a^
4*b^4*c^3*e*g*j^2 - 222*a^2*b^5*c^4*d^2*h*j + 192*a^4*b^2*c^5*f^2*g*i + 192
*a^4*b^2*c^5*e*f^2*k - 174*a^3*b^5*c^3*d*h^2*j - 156*a^3*b^6*c^2*d*h*j^2 +
48*a^3*b^4*c^4*e*f^2*k - 32*a^4*b^3*c^4*e*h^2*i + 16*a^3*b^6*c^2*e*g*j^2 +
16*a^3*b^4*c^4*f^2*g*i - 16*a^2*b^6*c^3*e*f^2*k + 12*a^2*b^7*c^2*d*h^2*j +
1728*a^5*b^2*c^4*d*f*k^2 + 1392*a^4*b^4*c^3*d*f*k^2 - 840*a^3*b^6*c^2*d*f*k
^2 - 768*a^4*b^2*c^5*e*g^2*i + 576*a^4*b^2*c^5*d*g^2*j + 96*a^4*b^3*c^4*d*h
*i^2 + 96*a^3*b^3*c^5*e^2*f*j - 80*a^3*b^4*c^4*d*g^2*j + 64*a^4*b^2*c^5*f*g
^2*h + 48*a^3*b^4*c^4*f*g^2*h + 6848*a^3*b^2*c^6*d^2*e*k - 3552*a^3*b^2*c^6
*d^2*f*j - 2448*a^2*b^4*c^5*d^2*e*k + 1332*a^2*b^4*c^5*d^2*f*j + 960*a^3*b^
2*c^6*d^2*g*i - 496*a^4*b^3*c^4*d*f*j^2 + 432*a^3*b^3*c^5*d*f^2*j - 240*a^2
*b^4*c^5*d^2*g*i - 222*a^3*b^5*c^3*d*f*j^2 + 192*a^4*b^2*c^5*e*g*h^2 - 174*
a^2*b^5*c^4*d*f^2*j + 42*a^2*b^7*c^2*d*f*j^2 - 32*a^3*b^3*c^5*e*f^2*i + 16*
a^3*b^4*c^4*e*g*h^2 - 320*a^3*b^2*c^6*d*e^2*j - 224*a^3*b^3*c^5*d*g^2*h + 1
92*a^4*b^2*c^5*d*f*i^2 + 192*a^3*b^2*c^6*e^2*f*h - 32*a^3*b^4*c^4*d*f*i^2 +
 24*a^2*b^5*c^4*d*g^2*h - 864*a^3*b^2*c^6*d*f^2*h + 480*a^2*b^3*c^6*d^2*e*i
 + 336*a^3*b^3*c^5*d*f*h^2 + 192*a^3*b^2*c^6*e*f^2*g + 144*a^2*b^3*c^6*d^2*
f*h - 30*a^2*b^5*c^4*d*f*h^2 + 16*a^2*b^4*c^5*e*f^2*g - 12*a^2*b^4*c^5*d*f^
2*h + 192*a^3*b^2*c^6*d*f*g^2 + 96*a^2*b^3*c^6*d*e^2*h + 48*a^2*b^4*c^5*d*f
*g^2 + 960*a^2*b^2*c^7*d^2*e*g + 192*a^2*b^2*c^7*d*e^2*f - 3072*a^8*b*c^2*j
^2*k^2 + 1104*a^7*b^3*c*j^2*k^2 + 768*a^6*b^4*c*i^2*k^2 - 256*a^6*b^3*c^2*i
^3*k + 1536*a^7*b*c^3*h^2*k^2 - 960*a^7*b*c^3*i^2*j^2 + 444*a^5*b^5*c*h^2*k
^2 - 16*a^5*b^5*c*i^2*j^2 - 3072*a^7*b^2*c^2*g*k^3 - 496*a^6*b^3*c^2*h*j^3
+ 192*a^4*b^6*c*g^2*k^2 - 192*a^4*b^4*c^3*g^3*k + 144*a^5*b^3*c^3*h^3*j + 3
2*a^3*b^6*c^2*g^3*k - 18*a^4*b^5*c^2*h^3*j - 9*a^4*b^6*c*h^2*j^2 - 192*a^6*
b*c^4*h^2*i^2 + 36*a^3*b^7*c*f^2*k^2 - 4*a^3*b^7*c*g^2*j^2 - 2176*a^6*b^3*c
^2*e*k^3 - 256*a^3*b^3*c^5*e^3*k - 192*a^6*b^2*c^3*f*j^3 - 192*a^4*b^2*c^5*
f^3*j + 132*a^5*b^4*c^2*f*j^3 + 128*a^4*b^3*c^4*g^3*i - 28*a^3*b^4*c^4*f^3*
j + 6*a^2*b^6*c^3*f^3*j + 10752*a^5*b*c^5*d^2*k^2 - 960*a^5*b*c^5*e^2*j^2 -
 192*a^5*b*c^5*f^2*i^2 - 1680*a^5*b^3*c^3*d*j^3 - 1680*a^2*b^3*c^6*d^3*j +
222*a^4*b^5*c^2*d*j^3 + 80*a^4*b^3*c^4*f*h^3 + 80*a^3*b^3*c^5*f^3*h + 30*a*
b^8*c^2*d^2*j^2 + 6*a^3*b^5*c^3*f*h^3 + 6*a^2*b^5*c^4*f^3*h - 960*a^4*b*c^6

*d^2*i^2 - 192*a^4*b*c^6*e^2*h^2 - 192*a^4*b^2*c^5*d*h^3 - 192*a^2*b^2*c^7*
d^3*h + 128*a^3*b^3*c^5*e*g^3 - 28*a^3*b^4*c^4*d*h^3 + 12*a*b^6*c^4*d^2*h^2
 + 6*a^2*b^6*c^3*d*h^3 - 192*a^3*b*c^7*e^2*f^2 + 60*a*b^5*c^5*d^2*g^2 + 198
*a*b^4*c^6*d^2*f^2 + 144*a^2*b^3*c^6*d*f^3 - 960*a^2*b*c^8*d^2*e^2 + 240*a*
b^3*c^7*d^2*e^2 + 4608*a^8*c^3*i*j^2*k - 3072*a^8*c^3*h*j*k^2 - 512*a^7*c^4
*h^2*i*k + 120*a^5*b^6*h*j*k^2 + 768*a^7*c^4*h*i^2*j + 4608*a^7*c^4*e*j^2*k
 + 512*a^6*c^5*f^2*i*k + 64*a^4*b^7*g*i*k^2 - 40*a^4*b^7*f*j*k^2 - 9216*a^7
*c^4*d*j*k^2 - 4096*a^7*c^4*e*i*k^2 - 1024*a^7*c^4*f*h*k^2 - 4608*a^5*c^6*d
^2*i*k - 512*a^6*c^5*e*h^2*k - 192*a^6*c^5*f*h^2*j - 40*a^3*b^8*d*j*k^2 + 2
4*a^3*b^8*f*h*k^2 + 2304*a^6*c^5*d*i^2*j + 768*a^5*c^6*e^2*h*j + 256*a^6*c^
5*f*h*i^2 + 8*b^9*c^2*d^2*g*k - 2*b^9*c^2*d^2*h*j + 6144*a^8*b*c^2*i*k^3 -
2176*a^7*b^3*c*i*k^3 - 1728*a^6*c^5*d*h*j^2 + 1536*a^7*b*c^3*i^3*k + 512*a^
5*c^6*e*f^2*k + 24*a^2*b^9*d*h*k^2 - 3072*a^6*c^5*d*f*k^2 - 16*b^8*c^3*d^2*
e*k + 6*b^8*c^3*d^2*f*j - 4608*a^4*c^7*d^2*e*k + 2016*a^7*b*c^3*h*j^3 - 172
8*a^4*c^7*d^2*f*j + 1088*a^6*b^4*c*g*k^3 + 224*a^6*b*c^4*h^3*j + 30*a^5*b^5
*c*h*j^3 + 2304*a^4*c^7*d*e^2*j + 768*a^5*c^6*d*f*i^2 + 256*a^4*c^7*e^2*f*h
 + 6*b^7*c^4*d^2*f*h + 6144*a^7*b*c^3*e*k^3 + 1536*a^4*b*c^6*e^3*k + 512*a^
6*b*c^4*g*i^3 + 192*a^5*b^5*c*e*k^3 - 192*a^4*c^7*d*f^2*h - 10*a^4*b^6*c*f*
j^3 + 108*a*b^9*c*d^2*k^2 + 16*b^6*c^5*d^2*e*g + 4320*a^6*b*c^4*d*j^3 + 432
0*a^3*b*c^7*d^3*j + 222*a*b^5*c^5*d^3*j + 96*a^5*b*c^5*f*h^3 + 96*a^4*b*c^6
*f^3*h - 10*a^3*b^7*c*d*j^3 + 768*a^3*c^8*d*e^2*f + 512*a^3*b*c^7*e^3*g + 1
32*a*b^4*c^6*d^3*h + 2016*a^2*b*c^8*d^3*f - 496*a*b^3*c^7*d^3*f + 224*a^3*b
*c^7*d*f^3 - 18*a*b^5*c^5*d*f^3 - 1920*a^7*b^2*c^2*i^2*k^2 - 1648*a^6*b^3*c
^2*h^2*k^2 + 240*a^6*b^3*c^2*i^2*j^2 - 960*a^6*b^2*c^3*h^2*j^2 - 512*a^6*b^
2*c^3*g^2*k^2 - 480*a^5*b^4*c^2*g^2*k^2 + 198*a^5*b^4*c^2*h^2*j^2 - 240*a^5
*b^3*c^3*g^2*j^2 - 240*a^5*b^3*c^3*f^2*k^2 + 60*a^4*b^5*c^2*g^2*j^2 - 36*a^
4*b^5*c^2*f^2*k^2 - 16*a^5*b^3*c^3*h^2*i^2 - 1920*a^5*b^2*c^4*e^2*k^2 + 768
*a^4*b^4*c^3*e^2*k^2 - 464*a^5*b^2*c^4*f^2*j^2 - 384*a^5*b^2*c^4*g^2*i^2 -
64*a^3*b^6*c^2*e^2*k^2 + 42*a^4*b^4*c^3*f^2*j^2 + 12*a^3*b^6*c^2*f^2*j^2 -
13104*a^4*b^3*c^4*d^2*k^2 + 5628*a^3*b^5*c^3*d^2*k^2 - 1128*a^2*b^7*c^2*d^2
*k^2 + 240*a^4*b^3*c^4*e^2*j^2 - 48*a^4*b^3*c^4*g^2*h^2 - 16*a^4*b^3*c^4*f^
2*i^2 - 16*a^3*b^5*c^3*e^2*j^2 - 4*a^3*b^5*c^3*g^2*h^2 - 2880*a^4*b^2*c^5*d
^2*j^2 + 1750*a^3*b^4*c^4*d^2*j^2 - 345*a^2*b^6*c^3*d^2*j^2 - 192*a^4*b^2*c
^5*f^2*h^2 - 42*a^3*b^4*c^4*f^2*h^2 + 240*a^3*b^3*c^5*d^2*i^2 - 48*a^3*b^3*
c^5*f^2*g^2 - 16*a^3*b^3*c^5*e^2*h^2 - 16*a^2*b^5*c^4*d^2*i^2 - 4*a^2*b^5*c
^4*f^2*g^2 - 464*a^3*b^2*c^6*d^2*h^2 - 384*a^3*b^2*c^6*e^2*g^2 + 42*a^2*b^4
*c^5*d^2*h^2 - 240*a^2*b^3*c^6*d^2*g^2 - 16*a^2*b^3*c^6*e^2*f^2 - 960*a^2*b
^2*c^7*d^2*f^2 - 8*a*b^10*d*f*k^2 - a^2*b^8*c*f^2*j^2 - 2048*a^8*c^3*i^2*k^
2 - 100*a^6*b^5*j^2*k^2 - 64*a^5*b^6*i^2*k^2 - 288*a^7*c^4*h^2*j^2 - 36*a^4
*b^7*h^2*k^2 - 16*a^3*b^8*g^2*k^2 - 2048*a^6*c^5*e^2*k^2 - 864*a^6*c^5*f^2*
j^2 - 4*a^2*b^9*f^2*k^2 - 2592*a^5*c^6*d^2*j^2 - 1536*a^5*c^6*e^2*i^2 - 32*
a^5*c^6*f^2*h^2 - 864*a^4*c^7*d^2*h^2 + 360*a^7*b^2*c^2*j^4 - 4*b^7*c^4*d^2
*g^2 - 9*b^6*c^5*d^2*f^2 - 288*a^3*c^8*d^2*f^2 - 24*a^5*b^2*c^4*h^4 - 16*b^
5*c^6*d^2*e^2 - 9*a^4*b^4*c^3*h^4 - 16*a^3*b^4*c^4*g^4 - 24*a^3*b^2*c^6*f^4
 - 9*a^2*b^4*c^5*f^4 - a^2*b^6*c^3*f^2*h^2 + 192*a^6*b^5*i*k^3 - 96*a^5*b^6

```
*g*k^3 - 1728*a^7*c^4*f*j^3 - 192*a^5*c^6*f^3*j - 10*b^7*c^4*d^3*j - 1024*a
^6*c^5*e*i^3 - 1024*a^4*c^7*e^3*i + 1536*a^8*b^2*c*k^4 - 10*b^6*c^5*d^3*h -
 1728*a^3*c^8*d^3*h - 192*a^5*c^6*d*h^3 - 25*a^6*b^4*c*j^4 + 30*b^5*c^6*d^3
*f + 360*a*b^2*c^8*d^4 - 4*b^11*d^2*k^2 - 4096*a^9*c^2*k^4 - 1296*a^8*c^3*j
^4 - 144*a^7*b^4*k^4 - 256*a^7*c^4*i^4 - 16*a^6*c^5*h^4 - 16*a^4*c^7*f^4 -
256*a^3*c^8*e^4 - 25*b^4*c^7*d^4 - 1296*a^2*c^9*d^4 - b^8*c^3*d^2*h^2 - b^1
0*c*d^2*j^2, z, n), n, 1, 4)
```

3.59 $\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx$

Optimal result

Integrand size = 50, antiderivative size = 1177

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx =$$

$$-\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j))\,x^2\right)}{4ac^2\,(b^2-4ac)\,(a+bx^2+cx^4)^2}$$

$$-\frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck-5a^2b}{4c^4\,(b^2-4ac)\,(a+bx^2+cx^4)^2}$$

$$+\frac{x\left(c\left(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\right)+(ab^2c^2f}{8a^2c\,(b^2-4ac)^2\,(a+bx^2+cx^4)}$$

$$+\frac{b^3c^2i+2bc^3(3ce+ai)+11ab^4k-\frac{b^6k}{c}+32a^3c^2k-3b^2(c^3g+13a^2ck)+2(6c^5e+b^2c^3i-c^4(3bg-2ai)}{4c^3\,(b^2-4ac)^2\,(a+bx^2+cx^4)}$$

$$+\frac{\left(ab^2c^2f+20a^2c^3f+b^3(3c^2d+a^2j)-4abc(6c^2d+3ach+4a^2j)+\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a^2j)+b}{\sqrt{b^2-4a}}\right.}{8\sqrt{2}a^2c^{3/2}\,(b^2-4ac)^2\,\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(ab^2c^2f+20a^2c^3f+b^3(3c^2d+a^2j)-4abc(6c^2d+3ach+4a^2j)-\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a^2j)+b}{\sqrt{b^2-4a}}\right.}{8\sqrt{2}a^2c^{3/2}\,(b^2-4ac)^2\,\sqrt{b+\sqrt{b^2-4ac}}}$$

$$-\frac{(12c^5e+2b^2c^3i-c^4(6bg-4ai)-b^5k+10ab^3ck-30a^2bc^2k)\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^3\,(b^2-4ac)^{5/2}}$$

$$+\frac{k\log(a+bx^2+cx^4)}{4c^3}$$

```
[Out] -1/4*x*(c^2*(a*b*f-b^2*(d+a^2*j/c^2)+2*a*(c*d-a*h+a^2*j/c))+(2*a*c^3*f-a*b^
3*j-b*c*(-3*a^2*j+a*c*h+c^2*d))*x^2)/a/c^2/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1
```

```
/4*(-b*c^3*(a*i+c*e)+a*b^4*k-4*a^2*b^2*c*k+2*a*c^2*(a^2*k+c^2*g)-(2*c^5*e+b
^2*c^3*i-c^4*(2*a*i+b*g)-b^5*k+5*a*b^3*c*k-5*a^2*b*c^2*k)*x^2)/c^4/(-4*a*c+
b^2)/(c*x^4+b*x^2+a)^2+1/8*x*(c*(a*b^3*f+8*a^2*b*c*f+4*a^2*(-9*a^2*j+a*c*h+
7*c^2*d)+b^4*(3*d-2*a^2*j/c^2)-a*b^2*(25*c*d+7*a*h-11*a^2*j/c))+(a*b^2*c^2*
f+20*a^2*c^3*f+b^3*(a^2*j+3*c^2*d)-4*a*b*c*(4*a^2*j+3*a*c*h+6*c^2*d))*x^2)/
a^2/c/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/4*(b^3*c^2*i+2*b*c^3*(a*i+3*c*e)+11*
a*b^4*k-b^6*k/c+32*a^3*c^2*k-3*b^2*(13*a^2*c*k+c^3*g)+2*(6*c^5*e+b^2*c^3*i-
c^4*(-2*a*i+3*b*g)+2*b^5*k-15*a*b^3*c*k+25*a^2*b*c^2*k)*x^2)/c^3/(-4*a*c+b^
2)^2/(c*x^4+b*x^2+a)-1/2*(12*c^5*e+2*b^2*c^3*i-c^4*(-4*a*i+6*b*g)-b^5*k+10*
a*b^3*c*k-30*a^2*b*c^2*k)*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/c^3/(-4*a
*c+b^2)^(5/2)+1/4*k*ln(c*x^4+b*x^2+a)/c^3+1/16*arctan(x*2^(1/2)*c^(1/2)/(b-
(-4*a*c+b^2)^(1/2))^(1/2))*(a*b^2*c^2*f+20*a^2*c^3*f+b^3*(a^2*j+3*c^2*d)-4*
a*b*c*(4*a^2*j+3*a*c*h+6*c^2*d)+(a*b^3*c^2*f-52*a^2*b*c^3*f-6*a*b^2*c*(-3*a
^2*j-3*a*c*h+5*c^2*d)+b^4*(-a^2*j+3*c^2*d)+8*a^2*c^2*(5*a^2*j+3*a*c*h+21*c^
2*d))/(-4*a*c+b^2)^(1/2))/a^2/c^(3/2)/(-4*a*c+b^2)^2*2^(1/2)/(b-(-4*a*c+b^2
)^(1/2))^(1/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*
(a*b^2*c^2*f+20*a^2*c^3*f+b^3*(a^2*j+3*c^2*d)-4*a*b*c*(4*a^2*j+3*a*c*h+6*c^
2*d)+(-a*b^3*c^2*f+52*a^2*b*c^3*f+6*a*b^2*c*(-3*a^2*j-3*a*c*h+5*c^2*d)-b^4*
(-a^2*j+3*c^2*d)-8*a^2*c^2*(5*a^2*j+3*a*c*h+21*c^2*d))/(-4*a*c+b^2)^(1/2))/
a^2/c^(3/2)/(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 5.12 (sec) , antiderivative size = 1179, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}} = 0.200$, Rules

used = {1687, 1692, 1180, 211, 1677, 1674, 648, 632, 212, 642}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx =$$

$$-\frac{x\left(\left(-\left(\left(\frac{ja^2}{c^2}+d\right)b^2\right)+afb+2a\left(\frac{ja^2}{c}-ha+cd\right)\right)c^2+(-ajb^3-c(-3ja^2+cha+c^2d)\,b+2ac^3f)\,x^2\right.}{4ac^2\,(b^2-4ac)\,(cx^4+bx^2+a)^2}$$

$$+\frac{\left(\left(\frac{ja^2}{c}+3cd\right)b^3+acfb^2-4a(4ja^2+3cha+6c^2d)\,b+20a^2c^2f+\frac{(3c^2d-a^2j)b^4+ac^2fb^3-6ac(-3ja^2-3cha+5c^2d)}{c\sqrt{b^2-4a}}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2\,\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(\left(\frac{ja^2}{c}+3cd\right)b^3+acfb^2-4a(4ja^2+3cha+6c^2d)\,b+20a^2c^2f-\frac{(3c^2d-a^2j)b^4+ac^2fb^3-6ac(-3ja^2-3cha+5c^2d)}{c\sqrt{b^2-4a}}\right.}{8\sqrt{2}a^2\sqrt{c}\,(b^2-4ac)^2\,\sqrt{b+\sqrt{b^2-4ac}}}$$

$$-\frac{(-kb^5+10ackb^3+2c^3ib^2-30a^2c^2kb+12c^5e-c^4(6bg-4ai))\,\operatorname{arctanh}\left(\frac{2cx^2+b}{\sqrt{b^2-4ac}}\right)}{2c^3\,(b^2-4ac)^{5/2}}$$

$$+\frac{k\log\left(cx^4+bx^2+a\right)}{4c^3}$$

$$+\frac{x\left(((ja^2+3c^2d)\,b^3+ac^2fb^2-4ac(4ja^2+3cha+6c^2d)\,b+20a^2c^3f)\,x^2+c\left(\left(3d-\frac{2a^2j}{c^2}\right)b^4+afb^3-a\right.\right.}{8a^2c\,(b^2-4ac)^2\,(cx^4+bx^2+a)}$$

$$+\frac{-\frac{kb^6}{c}+11akb^4+c^2ib^3-3(gc^3+13a^2kc)\,b^2+2c^3(3ce+ai)b+2(2kb^5-15ackb^3+c^3ib^2+25a^2c^2kb+}{4c^3\,(b^2-4ac)^2\,(cx^4+bx^2+a)}$$

$$-\frac{-akb^4+4a^2ckb^2+c^3(ce+ai)b+(-kb^5+5ackb^3+c^3ib^2-5a^2c^2kb+2c^5e-c^4(bg+2ai))\,x^2-2ac^2(}{4c^4\,(b^2-4ac)\,(cx^4+bx^2+a)^2}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^8 + k*x^11)/(a + b*x^2 +
 c*x^4)^3,x]
```

```
[Out] -1/4*(x*(c^2*(a*b*f - b^2*(d + (a^2*j)/c^2) + 2*a*(c*d - a*h + (a^2*j)/c))
+ (2*a*c^3*f - a*b^3*j - b*c*(c^2*d + a*c*h - 3*a^2*j))*x^2))/(a*c^2*(b^2 -
 4*a*c)*(a + b*x^2 + c*x^4)^2) - (b*c^3*(c*e + a*i) - a*b^4*k + 4*a^2*b^2*c
*k - 2*a*c^2*(c^2*g + a^2*k) + (2*c^5*e + b^2*c^3*i - c^4*(b*g + 2*a*i) - b
^5*k + 5*a*b^3*c*k - 5*a^2*b*c^2*k)*x^2)/(4*c^4*(b^2 - 4*a*c)*(a + b*x^2 +
c*x^4)^2) + (x*(c*(a*b^3*f + 8*a^2*b*c*f + 4*a^2*(7*c^2*d + a*c*h - 9*a^2*j
) + b^4*(3*d - (2*a^2*j)/c^2) - a*b^2*(25*c*d + 7*a*h - (11*a^2*j)/c)) + (a
*b^2*c^2*f + 20*a^2*c^3*f + b^3*(3*c^2*d + a^2*j) - 4*a*b*c*(6*c^2*d + 3*a*
c*h + 4*a^2*j))*x^2))/(8*a^2*c*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (b^3*
c^2*i + 2*b*c^3*(3*c*e + a*i) + 11*a*b^4*k - (b^6*k)/c + 32*a^3*c^2*k - 3*b
^2*(c^3*g + 13*a^2*c*k) + 2*(6*c^5*e + b^2*c^3*i - c^4*(3*b*g - 2*a*i) + 2*
b^5*k - 15*a*b^3*c*k + 25*a^2*b*c^2*k)*x^2)/(4*c^3*(b^2 - 4*a*c)^2*(a + b*x
^2 + c*x^4)) + ((a*b^2*c*f + 20*a^2*c^2*f - 4*a*b*(6*c^2*d + 3*a*c*h + 4*a^
2*j) + b^3*(3*c*d + (a^2*j)/c) + (a*b^3*c^2*f - 52*a^2*b*c^3*f - 6*a*b^2*c*
(5*c^2*d - 3*a*c*h - 3*a^2*j) + b^4*(3*c^2*d - a^2*j) + 8*a^2*c^2*(21*c^2*d
 + 3*a*c*h + 5*a^2*j))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sq
```

```
rt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*Sqrt[c]*(b^2 - 4*a*c)^2*Sqrt[b -
 Sqrt[b^2 - 4*a*c]]) + ((a*b^2*c*f + 20*a^2*c^2*f - 4*a*b*(6*c^2*d + 3*a*c*
h + 4*a^2*j) + b^3*(3*c*d + (a^2*j)/c) - (a*b^3*c^2*f - 52*a^2*b*c^3*f - 6*
a*b^2*c*(5*c^2*d - 3*a*c*h - 3*a^2*j) + b^4*(3*c^2*d - a^2*j) + 8*a^2*c^2*(
21*c^2*d + 3*a*c*h + 5*a^2*j))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(Sqrt[2]*Sqrt[
c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*Sqrt[c]*(b^2 - 4*a*c)^2*
Sqrt[b + Sqrt[b^2 - 4*a*c]]) - ((12*c^5*e + 2*b^2*c^3*i - c^4*(6*b*g - 4*a*
i) - b^5*k + 10*a*b^3*c*k - 30*a^2*b*c^2*k)*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2
- 4*a*c]])/(2*c^3*(b^2 - 4*a*c)^(5/2)) + (k*Log[a + b*x^2 + c*x^4])/(4*c^3)
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 642

```
Int[((d_) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> S
imp[d*(Log[RemoveContent[a + b*x + c*x^2, x]]/b), x] /; FreeQ[{a, b, c, d,
e}, x] && EqQ[2*c*d - b*e, 0]
```

Rule 648

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> D
ist[(2*c*d - b*e)/(2*c), Int[1/(a + b*x + c*x^2), x], x] + Dist[e/(2*c), In
t[(b + 2*c*x)/(a + b*x + c*x^2), x], x] /; FreeQ[{a, b, c, d, e}, x] && NeQ
[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] &&  !NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1674

```
Int[(Pq_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> With[{Q =
PolynomialQuotient[Pq, a + b*x + c*x^2, x], f = Coeff[PolynomialRemainder[P
q, a + b*x + c*x^2, x], x, 0], g = Coeff[PolynomialRemainder[Pq, a + b*x +
c*x^2, x], x, 1]}, Simp[(b*f - 2*a*g + (2*c*f - b*g)*x)*((a + b*x + c*x^2)^
(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] + Dist[1/((p + 1)*(b^2 - 4*a*c)), Int[
(a + b*x + c*x^2)^(p + 1)*ExpandToSum[(p + 1)*(b^2 - 4*a*c)*Q - (2*p + 3)*(
2*c*f - b*g), x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x] && NeQ[b^2
 - 4*a*c, 0] && LtQ[p, -1]
```

Rule 1677

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :
> Dist[1/2, Subst[Int[x^((m - 1)/2)*SubstFor[x^2, Pq, x]*(a + b*x + c*x^2)^
p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x^2] && IntegerQ[
(m - 1)/2]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rule 1692

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> With[{d =
 Coeff[PolynomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 0], e = Coeff[Poly
nomialRemainder[Pq, a + b*x^2 + c*x^4, x], x, 2]}, Simp[x*(a + b*x^2 + c*x^
4)^(p + 1)*((a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)/(2*a*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2 - 4*a*c)), Int[(a + b*x^2 + c*
x^4)^(p + 1)*ExpandToSum[2*a*(p + 1)*(b^2 - 4*a*c)*PolynomialQuotient[Pq, a
 + b*x^2 + c*x^4, x] + b^2*d*(2*p + 3) - 2*a*c*d*(4*p + 5) - a*b*e + c*(4*p
 + 7)*(b*d - 2*a*e)*x^2, x], x], x]] /; FreeQ[{a, b, c}, x] && PolyQ[Pq, x^
2] && Expon[Pq, x^2] > 1 && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2+hx^4+jx^8}{(a+bx^2+cx^4)^3}\,dx + \int \frac{x(e+gx^2+ix^4+kx^{10})}{(a+bx^2+cx^4)^3}\,dx$$

$$= -\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+\left(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j)\right)x^2\right)}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{1}{2}\text{Subst}\left(\int\frac{e+gx+ix^2+kx^5}{\left(a+bx+cx^2\right)^3}\,dx,x,x^2\right) - \frac{\int\frac{-abf-b^2\left(3d-\frac{a^2j}{c^2}\right)+2a\left(7cd+ah-\frac{a^2j}{c}\right)+\frac{\left(10ac^3f-ab^3j-bc\left(5c^2d+5ach+a^2j\right)\right)x^2}{c^2}+4a\left(4a-\frac{b^2}{c}\right)jx^4}{\left(a+bx^2+cx^4\right)^2}\,dx}{4a\left(b^2-4ac\right)}$$

$$= -\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+\left(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j)\right)x^2\right)}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} - \frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck}{4c^4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} + \frac{x\Big(c\Big(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\Big)+(}{8a^2c\left(b^2-4ac\right)^2\left(a+bx^2+cx\right.} + \frac{\int\frac{3b^4d+ab^3f-16a^2bcf+4a^2\left(21c^2d+3ach+5a^2j\right)-ab^2\left(27cd-3ah-\frac{a^2j}{c}\right)+\frac{\left(ab^2c^2f+20a^2c^3f+b^3\left(3c^2d+a^2j\right)-4abc\left(6c^2d+3ach+4a^2j\right)\right.}{c}}{a+bx^2+cx^4}}{8a^2\left(b^2-4ac\right)^2} - \frac{\text{Subst}\left(\int\frac{6ce-3bg+2ai+\frac{b^2i}{c}-\frac{b^5k}{c^4}+\frac{3ab^3k}{c^3}+\frac{a^2bk}{c^2}-\frac{2\left(b^4-5ab^2c+4a^2c^2\right)kx}{c^3}+\frac{2b\left(b^2-4ac\right)kx^2}{c^2}+2\left(4a-\frac{b^2}{c}\right)kx^3}{\left(a+bx+cx^2\right)^2}\,dx,x,x^2\right)}{4\left(b^2-4ac\right)}$$

$$= -\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j))x^2\right)}{4ac^2(b^2-4ac)(a+bx^2+cx^4)^2}$$

$$-\frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck}{4c^4(b^2-4ac)(a+bx^2+cx^4)^2}$$

$$+\frac{x\left(c\left(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\right)+\right.}{8a^2c(b^2-4ac)^2(a+bx^2+c}$$

$$+\frac{b^3c^2i+2bc^3(3ce+ai)+11ab^4k-\frac{b^6k}{c}+32a^3c^2k-3b^2(c^3g+13a^2ck)+2(6c^5e+b^2c^3i-c^4(3bg}{4c^3(b^2-4ac)^2(a+bx^2+cx^4)}$$

$$+\frac{\text{Subst}\left(\int\frac{2\left(6c^2e-3bcg+b^2i+2aci+\frac{ab^3k}{c^2}-\frac{7a^2bk}{c}\right)+\frac{2(b^2-4ac)^2kx}{c^2}}{a+bx+cx^2}\,dx,x,x^2\right)}{4(b^2-4ac)^2}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)-\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3}{}\right.}{16a^2(b^2-4ac)^2}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)+\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3}{}\right.}{16a^2(b^2-4ac)^2}$$

$$= -\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j))x^2\right)}{4ac^2(b^2-4ac)(a+bx^2+cx^4)^2}$$

$$-\frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck}{4c^4(b^2-4ac)(a+bx^2+cx^4)^2}$$

$$+\frac{x\left(c\left(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\right)+\right.}{8a^2c(b^2-4ac)^2(a+bx^2+c}$$

$$+\frac{b^3c^2i+2bc^3(3ce+ai)+11ab^4k-\frac{b^6k}{c}+32a^3c^2k-3b^2(c^3g+13a^2ck)+2(6c^5e+b^2c^3i-c^4(3bg}{4c^3(b^2-4ac)^2(a+bx^2+cx^4)}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)+\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3}{}\right.}{8\sqrt{2}a^2\sqrt{c}(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4a}}}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)-\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3}{}\right.}{8\sqrt{2}a^2\sqrt{c}(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4a}}}$$

$$+\frac{k\text{Subst}\left(\int\frac{b+2cx}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^3}$$

$$+\frac{(12c^5e+2b^2c^3i-c^4(6bg-4ai)-b^5k+10ab^3ck-30a^2bc^2k)\,\text{Subst}\left(\int\frac{1}{a+bx+cx^2}\,dx,x,x^2\right)}{4c^3(b^2-4ac)^2}$$

$$=$$

$$-\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+\left(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j)\right)x^2\right)}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$-\frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+\left(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck\right.}{4c^4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{x\left(c\left(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\right)+(\right.}{8a^2c\left(b^2-4ac\right)^2\left(a+bx^2+cx\right.}$$

$$+\frac{b^3c^2i+2bc^3(3ce+ai)+11ab^4k-\frac{b^6k}{c}+32a^3c^2k-3b^2(c^3g+13a^2ck)+2(6c^5e+b^2c^3i-c^4(3bg}{4c^3\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)+\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a}{c\sqrt{}}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)-\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a}{c\sqrt{}}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$

$$+\frac{k\log\left(a+bx^2+cx^4\right)}{4c^3}$$

$$-\frac{\left(12c^5e+2b^2c^3i-c^4(6bg-4ai)-b^5k+10ab^3ck-30a^2bc^2k\right)\text{Subst}\left(\int\frac{1}{b^2-4ac-x^2}\,dx,x,b+2cx^2\right)}{2c^3\left(b^2-4ac\right)^2}$$

$$=$$

$$-\frac{x\left(c^2\left(abf-b^2\left(d+\frac{a^2j}{c^2}\right)+2a\left(cd-ah+\frac{a^2j}{c}\right)\right)+\left(2ac^3f-ab^3j-bc(c^2d+ach-3a^2j)\right)x^2\right)}{4ac^2\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$-\frac{bc^3(ce+ai)-ab^4k+4a^2b^2ck-2ac^2(c^2g+a^2k)+\left(2c^5e+b^2c^3i-c^4(bg+2ai)-b^5k+5ab^3ck\right.}{4c^4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{x\left(c\left(ab^3f+8a^2bcf+4a^2(7c^2d+ach-9a^2j)+b^4\left(3d-\frac{2a^2j}{c^2}\right)-ab^2\left(25cd+7ah-\frac{11a^2j}{c}\right)\right)+(\right.}{8a^2c\left(b^2-4ac\right)^2\left(a+bx^2+cx\right.}$$

$$+\frac{b^3c^2i+2bc^3(3ce+ai)+11ab^4k-\frac{b^6k}{c}+32a^3c^2k-3b^2(c^3g+13a^2ck)+2(6c^5e+b^2c^3i-c^4(3bg}{4c^3\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)+\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a}{c\sqrt{}}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(ab^2cf+20a^2c^2f-4ab(6c^2d+3ach+4a^2j)+b^3\left(3cd+\frac{a^2j}{c}\right)-\frac{ab^3c^2f-52a^2bc^3f-6ab^2c(5c^2d-3ach-3a}{c\sqrt{}}\right.}{8\sqrt{2}a^2\sqrt{c}\left(b^2-4ac\right)^2\sqrt{b+\sqrt{b^2-4ac}}}$$

$$-\frac{\left(12c^5e+2b^2c^3i-c^4(6bg-4ai)-b^5k+10ab^3ck-30a^2bc^2k\right)\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{2c^3\left(b^2-4ac\right)^{5/2}}$$

$$+\frac{k\log\left(a+bx^2+cx^4\right)}{4c^3}$$

Mathematica [A] (verified)

Time = 6.80 (sec) , antiderivative size = 1649, normalized size of antiderivative = 1.40

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + ix^5 + jx^8 + kx^{11}}{(a + bx^2 + cx^4)^3} \, dx$$
$$= \frac{abc^4e - 2a^2c^4g + a^2bc^3i - a^2b^4k + 4a^3b^2ck - 2a^4c^2k - b^2c^4dx + 2ac^5dx + abc^4fx - 2a^2c^4hx - a^2b^2c^2jx}{}$$
$$+ \frac{12a^2bc^5e - 6a^2b^2c^4g + 2a^2b^3c^3i + 4a^3bc^4i - 2a^2b^6k + 22a^3b^4ck - 78a^4b^2c^2k + 64a^5c^3k + 3b^4c^4dx - 25a}{}$$
$$+ \frac{\left(3b^4c^2d - 30ab^2c^3d + 168a^2c^4d + 3b^3c^2\sqrt{b^2 - 4ac}d - 24abc^3\sqrt{b^2 - 4ac}d + ab^3c^2f - 52a^2bc^3f + ab^2c^2\sqrt{}\right.}{}$$
$$+ \frac{\left(-3b^4c^2d + 30ab^2c^3d - 168a^2c^4d + 3b^3c^2\sqrt{b^2 - 4ac}d - 24abc^3\sqrt{b^2 - 4ac}d - ab^3c^2f + 52a^2bc^3f + ab^2c^2\right.}{}$$
$$+ \frac{\left(12c^5e - 6bc^4g + 2b^2c^3i + 4ac^4i - b^5k + 10ab^3ck - 30a^2bc^2k + b^4\sqrt{b^2 - 4ac}k - 8ab^2c\sqrt{b^2 - 4ac}k + 16a\right.}{4c^3 (b^2 - 4ac)^{5/2}}$$
$$+ \frac{\left(-12c^5e + 6bc^4g - 2b^2c^3i - 4ac^4i + b^5k - 10ab^3ck + 30a^2bc^2k + b^4\sqrt{b^2 - 4ac}k - 8ab^2c\sqrt{b^2 - 4ac}k + 1\right.}{4c^3 (b^2 - 4ac)^{5/2}}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^8 + k*x^11)/(a + b
*x^2 + c*x^4)^3,x]
```

```
[Out] (a*b*c^4*e - 2*a^2*c^4*g + a^2*b*c^3*i - a^2*b^4*k + 4*a^3*b^2*c*k - 2*a^4*
c^2*k - b^2*c^4*d*x + 2*a*c^5*d*x + a*b*c^4*f*x - 2*a^2*c^4*h*x - a^2*b^2*c
^2*j*x + 2*a^3*c^3*j*x + 2*a*c^5*e*x^2 - a*b*c^4*g*x^2 + a*b^2*c^3*i*x^2 -
2*a^2*c^4*i*x^2 - a*b^5*k*x^2 + 5*a^2*b^3*c*k*x^2 - 5*a^3*b*c^2*k*x^2 - b*c
^5*d*x^3 + 2*a*c^5*f*x^3 - a*b*c^4*h*x^3 - a*b^3*c^2*j*x^3 + 3*a^2*b*c^3*j*
x^3)/(4*a*c^4*(-b^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (12*a^2*b*c^5*e - 6*a
^2*b^2*c^4*g + 2*a^2*b^3*c^3*i + 4*a^3*b*c^4*i - 2*a^2*b^6*k + 22*a^3*b^4*c
*k - 78*a^4*b^2*c^2*k + 64*a^5*c^3*k + 3*b^4*c^4*d*x - 25*a*b^2*c^5*d*x + 2
8*a^2*c^6*d*x + a*b^3*c^4*f*x + 8*a^2*b*c^5*f*x - 7*a^2*b^2*c^4*h*x + 4*a^3
*c^5*h*x - 2*a^2*b^4*c^2*j*x + 11*a^3*b^2*c^3*j*x - 36*a^4*c^4*j*x + 24*a^2
*c^6*e*x^2 - 12*a^2*b*c^5*g*x^2 + 4*a^2*b^2*c^4*i*x^2 + 8*a^3*c^5*i*x^2 + 8
*a^2*b^5*c*k*x^2 - 60*a^3*b^3*c^2*k*x^2 + 100*a^4*b*c^3*k*x^2 + 3*b^3*c^5*d
*x^3 - 24*a*b*c^6*d*x^3 + a*b^2*c^5*f*x^3 + 20*a^2*c^6*f*x^3 - 12*a^2*b*c^5
*h*x^3 + a^2*b^3*c^3*j*x^3 - 16*a^3*b*c^4*j*x^3)/(8*a^2*c^4*(-b^2 + 4*a*c)^
2*(a + b*x^2 + c*x^4)) + ((3*b^4*c^2*d - 30*a*b^2*c^3*d + 168*a^2*c^4*d + 3
*b^3*c^2*Sqrt[b^2 - 4*a*c]*d - 24*a*b*c^3*Sqrt[b^2 - 4*a*c]*d + a*b^3*c^2*f
 - 52*a^2*b*c^3*f + a*b^2*c^2*Sqrt[b^2 - 4*a*c]*f + 20*a^2*c^3*Sqrt[b^2 - 4
*a*c]*f + 18*a^2*b^2*c^2*h + 24*a^3*c^3*h - 12*a^2*b*c^2*Sqrt[b^2 - 4*a*c]*
h - a^2*b^4*j + 18*a^3*b^2*c*j + 40*a^4*c^2*j + a^2*b^3*Sqrt[b^2 - 4*a*c]*j
```

```
 - 16*a^3*b*c*Sqrt[b^2 - 4*a*c]*j)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt
[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*c^(3/2)*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b
^2 - 4*a*c]]) + ((-3*b^4*c^2*d + 30*a*b^2*c^3*d - 168*a^2*c^4*d + 3*b^3*c^2
*Sqrt[b^2 - 4*a*c]*d - 24*a*b*c^3*Sqrt[b^2 - 4*a*c]*d - a*b^3*c^2*f + 52*a^
2*b*c^3*f + a*b^2*c^2*Sqrt[b^2 - 4*a*c]*f + 20*a^2*c^3*Sqrt[b^2 - 4*a*c]*f
- 18*a^2*b^2*c^2*h - 24*a^3*c^3*h - 12*a^2*b*c^2*Sqrt[b^2 - 4*a*c]*h + a^2*
b^4*j - 18*a^3*b^2*c*j - 40*a^4*c^2*j + a^2*b^3*Sqrt[b^2 - 4*a*c]*j - 16*a^
3*b*c*Sqrt[b^2 - 4*a*c]*j)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4
*a*c]]])/(8*Sqrt[2]*a^2*c^(3/2)*(b^2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a
*c]]) + ((12*c^5*e - 6*b*c^4*g + 2*b^2*c^3*i + 4*a*c^4*i - b^5*k + 10*a*b^3
*c*k - 30*a^2*b*c^2*k + b^4*Sqrt[b^2 - 4*a*c]*k - 8*a*b^2*c*Sqrt[b^2 - 4*a*
c]*k + 16*a^2*c^2*Sqrt[b^2 - 4*a*c]*k)*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2
])/(4*c^3*(b^2 - 4*a*c)^(5/2)) + ((-12*c^5*e + 6*b*c^4*g - 2*b^2*c^3*i - 4*
a*c^4*i + b^5*k - 10*a*b^3*c*k + 30*a^2*b*c^2*k + b^4*Sqrt[b^2 - 4*a*c]*k -
 8*a*b^2*c*Sqrt[b^2 - 4*a*c]*k + 16*a^2*c^2*Sqrt[b^2 - 4*a*c]*k)*Log[b + Sq
rt[b^2 - 4*a*c] + 2*c*x^2])/(4*c^3*(b^2 - 4*a*c)^(5/2))
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 1.14 (sec) , antiderivative size = 1182, normalized size of antiderivative = 1.00

method	result	size
risch	Expression too large to display	1182
default	Expression too large to display	2058

```
[In] int((k*x^11+j*x^8+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,method
=_RETURNVERBOSE)
```

```
[Out] (-1/8*(16*a^3*b*c*j-a^2*b^3*j+12*a^2*b*c^2*h-20*a^2*c^3*f-a*b^2*c^2*f+24*a*
b*c^3*d-3*b^3*c^2*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^7+1/2*(25*a^2*b*c^2*k
-15*a*b^3*c*k+2*a*c^4*i+2*b^5*k+b^2*c^3*i-3*b*c^4*g+6*c^5*e)/c^2/(16*a^2*c^
2-8*a*b^2*c+b^4)*x^6-1/8/a^2*(36*a^4*c^2*j+5*a^3*b^2*c*j-4*a^3*c^3*h+a^2*b^
4*j+19*a^2*b^2*c^2*h-28*a^2*b*c^3*f-28*a^2*c^4*d-2*a*b^3*c^2*f+49*a*b^2*c^3
*d-6*b^4*c^2*d)/(16*a^2*c^2-8*a*b^2*c+b^4)/c*x^5+1/4*(32*a^3*c^3*k+11*a^2*b
^2*c^2*k-19*a*b^4*c*k+6*a*b*c^4*i+3*b^6*k+3*b^3*c^3*i-9*b^2*c^4*g+18*b*c^5*
e)/(16*a^2*c^2-8*a*b^2*c+b^4)/c^3*x^4-1/8/c*(28*a^4*b*c*j+2*a^3*b^3*j+16*a^
3*b*c^2*h-36*a^3*c^3*f+5*a^2*b^3*c*h-5*a^2*b^2*c^2*f+4*a^2*b*c^3*d-a*b^4*c*
f+20*a*b^3*c^2*d-3*b^5*c*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^3+1/2/c^3*(31*
a^3*b*c^2*k-22*a^2*b^3*c*k-2*a^2*c^4*i+3*a*b^5*k+5*a*b^2*c^3*i-5*a*b*c^4*g+
10*a*c^5*e-b^3*c^3*g+2*b^2*c^4*e)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2-1/8*(20*a^
4*c*j+a^3*b^2*j+12*a^3*c^2*h+3*a^2*b^2*c*h-16*a^2*b*c^2*f-44*a^2*c^3*d+a*b^
3*c*f+37*a*b^2*c^2*d-5*b^4*c*d)/c/(16*a^2*c^2-8*a*b^2*c+b^4)/a*x+1/4*(24*a^
4*c^2*k-21*a^3*b^2*c*k+3*a^2*b^4*k+6*a^2*b*c^3*i-8*a^2*c^4*g-a*b^2*c^3*g+10
```

```
*a*b*c^4*e-b^3*c^3*e)/(16*a^2*c^2-8*a*b^2*c+b^4)/c^3)/(c*x^4+b*x^2+a)^2+1/1
6/c*sum((8/c*k*_R^3-1/a^2*(16*a^3*b*c*j-a^2*b^3*j+12*a^2*b*c^2*h-20*a^2*c^3
*f-a*b^2*c^2*f+24*a*b*c^3*d-3*b^3*c^2*d)/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2-8/
c*(7*a^2*b*c*k-a*b^3*k-2*a*c^3*i-b^2*c^2*i+3*b*c^3*g-6*c^4*e)/(16*a^2*c^2-8
*a*b^2*c+b^4)*_R+1/a^2*(20*a^4*c*j+a^3*b^2*j+12*a^3*c^2*h+3*a^2*b^2*c*h-16*
a^2*b*c^2*f+84*a^2*c^3*d+a*b^3*c*f-27*a*b^2*c^2*d+3*b^4*c*d)/(16*a^2*c^2-8*
a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx=\text{Timed out}$$

```
[In] integrate((k*x^11+j*x^8+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,
 algorithm="fricas")
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx=\text{Timed out}$$

```
[In] integrate((k*x**11+j*x**8+i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2
+a)**3,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx = \int \frac{kx^{11}+jx^8+ix^5+hx^4+gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^3}\,dx$$

```
[In] integrate((k*x^11+j*x^8+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,
 algorithm="maxima")
[Out] 1/8*(12*a^4*b*c^3*i - (12*a^2*b*c^5*h - 3*(b^3*c^5 - 8*a*b*c^6)*d - (a*b^2*
c^5 + 20*a^2*c^6)*f - (a^2*b^3*c^3 - 16*a^3*b*c^4)*j)*x^7 + 4*(6*a^2*c^6*e
```

```
- 3*a^2*b*c^5*g + (a^2*b^2*c^4 + 2*a^3*c^5)*i + (2*a^2*b^5*c - 15*a^3*b^3*c
^2 + 25*a^4*b*c^3)*k)*x^6 + ((6*b^4*c^4 - 49*a*b^2*c^5 + 28*a^2*c^6)*d + 2*
(a*b^3*c^4 + 14*a^2*b*c^5)*f - (19*a^2*b^2*c^4 - 4*a^3*c^5)*h - (a^2*b^4*c^
2 + 5*a^3*b^2*c^3 + 36*a^4*c^4)*j)*x^5 + 2*(18*a^2*b*c^5*e - 9*a^2*b^2*c^4*
g + 3*(a^2*b^3*c^3 + 2*a^3*b*c^4)*i + (3*a^2*b^6 - 19*a^3*b^4*c + 11*a^4*b^
2*c^2 + 32*a^5*c^3)*k)*x^4 + ((3*b^5*c^3 - 20*a*b^3*c^4 - 4*a^2*b*c^5)*d +
(a*b^4*c^3 + 5*a^2*b^2*c^4 + 36*a^3*c^5)*f - (5*a^2*b^3*c^3 + 16*a^3*b*c^4)
*h - 2*(a^3*b^3*c^2 + 14*a^4*b*c^3)*j)*x^3 + 4*(2*(a^2*b^2*c^4 + 5*a^3*c^5)
*e - (a^2*b^3*c^3 + 5*a^3*b*c^4)*g + (5*a^3*b^2*c^3 - 2*a^4*c^4)*i + (3*a^3
*b^5 - 22*a^4*b^3*c + 31*a^5*b*c^2)*k)*x^2 - 2*(a^2*b^3*c^3 - 10*a^3*b*c^4)
*e - 2*(a^3*b^2*c^3 + 8*a^4*c^4)*g + 6*(a^4*b^4 - 7*a^5*b^2*c + 8*a^6*c^2)*
k + ((5*a*b^4*c^3 - 37*a^2*b^2*c^4 + 44*a^3*c^5)*d - (a^2*b^3*c^3 - 16*a^3*
b*c^4)*f - 3*(a^3*b^2*c^3 + 4*a^4*c^4)*h - (a^4*b^2*c^2 + 20*a^5*c^3)*j)*x)
/(a^4*b^4*c^3 - 8*a^5*b^2*c^4 + 16*a^6*c^5 + (a^2*b^4*c^5 - 8*a^3*b^2*c^6 +
 16*a^4*c^7)*x^8 + 2*(a^2*b^5*c^4 - 8*a^3*b^3*c^5 + 16*a^4*b*c^6)*x^6 + (a^
2*b^6*c^3 - 6*a^3*b^4*c^4 + 32*a^5*c^6)*x^4 + 2*(a^3*b^5*c^3 - 8*a^4*b^3*c^
4 + 16*a^5*b*c^5)*x^2) + 1/8*integrate((8*(a^2*b^4 - 8*a^3*b^2*c + 16*a^4*c
^2)*k*x^3 - (12*a^2*b*c^3*h - 3*(b^3*c^3 - 8*a*b*c^4)*d - (a*b^2*c^3 + 20*a
^2*c^4)*f - (a^2*b^3*c - 16*a^3*b*c^2)*j)*x^2 + 3*(b^4*c^2 - 9*a*b^2*c^3 +
28*a^2*c^4)*d + (a*b^3*c^2 - 16*a^2*b*c^3)*f + 3*(a^2*b^2*c^2 + 4*a^3*c^3)*
h + (a^3*b^2*c + 20*a^4*c^2)*j + 8*(6*a^2*c^4*e - 3*a^2*b*c^3*g + (a^2*b^2*
c^2 + 2*a^3*c^3)*i + (a^3*b^3 - 7*a^4*b*c)*k)*x)/(c*x^4 + b*x^2 + a), x)/(a
^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 29142 vs. $2(1123) = 2246$.

Time = 4.87 (sec) , antiderivative size = 29142, normalized size of antiderivative = 24.76

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] integrate((k*x^11+j*x^8+i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^3,x,
 algorithm="giac")
```

```
[Out] 1/64*(3*(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 +
256*a^8*c^9)^2*(2*b^5*c^4 - 24*a*b^3*c^5 + 64*a^2*b*c^6 - sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*b^3*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 16*(b^2 - 4*a*c)*a*b*
```

c^5)*d + (a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)^2*(2*a*b^4*c^4 + 32*a^2*b^2*c^5 - 160*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^5 - 2*(b^2 - 4*a*c)*a*b^2*c^4 - 40*(b^2 - 4*a*c)*a^2*c^5)*f - 12*(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)^2*(2*a^2*b^3*c^4 - 8*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b*c^4)*h + (a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)^2*(2*a^2*b^5*c^2 - 40*a^3*b^3*c^3 + 128*a^4*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 32*(b^2 - 4*a*c)*a^3*b*c^3)*j + 6*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^14*c^7 - 29*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^8 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^13*c^8 - 2*a^4*b^14*c^8 + 368*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^9 + 50*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^9 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^12*c^9 + 58*a^5*b^12*c^9 - 2640*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^10 - 536*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^10 - 25*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^10 - 736*a^6*b^10*c^10 + 11520*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^11 + 3136*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^11 + 268*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^8*c^11 + 5280*a^7*b^8*c^11 - 30464*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^12 - 10496*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^12 - 1568*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c^12 - 23040*a^8*b^6*c^12 + 45056*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^13 + 18944*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^13 + 5248*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^13 + 60928*a^9*b^4*c^13 - 28672*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*c^14 - 14336*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^14 - 9472*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^2*c^14 - 90112*a^10*b^2*c^14 + 7168*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*c^15 + 57344*a^11*c^15 + 2*(b^2 - 4*a*c)*a^4*b^12*c^8 - 50*(b^2 - 4*a*c)*a^5*b^10*c^9 + 536*(b^2 - 4*a*c)*a^

6*b^8*c^10 - 3136*(b^2 - 4*a*c)*a^7*b^6*c^11 + 10496*(b^2 - 4*a*c)*a^8*b^4*
c^12 - 18944*(b^2 - 4*a*c)*a^9*b^2*c^13 + 14336*(b^2 - 4*a*c)*a^10*c^14)*d*
abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a
^8*c^9) + 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^13*c^7 - 36*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^8 - 2*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^5*b^12*c^8 - 2*a^5*b^13*c^8 + 480*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^7*b^9*c^9 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^6*b^10*c^9 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^9 + 72*a
^6*b^11*c^9 - 3200*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^10 - 7
04*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^10 - 32*sqrt(2)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^10 - 960*a^7*b^9*c^10 + 11520*sqrt(2)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^11 + 3584*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a^8*b^6*c^11 + 352*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a^7*b^7*c^11 + 6400*a^8*b^7*c^11 - 21504*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^10*b^3*c^12 - 8704*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^4*
c^12 - 1792*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^12 - 23040*a^
9*b^5*c^12 + 16384*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b*c^13 + 81
92*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^13 + 4352*sqrt(2)*sqr
t(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^13 + 43008*a^10*b^3*c^13 - 4096*sqrt
(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b*c^14 - 32768*a^11*b*c^14 + 2*(b^
2 - 4*a*c)*a^5*b^11*c^8 - 64*(b^2 - 4*a*c)*a^6*b^9*c^9 + 704*(b^2 - 4*a*c)*
a^7*b^7*c^10 - 3584*(b^2 - 4*a*c)*a^8*b^5*c^11 + 8704*(b^2 - 4*a*c)*a^9*b^3
c^12 - 8192(b^2 - 4*a*c)*a^10*b*c^13)*f*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6
+ 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) + 6*(sqrt(2)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^6*b^12*c^7 - 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^7*b^10*c^8 - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^8 -
2*a^6*b^12*c^8 + 80*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^8*c^9 + 2
4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^9 + sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^9 + 32*a^7*b^10*c^9 - 64*sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^10 - 12*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^7*b^8*c^10 - 160*a^8*b^8*c^10 - 1280*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^10*b^4*c^11 - 256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^
5*c^11 + 32*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^11 + 4096*sqr
t(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^2*c^12 + 1536*sqrt(2)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^10*b^3*c^12 + 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^9*b^4*c^12 + 2560*a^10*b^4*c^12 - 4096*sqrt(2)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^12*c^13 - 2048*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11
*b*c^13 - 768*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^13 - 8192*
a^11*b^2*c^13 + 1024*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*c^14 + 81
92*a^12*c^14 + 2*(b^2 - 4*a*c)*a^6*b^10*c^8 - 24*(b^2 - 4*a*c)*a^7*b^8*c^9
+ 64*(b^2 - 4*a*c)*a^8*b^6*c^10 + 256*(b^2 - 4*a*c)*a^9*b^4*c^11 - 1536*(b^
2 - 4*a*c)*a^10*b^2*c^12 + 2048*(b^2 - 4*a*c)*a^11*c^13)*h*abs(a^4*b^8*c^5
- 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) + 2*(sqr
t(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^12*c^6 - 2*sqrt(2)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^7*b^11*c^7 - 2*a^7*b^12*c^7 - 240*sqrt(2)*sqrt(b*c + s

qrt(b^2 - 4*a*c)*c)*a^9*b^8*c^8 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^8*b^9*c^8 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^7*b^10*c^8 + 2560*
sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^6*c^9 + 448*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^9*b^7*c^9 + 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c
)*c)*a^8*b^8*c^9 + 480*a^9*b^8*c^9 - 11520*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^11*b^4*c^10 - 3328*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b
^5*c^10 - 224*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^6*c^10 - 5120*a
^10*b^6*c^10 + 24576*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^2*c^11
 + 9728*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^3*c^11 + 1664*sqrt(2)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^4*c^11 + 23040*a^11*b^4*c^11 - 2048
0*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*c^12 - 10240*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^12*b*c^12 - 4864*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^11*b^2*c^12 - 49152*a^12*b^2*c^12 + 5120*sqrt(2)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^12*c^13 + 40960*a^13*c^13 + 2*(b^2 - 4*a*c)*a^7*b^10*c^7
 + 8*(b^2 - 4*a*c)*a^8*b^8*c^8 - 448*(b^2 - 4*a*c)*a^9*b^6*c^9 + 3328*(b^2
 - 4*a*c)*a^10*b^4*c^10 - 9728*(b^2 - 4*a*c)*a^11*b^2*c^11 + 10240*(b^2 - 4*
a*c)*a^12*c^12)*j*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a
^7*b^2*c^8 + 256*a^8*c^9) + 3*(2*a^8*b^21*c^14 - 84*a^9*b^19*c^15 + 1648*a^
10*b^17*c^16 - 19712*a^11*b^15*c^17 + 157696*a^12*b^13*c^18 - 874496*a^13*b
^11*c^19 + 3383296*a^14*b^9*c^20 - 8978432*a^15*b^7*c^21 + 15597568*a^16*b^
5*c^22 - 15990784*a^17*b^3*c^23 + 7340032*a^18*b*c^24 - sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^21*c^12 + 42*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^19*c^13 + 2*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^20*c^13 - 824*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 - 76*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^18*c^14 - sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^8*b^19*c^14 + 9856*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^15*c^15 + 1344*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^16*c^15 + 38
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^17*c^15 -
78848*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^13*c
^16 - 14336*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*
b^14*c^16 - 672*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^10*b^15*c^16 + 437248*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^13*b^11*c^17 + 100352*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^12*b^12*c^17 + 7168*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^11*b^13*c^17 - 1691648*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^9*c^18 - 473088*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^10*c^18 - 50176*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^11*c^18 + 4489216*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^7*c^19 + 1490944*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^8*c^19 + 236544
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^9*c^19 -
7798784*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^16*b^5*
c^20 - 3014656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^

15*b^6*c^20 - 745472*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^14*b^7*c^20 + 7995392*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^17*b^3*c^21 + 3538944*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^16*b^4*c^21 + 1507328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^15*b^5*c^21 - 3670016*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b*c^22 - 1835008*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^2*c^22 - 1769472*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^16*b^3*c^22 + 917504*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b*c^23 - 2*(b^2 - 4*
a*c)*a^8*b^19*c^14 + 76*(b^2 - 4*a*c)*a^9*b^17*c^15 - 1344*(b^2 - 4*a*c)*a^
10*b^15*c^16 + 14336*(b^2 - 4*a*c)*a^11*b^13*c^17 - 100352*(b^2 - 4*a*c)*a^
12*b^11*c^18 + 473088*(b^2 - 4*a*c)*a^13*b^9*c^19 - 1490944*(b^2 - 4*a*c)*a
^14*b^7*c^20 + 3014656*(b^2 - 4*a*c)*a^15*b^5*c^21 - 3538944*(b^2 - 4*a*c)*
a^16*b^3*c^22 + 1835008*(b^2 - 4*a*c)*a^17*b*c^23)*d + (2*a^9*b^20*c^14 - 1
68*a^10*b^18*c^15 + 4224*a^11*b^16*c^16 - 53760*a^12*b^14*c^17 + 408576*a^1
3*b^12*c^18 - 1978368*a^14*b^10*c^19 + 6193152*a^15*b^8*c^20 - 12189696*a^1
6*b^6*c^21 + 13762560*a^17*b^4*c^22 - 6815744*a^18*b^2*c^23 - sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^20*c^12 + 84*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^18*c^13 + 2*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^19*c^13 - 2112*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^16*c^14 - 160*s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^9*b^18*c^14 + 26
880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^14*c^1
5 + 3584*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^1
5*c^15 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*
b^16*c^15 - 204288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c
)*a^13*b^12*c^16 - 39424*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^12*b^13*c^16 - 1792*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^11*b^14*c^16 + 989184*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^14*b^10*c^17 + 250880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^11*c^17 + 19712*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^12*c^17 - 3096576*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^8*c^18 - 974848*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^9*c^18 - 125440*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^10*c^18 + 60948
48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^16*b^6*c^19
+ 2293760*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^
7*c^19 + 487424*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^14*b^8*c^19 - 6881280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^17*b^4*c^20 - 3014656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^16*b^5*c^20 - 1146880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^15*b^6*c^20 + 3407872*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b^2*c^21 + 1703936*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^3*c^21 + 1507328*sqrt(2)*sqrt(b^2

- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^16*b^4*c^21 - 851968*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^2*c^22 - 2*(b^2 - 4*a
*c)*a^9*b^18*c^14 + 160*(b^2 - 4*a*c)*a^10*b^16*c^15 - 3584*(b^2 - 4*a*c)*a
^11*b^14*c^16 + 39424*(b^2 - 4*a*c)*a^12*b^12*c^17 - 250880*(b^2 - 4*a*c)*a
^13*b^10*c^18 + 974848*(b^2 - 4*a*c)*a^14*b^8*c^19 - 2293760*(b^2 - 4*a*c)*
a^15*b^6*c^20 + 3014656*(b^2 - 4*a*c)*a^16*b^4*c^21 - 1703936*(b^2 - 4*a*c)
*a^17*b^2*c^22)*f + 6*(6*a^10*b^19*c^14 - 184*a^11*b^17*c^15 + 2432*a^12*b^
15*c^16 - 17920*a^13*b^13*c^17 + 78848*a^14*b^11*c^18 - 200704*a^15*b^9*c^1
9 + 229376*a^16*b^7*c^20 + 131072*a^17*b^5*c^21 - 655360*a^18*b^3*c^22 + 52
4288*a^19*b*c^23 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)
*c)*a^10*b^19*c^12 + 92*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^11*b^17*c^13 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a^10*b^18*c^13 - 1216*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a^12*b^15*c^14 - 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^11*b^16*c^14 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + s
qrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 + 8960*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^13*b^13*c^15 + 1792*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^14*c^15 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^15*c^15 - 39424*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^11*c^16 - 10752*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^12*c^16 - 896*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^13*c^16 + 100352*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^9*c^17 + 358
40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^10*c^17
 + 5376*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^11
*c^17 - 114688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^
16*b^7*c^18 - 57344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^15*b^8*c^18 - 17920*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^14*b^9*c^18 - 65536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a^17*b^5*c^19 + 28672*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^15*b^7*c^19 + 327680*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^18*b^3*c^20 + 131072*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^4*c^20 - 262144*sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^19*b*c^21 - 131072*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b^2*c^21 - 65536*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^3*c^21 + 65536*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b*c^22 - 6*(b^2 - 4*
a*c)*a^10*b^17*c^14 + 160*(b^2 - 4*a*c)*a^11*b^15*c^15 - 1792*(b^2 - 4*a*c)
*a^12*b^13*c^16 + 10752*(b^2 - 4*a*c)*a^13*b^11*c^17 - 35840*(b^2 - 4*a*c)*
a^14*b^9*c^18 + 57344*(b^2 - 4*a*c)*a^15*b^7*c^19 - 131072*(b^2 - 4*a*c)*a^
17*b^3*c^21 + 131072*(b^2 - 4*a*c)*a^18*b*c^22)*h - (2*a^10*b^21*c^12 - 100
*a^11*b^19*c^13 + 1968*a^12*b^17*c^14 - 20736*a^13*b^15*c^15 + 129024*a^14*
b^13*c^16 - 473088*a^15*b^11*c^17 + 860160*a^16*b^9*c^18 + 196608*a^17*b^7*
c^19 - 4325376*a^18*b^5*c^20 + 8126464*a^19*b^3*c^21 - 5242880*a^20*b*c^22
- sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^21*c^10

+ 50*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^11*b^19*c^
11 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^10*b^20*
c^11 - 984*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b
^17*c^12 - 92*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^1
1*b^18*c^12 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^1
0*b^19*c^12 + 10368*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a^13*b^15*c^13 + 1600*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^12*b^16*c^13 + 46*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^11*b^17*c^13 - 64512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt
(b^2 - 4*a*c)*c)*a^14*b^13*c^14 - 14336*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^13*b^14*c^14 - 800*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^12*b^15*c^14 + 236544*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^11*c^15 + 71680*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^12*c^15 + 7168*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^13*b^13*c^15 - 430080*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^16*b^9*c^16 - 186368*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^15*b^10*c^16 -
35840*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^14*b^11*c
^16 - 98304*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*
b^7*c^17 + 114688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)
*a^16*b^8*c^17 + 93184*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a^15*b^9*c^17 + 2162688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^18*b^5*c^18 + 655360*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*a^17*b^6*c^18 - 57344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^16*b^7*c^18 - 4063232*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^19*b^3*c^19 - 1703936*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b^4*c^19 - 327680*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^17*b^5*c^19 + 2621440*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^20*b*c^20 + 1310720*sq
rt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^19*b^2*c^20 + 851
968*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^18*b^3*c^20
- 655360*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^19*b*
c^21 - 2*(b^2 - 4*a*c)*a^10*b^19*c^12 + 92*(b^2 - 4*a*c)*a^11*b^17*c^13 - 1
600*(b^2 - 4*a*c)*a^12*b^15*c^14 + 14336*(b^2 - 4*a*c)*a^13*b^13*c^15 - 716
80*(b^2 - 4*a*c)*a^14*b^11*c^16 + 186368*(b^2 - 4*a*c)*a^15*b^9*c^17 - 1146
88*(b^2 - 4*a*c)*a^16*b^7*c^18 - 655360*(b^2 - 4*a*c)*a^17*b^5*c^19 + 17039
36*(b^2 - 4*a*c)*a^18*b^3*c^20 - 1310720*(b^2 - 4*a*c)*a^19*b*c^21)*j)*arct
an(2*sqrt(1/2)*x/sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 256*
a^7*b^3*c^8 + 256*a^8*b*c^9 + sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b
^5*c^7 - 256*a^7*b^3*c^8 + 256*a^8*b*c^9)^2 - 4*(a^5*b^8*c^5 - 16*a^6*b^6*c
^6 + 96*a^7*b^4*c^7 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*(a^4*b^8*c^6 - 16*a^5*
b^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/(a^4*b^8*c^6 -
16*a^5*b^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/((a^7*
b^14*c^7 - 28*a^8*b^12*c^8 - 2*a^7*b^13*c^8 + 336*a^9*b^10*c^9 + 48*a^8*b^1
1*c^9 + a^7*b^12*c^9 - 2240*a^10*b^8*c^10 - 480*a^9*b^9*c^10 - 24*a^8*b^10*

c^10 + 8960*a^11*b^6*c^11 + 2560*a^10*b^7*c^11 + 240*a^9*b^8*c^11 - 21504*a
^12*b^4*c^12 - 7680*a^11*b^5*c^12 - 1280*a^10*b^6*c^12 + 28672*a^13*b^2*c^1
3 + 12288*a^12*b^3*c^13 + 3840*a^11*b^4*c^13 - 16384*a^14*c^14 - 8192*a^13*
b*c^14 - 6144*a^12*b^2*c^14 + 4096*a^13*c^15)*abs(a^4*b^8*c^5 - 16*a^5*b^6*
c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)*abs(c)) - 1/64*(3*(a^
4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9
)^2*(2*b^5*c^4 - 24*a*b^3*c^5 + 64*a^2*b*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 12*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*b^4*c^3 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt
(b^2 - 4*a*c)*c)*a*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*b^3*c^4 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a*b*c^5 - 2*(b^2 - 4*a*c)*b^3*c^4 + 16*(b^2 - 4*a*c)*a*b*c^5)*d + (a
^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^
9)^2*(2*a*b^4*c^4 + 32*a^2*b^2*c^5 - 160*a^3*c^6 - sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a*b^2*c^4 - 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^2*c^5 - 2*(b^2 - 4*a*c)*a*b^2*c^4 - 40*(b^2 - 4*a*c)*a^
2*c^5)*f - 12*(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*
c^8 + 256*a^8*c^9)^2*(2*a^2*b^3*c^4 - 8*a^3*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b*c^4)*h + (a^4
*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)
^2*(2*a^2*b^5*c^2 - 40*a^3*b^3*c^3 + 128*a^4*b*c^4 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 64*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c^2 + 32*(b^2 - 4
*a*c)*a^3*b*c^3)*j - 6*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^14*c^
7 - 29*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^12*c^8 - 2*sqrt(2)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^13*c^8 + 2*a^4*b^14*c^8 + 368*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c^9 + 50*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^5*b^11*c^9 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^1
2*c^9 - 58*a^5*b^12*c^9 - 2640*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*
b^8*c^10 - 536*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^9*c^10 - 25*sq

rt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^10*c^10 + 736*a^6*b^10*c^10 + 1
1520*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^11 + 3136*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^11 + 268*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^6*b^8*c^11 - 5280*a^7*b^8*c^11 - 30464*sqrt(2)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^9*b^4*c^12 - 10496*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^8*b^5*c^12 - 1568*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^6*c
^12 + 23040*a^8*b^6*c^12 + 45056*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
10*b^2*c^13 + 18944*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^3*c^13 +
5248*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^4*c^13 - 60928*a^9*b^4*c
^13 - 28672*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*c^14 - 14336*sqrt(
2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b*c^14 - 9472*sqrt(2)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^9*b^2*c^14 + 90112*a^10*b^2*c^14 + 7168*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^10*c^15 - 57344*a^11*c^15 - 2*(b^2 - 4*a*c)*a^4
*b^12*c^8 + 50*(b^2 - 4*a*c)*a^5*b^10*c^9 - 536*(b^2 - 4*a*c)*a^6*b^8*c^10
+ 3136*(b^2 - 4*a*c)*a^7*b^6*c^11 - 10496*(b^2 - 4*a*c)*a^8*b^4*c^12 + 1894
4*(b^2 - 4*a*c)*a^9*b^2*c^13 - 14336*(b^2 - 4*a*c)*a^10*c^14)*d*abs(a^4*b^8
*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - 2
*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^13*c^7 - 36*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^8 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a^5*b^12*c^8 + 2*a^5*b^13*c^8 + 480*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^7*b^9*c^9 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^10*c
^9 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^11*c^9 - 72*a^6*b^11*c^9
 - 3200*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^7*c^10 - 704*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^8*c^10 - 32*sqrt(2)*sqrt(b*c - sqrt(b
^2 - 4*a*c)*c)*a^6*b^9*c^10 + 960*a^7*b^9*c^10 + 11520*sqrt(2)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^9*b^5*c^11 + 3584*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c
)*c)*a^8*b^6*c^11 + 352*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^7*c^1
1 - 6400*a^8*b^7*c^11 - 21504*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*
b^3*c^12 - 8704*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^4*c^12 - 1792
*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^5*c^12 + 23040*a^9*b^5*c^12
+ 16384*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b*c^13 + 8192*sqrt(2)*
sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^13 + 4352*sqrt(2)*sqrt(b*c - sqr
t(b^2 - 4*a*c)*c)*a^9*b^3*c^13 - 43008*a^10*b^3*c^13 - 4096*sqrt(2)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^10*b*c^14 + 32768*a^11*b*c^14 - 2*(b^2 - 4*a*c)*
a^5*b^11*c^8 + 64*(b^2 - 4*a*c)*a^6*b^9*c^9 - 704*(b^2 - 4*a*c)*a^7*b^7*c^1
0 + 3584*(b^2 - 4*a*c)*a^8*b^5*c^11 - 8704*(b^2 - 4*a*c)*a^9*b^3*c^12 + 819
2*(b^2 - 4*a*c)*a^10*b*c^13)*f*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^
4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - 6*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^6*b^12*c^7 - 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^10
*c^8 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^6*b^11*c^8 + 2*a^6*b^12*
c^8 + 80*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^8*c^9 + 24*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^9*c^9 + sqrt(2)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*a^6*b^10*c^9 - 32*a^7*b^10*c^9 - 64*sqrt(2)*sqrt(b*c - sqrt(b^2 -
 4*a*c)*c)*a^8*b^7*c^10 - 12*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^
8*c^10 + 160*a^8*b^8*c^10 - 1280*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^

10*b^4*c^11 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^5*c^11 + 32
*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^6*c^11 + 4096*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^2*c^12 + 1536*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^10*b^3*c^12 + 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9
*b^4*c^12 - 2560*a^10*b^4*c^12 - 4096*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*a^12*c^13 - 2048*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b*c^13 - 7
68*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^2*c^13 + 8192*a^11*b^2*c^
13 + 1024*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*c^14 - 8192*a^12*c^1
4 - 2*(b^2 - 4*a*c)*a^6*b^10*c^8 + 24*(b^2 - 4*a*c)*a^7*b^8*c^9 - 64*(b^2 -
 4*a*c)*a^8*b^6*c^10 - 256*(b^2 - 4*a*c)*a^9*b^4*c^11 + 1536*(b^2 - 4*a*c)*
a^10*b^2*c^12 - 2048*(b^2 - 4*a*c)*a^11*c^13)*h*abs(a^4*b^8*c^5 - 16*a^5*b^
6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - 2*(sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^12*c^6 - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^7*b^11*c^7 + 2*a^7*b^12*c^7 - 240*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^9*b^8*c^8 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^9*c^
8 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^7*b^10*c^8 + 2560*sqrt(2)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^6*c^9 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^9*b^7*c^9 + 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^
8*c^9 - 480*a^9*b^8*c^9 - 11520*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^1
1*b^4*c^10 - 3328*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^5*c^10 - 2
24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^6*c^10 + 5120*a^10*b^6*c^1
0 + 24576*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^2*c^11 + 9728*sqrt
(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^3*c^11 + 1664*sqrt(2)*sqrt(b*c -
 sqrt(b^2 - 4*a*c)*c)*a^10*b^4*c^11 - 23040*a^11*b^4*c^11 - 20480*sqrt(2)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*c^12 - 10240*sqrt(2)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^12*b*c^12 - 4864*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^
11*b^2*c^12 + 49152*a^12*b^2*c^12 + 5120*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^12*c^13 - 40960*a^13*c^13 - 2*(b^2 - 4*a*c)*a^7*b^10*c^7 - 8*(b^2 -
 4*a*c)*a^8*b^8*c^8 + 448*(b^2 - 4*a*c)*a^9*b^6*c^9 - 3328*(b^2 - 4*a*c)*a^
10*b^4*c^10 + 9728*(b^2 - 4*a*c)*a^11*b^2*c^11 - 10240*(b^2 - 4*a*c)*a^12*c
^12)*j*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8
 + 256*a^8*c^9) + 3*(2*a^8*b^21*c^14 - 84*a^9*b^19*c^15 + 1648*a^10*b^17*c^1
6 - 19712*a^11*b^15*c^17 + 157696*a^12*b^13*c^18 - 874496*a^13*b^11*c^19 +
3383296*a^14*b^9*c^20 - 8978432*a^15*b^7*c^21 + 15597568*a^16*b^5*c^22 - 15
990784*a^17*b^3*c^23 + 7340032*a^18*b*c^24 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^21*c^12 + 42*sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^19*c^13 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^20*c^13 - 824*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 - 76*sqrt(2)*sqrt(b^2 - 4*
a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^18*c^14 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^8*b^19*c^14 + 9856*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^15*c^15 + 1344*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^16*c^15 + 38*sqrt(2)*sq
rt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^17*c^15 - 78848*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^13*c^16 - 14336

*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^14*c^16 - 672*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^15*c^16 + 437248*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^11*c^17 + 100352*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^12*c^17 + 7168*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^13*c^17 - 1691648*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^9*c^18 - 473088*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^10*c^18 - 50176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^11*c^18 + 4489216*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^7*c^19 + 1490944*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^8*c^19 + 236544*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^9*c^19 - 7798784*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^5*c^20 - 3014656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^6*c^20 - 745472*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^7*c^20 + 7995392*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^3*c^21 + 3538944*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^4*c^21 + 1507328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^5*c^21 - 3670016*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b*c^22 - 1835008*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^2*c^22 - 1769472*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^3*c^22 + 917504*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b*c^23 - 2*(b^2 - 4*a*c)*a^8*b^19*c^14 + 76*(b^2 - 4*a*c)*a^9*b^17*c^15 - 1344*(b^2 - 4*a*c)*a^10*b^15*c^16 + 14336*(b^2 - 4*a*c)*a^11*b^13*c^17 - 100352*(b^2 - 4*a*c)*a^12*b^11*c^18 + 473088*(b^2 - 4*a*c)*a^13*b^9*c^19 - 1490944*(b^2 - 4*a*c)*a^14*b^7*c^20 + 3014656*(b^2 - 4*a*c)*a^15*b^5*c^21 - 3538944*(b^2 - 4*a*c)*a^16*b^3*c^22 + 1835008*(b^2 - 4*a*c)*a^17*b*c^23)*d + (2*a^9*b^20*c^14 - 168*a^10*b^18*c^15 + 4224*a^11*b^16*c^16 - 53760*a^12*b^14*c^17 + 408576*a^13*b^12*c^18 - 1978368*a^14*b^10*c^19 + 6193152*a^15*b^8*c^20 - 12189696*a^16*b^6*c^21 + 13762560*a^17*b^4*c^22 - 6815744*a^18*b^2*c^23 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^20*c^12 + 84*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^18*c^13 + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^19*c^13 - 2112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^16*c^14 - 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^9*b^18*c^14 + 26880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^14*c^15 + 3584*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^15*c^15 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^16*c^15 - 204288*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^12*c^16 - 39424*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^13*c^16 - 1792*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^14*c^16 + 989184*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^10*c^17 + 250880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(

b^2 - 4*a*c)*c)*a^13*b^11*c^17 + 19712*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^12*c^17 - 3096576*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^8*c^18 - 974848*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^9*c^18 - 125440*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^10*c^18 + 6094848*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^6*c^19 + 2293760*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^7*c^19 + 487424*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^8*c^19 - 6881280*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^4*c^20 - 3014656*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^5*c^20 - 1146880*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^6*c^20 + 3407872*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b^2*c^21 + 1703936*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^3*c^21 + 1507328*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^4*c^21 - 851968*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^2*c^22 - 2*(b^2 - 4*a*c)*a^9*b^18*c^14 + 160*(b^2 - 4*a*c)*a^10*b^16*c^15 - 3584*(b^2 - 4*a*c)*a^11*b^14*c^16 + 39424*(b^2 - 4*a*c)*a^12*b^12*c^17 - 250880*(b^2 - 4*a*c)*a^13*b^10*c^18 + 974848*(b^2 - 4*a*c)*a^14*b^8*c^19 - 2293760*(b^2 - 4*a*c)*a^15*b^6*c^20 + 3014656*(b^2 - 4*a*c)*a^16*b^4*c^21 - 1703936*(b^2 - 4*a*c)*a^17*b^2*c^22)*f + 6*(6*a^10*b^19*c^14 - 184*a^11*b^17*c^15 + 2432*a^12*b^15*c^16 - 17920*a^13*b^13*c^17 + 78848*a^14*b^11*c^18 - 200704*a^15*b^9*c^19 + 229376*a^16*b^7*c^20 + 131072*a^17*b^5*c^21 - 655360*a^18*b^3*c^22 + 524288*a^19*b*c^23 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^19*c^12 + 92*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^17*c^13 + 6*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^18*c^13 - 1216*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^15*c^14 - 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^16*c^14 - 3*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^17*c^14 + 8960*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^13*c^15 + 1792*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^14*c^15 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^15*c^15 - 39424*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^11*c^16 - 10752*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^12*c^16 - 896*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^13*c^16 + 100352*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^9*c^17 + 35840*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^10*c^17 + 5376*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^11*c^17 - 114688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^7*c^18 - 57344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^8*c^18 - 17920*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^9*c^18 - 65536*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^5*c^19 + 28672*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^7*c^19 + 327680*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2

```
 - 4*a*c)*c)*a^18*b^3*c^20 + 131072*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sq
rt(b^2 - 4*a*c)*c)*a^17*b^4*c^20 - 262144*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c - sqrt(b^2 - 4*a*c)*c)*a^19*b*c^21 - 131072*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b^2*c^21 - 65536*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^3*c^21 + 65536*sqrt(2)*sqrt(b^2 -
4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b*c^22 - 6*(b^2 - 4*a*c)*a^10*b
^17*c^14 + 160*(b^2 - 4*a*c)*a^11*b^15*c^15 - 1792*(b^2 - 4*a*c)*a^12*b^13*
c^16 + 10752*(b^2 - 4*a*c)*a^13*b^11*c^17 - 35840*(b^2 - 4*a*c)*a^14*b^9*c^
18 + 57344*(b^2 - 4*a*c)*a^15*b^7*c^19 - 131072*(b^2 - 4*a*c)*a^17*b^3*c^21
 + 131072*(b^2 - 4*a*c)*a^18*b*c^22)*h - (2*a^10*b^21*c^12 - 100*a^11*b^19*
c^13 + 1968*a^12*b^17*c^14 - 20736*a^13*b^15*c^15 + 129024*a^14*b^13*c^16 -
 473088*a^15*b^11*c^17 + 860160*a^16*b^9*c^18 + 196608*a^17*b^7*c^19 - 4325
376*a^18*b^5*c^20 + 8126464*a^19*b^3*c^21 - 5242880*a^20*b*c^22 - sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^21*c^10 + 50*sqrt(2
)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^19*c^11 + 2*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^20*c^11 - 984*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^12*b^17*c^12 -
92*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^11*b^18*c^12
 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^10*b^19*c^12
 + 10368*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^1
5*c^13 + 1600*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^1
2*b^16*c^13 + 46*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*
a^11*b^17*c^13 - 64512*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*
c)*c)*a^14*b^13*c^14 - 14336*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^13*b^14*c^14 - 800*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(
b^2 - 4*a*c)*c)*a^12*b^15*c^14 + 236544*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*a^15*b^11*c^15 + 71680*sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^12*c^15 + 7168*sqrt(2)*sqrt(b^2 - 4*a*c
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^13*b^13*c^15 - 430080*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^9*c^16 - 186368*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*b^10*c^16 - 35840*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^14*b^11*c^16 - 98304
*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^7*c^17 +
114688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^16*b^8*c
^17 + 93184*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^15*
b^9*c^17 + 2162688*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a^18*b^5*c^18 + 655360*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*
a*c)*c)*a^17*b^6*c^18 - 57344*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2
 - 4*a*c)*c)*a^16*b^7*c^18 - 4063232*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - s
qrt(b^2 - 4*a*c)*c)*a^19*b^3*c^19 - 1703936*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b^4*c^19 - 327680*sqrt(2)*sqrt(b^2 - 4*a*c)
*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^17*b^5*c^19 + 2621440*sqrt(2)*sqrt(b^2 -
 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^20*b*c^20 + 1310720*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^19*b^2*c^20 + 851968*sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^18*b^3*c^20 - 655360*s
```

qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^19*b*c^21 - 2*(b
^2 - 4*a*c)*a^10*b^19*c^12 + 92*(b^2 - 4*a*c)*a^11*b^17*c^13 - 1600*(b^2 -
4*a*c)*a^12*b^15*c^14 + 14336*(b^2 - 4*a*c)*a^13*b^13*c^15 - 71680*(b^2 - 4
*a*c)*a^14*b^11*c^16 + 186368*(b^2 - 4*a*c)*a^15*b^9*c^17 - 114688*(b^2 - 4
*a*c)*a^16*b^7*c^18 - 655360*(b^2 - 4*a*c)*a^17*b^5*c^19 + 1703936*(b^2 - 4
*a*c)*a^18*b^3*c^20 - 1310720*(b^2 - 4*a*c)*a^19*b*c^21)*j)*arctan(2*sqrt(1
/2)*x/sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 256*a^7*b^3*c^8
 + 256*a^8*b*c^9 - sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 25
6*a^7*b^3*c^8 + 256*a^8*b*c^9)^2 - 4*(a^5*b^8*c^5 - 16*a^6*b^6*c^6 + 96*a^7
*b^4*c^7 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*(a^4*b^8*c^6 - 16*a^5*b^6*c^7 + 9
6*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/(a^4*b^8*c^6 - 16*a^5*b^6
*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/((a^7*b^14*c^7 -
28*a^8*b^12*c^8 - 2*a^7*b^13*c^8 + 336*a^9*b^10*c^9 + 48*a^8*b^11*c^9 + a^7
*b^12*c^9 - 2240*a^10*b^8*c^10 - 480*a^9*b^9*c^10 - 24*a^8*b^10*c^10 + 8960
*a^11*b^6*c^11 + 2560*a^10*b^7*c^11 + 240*a^9*b^8*c^11 - 21504*a^12*b^4*c^1
2 - 7680*a^11*b^5*c^12 - 1280*a^10*b^6*c^12 + 28672*a^13*b^2*c^13 + 12288*a
^12*b^3*c^13 + 3840*a^11*b^4*c^13 - 16384*a^14*c^14 - 8192*a^13*b*c^14 - 61
44*a^12*b^2*c^14 + 4096*a^13*c^15)*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^
6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)*abs(c)) + 1/4*k*log(abs(c*x^4 +
b*x^2 + a))/c^3 + 1/16*(12*(b^3*c^5 - 4*a*b*c^6 - 2*b^2*c^6 + b*c^7 - (b^2*
c^5 - 4*a*c^6 - 2*b*c^6 + c^7)*sqrt(b^2 - 4*a*c))*e*abs(a^4*b^8*c^5 - 16*a^
5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - 6*(b^4*c^4 -
4*a*b^2*c^5 - 2*b^3*c^5 + b^2*c^6 + (b^3*c^4 - 4*a*b*c^5 - 2*b^2*c^5 + b*c^
6)*sqrt(b^2 - 4*a*c))*g*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 -
 256*a^7*b^2*c^8 + 256*a^8*c^9) + 2*(b^5*c^3 - 2*a*b^3*c^4 - 2*b^4*c^4 - 8*
a^2*b*c^5 - 4*a*b^2*c^5 + b^3*c^5 + 2*a*b*c^6 + (b^4*c^3 - 2*a*b^2*c^4 - 2*
b^3*c^4 - 8*a^2*c^5 - 4*a*b*c^5 + b^2*c^5 + 2*a*c^6)*sqrt(b^2 - 4*a*c))*i*a
bs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^
8*c^9) - (b^8 - 14*a*b^6*c - 2*b^7*c + 70*a^2*b^4*c^2 + 20*a*b^5*c^2 + b^6*
c^2 - 120*a^3*b^2*c^3 - 60*a^2*b^3*c^3 - 10*a*b^4*c^3 + 30*a^2*b^2*c^4 - (b
^7 - 14*a*b^5*c - 2*b^6*c + 70*a^2*b^3*c^2 + 20*a*b^4*c^2 + b^5*c^2 - 120*a
^3*b*c^3 - 60*a^2*b^2*c^3 - 10*a*b^3*c^3 + 30*a^2*b*c^4)*sqrt(b^2 - 4*a*c))
*k*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 25
6*a^8*c^9) + 12*(a^4*b^11*c^10 - 20*a^5*b^9*c^11 - 2*a^4*b^10*c^11 + 160*a^
6*b^7*c^12 + 32*a^5*b^8*c^12 + a^4*b^9*c^12 - 640*a^7*b^5*c^13 - 192*a^6*b^
6*c^13 - 16*a^5*b^7*c^13 + 1280*a^8*b^3*c^14 + 512*a^7*b^4*c^14 + 96*a^6*b^
5*c^14 - 1024*a^9*b*c^15 - 512*a^8*b^2*c^15 - 256*a^7*b^3*c^15 + 256*a^8*b*
c^16 + (a^4*b^10*c^10 - 16*a^5*b^8*c^11 - 2*a^4*b^9*c^11 + 96*a^6*b^6*c^12
+ 24*a^5*b^7*c^12 + a^4*b^8*c^12 - 256*a^7*b^4*c^13 - 96*a^6*b^5*c^13 - 12*
a^5*b^6*c^13 + 256*a^8*b^2*c^14 + 128*a^7*b^3*c^14 + 48*a^6*b^4*c^14 - 64*a
^7*b^2*c^15)*sqrt(b^2 - 4*a*c))*e - 6*(a^4*b^12*c^9 - 20*a^5*b^10*c^10 - 2*
a^4*b^11*c^10 + 160*a^6*b^8*c^11 + 32*a^5*b^9*c^11 + a^4*b^10*c^11 - 640*a^
7*b^6*c^12 - 192*a^6*b^7*c^12 - 16*a^5*b^8*c^12 + 1280*a^8*b^4*c^13 + 512*a
^7*b^5*c^13 + 96*a^6*b^6*c^13 - 1024*a^9*b^2*c^14 - 512*a^8*b^3*c^14 - 256*
a^7*b^4*c^14 + 256*a^8*b^2*c^15 + (a^4*b^11*c^9 - 16*a^5*b^9*c^10 - 2*a^4*b

^10*c^10 + 96*a^6*b^7*c^11 + 24*a^5*b^8*c^11 + a^4*b^9*c^11 - 256*a^7*b^5*c
^12 - 96*a^6*b^6*c^12 - 12*a^5*b^7*c^12 + 256*a^8*b^3*c^13 + 128*a^7*b^4*c^
13 + 48*a^6*b^5*c^13 - 64*a^7*b^3*c^14)*sqrt(b^2 - 4*a*c))*g + 2*(a^4*b^13*
c^8 - 18*a^5*b^11*c^9 - 2*a^4*b^12*c^9 + 120*a^6*b^9*c^10 + 28*a^5*b^10*c^1
0 + a^4*b^11*c^10 - 320*a^7*b^7*c^11 - 128*a^6*b^8*c^11 - 14*a^5*b^9*c^11 +
 128*a^7*b^6*c^12 + 64*a^6*b^7*c^12 + 1536*a^9*b^3*c^13 + 512*a^8*b^4*c^13
 - 64*a^7*b^5*c^13 - 2048*a^10*b*c^14 - 1024*a^9*b^2*c^14 - 256*a^8*b^3*c^14
 + 512*a^9*b*c^15 + (a^4*b^12*c^8 - 14*a^5*b^10*c^9 - 2*a^4*b^11*c^9 + 64*a
^6*b^8*c^10 + 20*a^5*b^9*c^10 + a^4*b^10*c^10 - 64*a^7*b^6*c^11 - 48*a^6*b^
7*c^11 - 10*a^5*b^8*c^11 - 256*a^8*b^4*c^12 - 64*a^7*b^5*c^12 + 24*a^6*b^6*
c^12 + 512*a^9*b^2*c^13 + 256*a^8*b^3*c^13 + 32*a^7*b^4*c^13 - 128*a^8*b^2*
c^14)*sqrt(b^2 - 4*a*c))*i - (a^4*b^16*c^5 - 30*a^5*b^14*c^6 - 2*a^4*b^15*c
^6 + 390*a^6*b^12*c^7 + 52*a^5*b^13*c^7 + a^4*b^14*c^7 - 2840*a^7*b^10*c^8
 - 572*a^6*b^11*c^8 - 26*a^5*b^12*c^8 + 12480*a^8*b^8*c^9 + 3392*a^7*b^9*c^9
 + 286*a^6*b^10*c^9 - 33024*a^9*b^6*c^10 - 11392*a^8*b^7*c^10 - 1696*a^7*b^
8*c^10 + 48640*a^10*b^4*c^11 + 20480*a^9*b^5*c^11 + 5696*a^8*b^6*c^11 - 307
20*a^11*b^2*c^12 - 15360*a^10*b^3*c^12 - 10240*a^9*b^4*c^12 + 7680*a^10*b^2
*c^13 + (a^4*b^15*c^5 - 26*a^5*b^13*c^6 - 2*a^4*b^14*c^6 + 286*a^6*b^11*c^7
 + 44*a^5*b^12*c^7 + a^4*b^13*c^7 - 1696*a^7*b^9*c^8 - 396*a^6*b^10*c^8 - 2
2*a^5*b^11*c^8 + 5696*a^8*b^7*c^9 + 1808*a^7*b^8*c^9 + 198*a^6*b^9*c^9 - 10
240*a^9*b^5*c^10 - 4160*a^8*b^6*c^10 - 904*a^7*b^7*c^10 + 7680*a^10*b^3*c^1
1 + 3840*a^9*b^4*c^11 + 2080*a^8*b^5*c^11 - 1920*a^9*b^3*c^12)*sqrt(b^2 - 4
*a*c))*k)*log(x^2 + 1/2*(a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 25
6*a^7*b^3*c^8 + 256*a^8*b*c^9 + sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6
*b^5*c^7 - 256*a^7*b^3*c^8 + 256*a^8*b*c^9)^2 - 4*(a^5*b^8*c^5 - 16*a^6*b^6
*c^6 + 96*a^7*b^4*c^7 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*(a^4*b^8*c^6 - 16*a^
5*b^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/(a^4*b^8*c^6
 - 16*a^5*b^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10))/((a*b
^6*c^2 - 12*a^2*b^4*c^3 - 2*a*b^5*c^3 + 48*a^3*b^2*c^4 + 16*a^2*b^3*c^4 + a
*b^4*c^4 - 64*a^4*c^5 - 32*a^3*b*c^5 - 8*a^2*b^2*c^5 + 16*a^3*c^6)*c^2*abs(
a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c
^9)) + 1/16*(12*(b^3*c^5 - 4*a*b*c^6 - 2*b^2*c^6 + b*c^7 - (b^2*c^5 - 4*a*c
^6 - 2*b*c^6 + c^7)*sqrt(b^2 - 4*a*c))*e*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 +
 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - 6*(b^4*c^4 - 4*a*b^2*c^5
 - 2*b^3*c^5 + b^2*c^6 - (b^3*c^4 - 4*a*b*c^5 - 2*b^2*c^5 + b*c^6)*sqrt(b^2
 - 4*a*c))*g*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^
2*c^8 + 256*a^8*c^9) + 2*(b^5*c^3 - 2*a*b^3*c^4 - 2*b^4*c^4 - 8*a^2*b*c^5 -
 4*a*b^2*c^5 + b^3*c^5 + 2*a*b*c^6 - (b^4*c^3 - 2*a*b^2*c^4 - 2*b^3*c^4 - 8
*a^2*c^5 - 4*a*b*c^5 + b^2*c^5 + 2*a*c^6)*sqrt(b^2 - 4*a*c))*i*abs(a^4*b^8*
c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9) - (b
^8 - 14*a*b^6*c - 2*b^7*c + 70*a^2*b^4*c^2 + 20*a*b^5*c^2 + b^6*c^2 - 120*a
^3*b^2*c^3 - 60*a^2*b^3*c^3 - 10*a*b^4*c^3 + 30*a^2*b^2*c^4 - (b^7 - 14*a*b
^5*c - 2*b^6*c + 70*a^2*b^3*c^2 + 20*a*b^4*c^2 + b^5*c^2 - 120*a^3*b*c^3 -
60*a^2*b^2*c^3 - 10*a*b^3*c^3 + 30*a^2*b*c^4)*sqrt(b^2 - 4*a*c))*k*abs(a^4*
b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)

- 12*(a^4*b^11*c^10 - 20*a^5*b^9*c^11 - 2*a^4*b^10*c^11 + 160*a^6*b^7*c^12 + 32*a^5*b^8*c^12 + a^4*b^9*c^12 - 640*a^7*b^5*c^13 - 192*a^6*b^6*c^13 - 16 *a^5*b^7*c^13 + 1280*a^8*b^3*c^14 + 512*a^7*b^4*c^14 + 96*a^6*b^5*c^14 - 10 24*a^9*b*c^15 - 512*a^8*b^2*c^15 - 256*a^7*b^3*c^15 + 256*a^8*b*c^16 - (a^4 *b^10*c^10 - 16*a^5*b^8*c^11 - 2*a^4*b^9*c^11 + 96*a^6*b^6*c^12 + 24*a^5*b^ 7*c^12 + a^4*b^8*c^12 - 256*a^7*b^4*c^13 - 96*a^6*b^5*c^13 - 12*a^5*b^6*c^1 3 + 256*a^8*b^2*c^14 + 128*a^7*b^3*c^14 + 48*a^6*b^4*c^14 - 64*a^7*b^2*c^15)*sqrt(b^2 - 4*a*c))*e + 6*(a^4*b^12*c^9 - 20*a^5*b^10*c^10 - 2*a^4*b^11*c^ 10 + 160*a^6*b^8*c^11 + 32*a^5*b^9*c^11 + a^4*b^10*c^11 - 640*a^7*b^6*c^12 - 192*a^6*b^7*c^12 - 16*a^5*b^8*c^12 + 1280*a^8*b^4*c^13 + 512*a^7*b^5*c^13 + 96*a^6*b^6*c^13 - 1024*a^9*b^2*c^14 - 512*a^8*b^3*c^14 - 256*a^7*b^4*c^1 4 + 256*a^8*b^2*c^15 + (a^4*b^11*c^9 - 16*a^5*b^9*c^10 - 2*a^4*b^10*c^10 + 96*a^6*b^7*c^11 + 24*a^5*b^8*c^11 + a^4*b^9*c^11 - 256*a^7*b^5*c^12 - 96*a^ 6*b^6*c^12 - 12*a^5*b^7*c^12 + 256*a^8*b^3*c^13 + 128*a^7*b^4*c^13 + 48*a^6 *b^5*c^13 - 64*a^7*b^3*c^14)*sqrt(b^2 - 4*a*c))*g - 2*(a^4*b^13*c^8 - 18*a^ 5*b^11*c^9 - 2*a^4*b^12*c^9 + 120*a^6*b^9*c^10 + 28*a^5*b^10*c^10 + a^4*b^1 1*c^10 - 320*a^7*b^7*c^11 - 128*a^6*b^8*c^11 - 14*a^5*b^9*c^11 + 128*a^7*b^ 6*c^12 + 64*a^6*b^7*c^12 + 1536*a^9*b^3*c^13 + 512*a^8*b^4*c^13 - 64*a^7*b^ 5*c^13 - 2048*a^10*b*c^14 - 1024*a^9*b^2*c^14 - 256*a^8*b^3*c^14 + 512*a^9* b*c^15 + (a^4*b^12*c^8 - 14*a^5*b^10*c^9 - 2*a^4*b^11*c^9 + 64*a^6*b^8*c^10 + 20*a^5*b^9*c^10 + a^4*b^10*c^10 - 64*a^7*b^6*c^11 - 48*a^6*b^7*c^11 - 10 *a^5*b^8*c^11 - 256*a^8*b^4*c^12 - 64*a^7*b^5*c^12 + 24*a^6*b^6*c^12 + 512* a^9*b^2*c^13 + 256*a^8*b^3*c^13 + 32*a^7*b^4*c^13 - 128*a^8*b^2*c^14)*sqrt(b^2 - 4*a*c))*i + (a^4*b^16*c^5 - 30*a^5*b^14*c^6 - 2*a^4*b^15*c^6 + 390*a^ 6*b^12*c^7 + 52*a^5*b^13*c^7 + a^4*b^14*c^7 - 2840*a^7*b^10*c^8 - 572*a^6*b ^11*c^8 - 26*a^5*b^12*c^8 + 12480*a^8*b^8*c^9 + 3392*a^7*b^9*c^9 + 286*a^6* b^10*c^9 - 33024*a^9*b^6*c^10 - 11392*a^8*b^7*c^10 - 1696*a^7*b^8*c^10 + 48 640*a^10*b^4*c^11 + 20480*a^9*b^5*c^11 + 5696*a^8*b^6*c^11 - 30720*a^11*b^2 *c^12 - 15360*a^10*b^3*c^12 - 10240*a^9*b^4*c^12 + 7680*a^10*b^2*c^13 - (a^ 4*b^15*c^5 - 26*a^5*b^13*c^6 - 2*a^4*b^14*c^6 + 286*a^6*b^11*c^7 + 44*a^5*b ^12*c^7 + a^4*b^13*c^7 - 1696*a^7*b^9*c^8 - 396*a^6*b^10*c^8 - 22*a^5*b^11* c^8 + 5696*a^8*b^7*c^9 + 1808*a^7*b^8*c^9 + 198*a^6*b^9*c^9 - 10240*a^9*b^5 *c^10 - 4160*a^8*b^6*c^10 - 904*a^7*b^7*c^10 + 7680*a^10*b^3*c^11 + 3840*a^ 9*b^4*c^11 + 2080*a^8*b^5*c^11 - 1920*a^9*b^3*c^12)*sqrt(b^2 - 4*a*c))*k)*l og(x^2 + 1/2*(a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 256*a^7*b^3*c ^8 + 256*a^8*b*c^9 - sqrt((a^4*b^9*c^5 - 16*a^5*b^7*c^6 + 96*a^6*b^5*c^7 - 256*a^7*b^3*c^8 + 256*a^8*b*c^9)^2 - 4*(a^5*b^8*c^5 - 16*a^6*b^6*c^6 + 96*a ^7*b^4*c^7 - 256*a^8*b^2*c^8 + 256*a^9*c^9)*(a^4*b^8*c^6 - 16*a^5*b^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10)))/(a^4*b^8*c^6 - 16*a^5*b ^6*c^7 + 96*a^6*b^4*c^8 - 256*a^7*b^2*c^9 + 256*a^8*c^10))/((a*b^6*c^2 - 12 *a^2*b^4*c^3 - 2*a*b^5*c^3 + 48*a^3*b^2*c^4 + 16*a^2*b^3*c^4 + a*b^4*c^4 - 64*a^4*c^5 - 32*a^3*b*c^5 - 8*a^2*b^2*c^5 + 16*a^3*c^6)*c^2*abs(a^4*b^8*c^5 - 16*a^5*b^6*c^6 + 96*a^6*b^4*c^7 - 256*a^7*b^2*c^8 + 256*a^8*c^9)) - 1/8* (2*a^2*b^3*c^3*e - 20*a^3*b*c^4*e + 2*a^3*b^2*c^3*g + 16*a^4*c^4*g - 12*a^4 *b*c^3*i - 6*a^4*b^4*k + 42*a^5*b^2*c*k - 48*a^6*c^2*k - (3*b^3*c^5*d - 24*

```
a*b*c^6*d + a*b^2*c^5*f + 20*a^2*c^6*f - 12*a^2*b*c^5*h + a^2*b^3*c^3*j - 1
6*a^3*b*c^4*j)*x^7 - 4*(6*a^2*c^6*e - 3*a^2*b*c^5*g + a^2*b^2*c^4*i + 2*a^3
*c^5*i + 2*a^2*b^5*c*k - 15*a^3*b^3*c^2*k + 25*a^4*b*c^3*k)*x^6 - (6*b^4*c^
4*d - 49*a*b^2*c^5*d + 28*a^2*c^6*d + 2*a*b^3*c^4*f + 28*a^2*b*c^5*f - 19*a
^2*b^2*c^4*h + 4*a^3*c^5*h - a^2*b^4*c^2*j - 5*a^3*b^2*c^3*j - 36*a^4*c^4*j
)*x^5 - 2*(18*a^2*b*c^5*e - 9*a^2*b^2*c^4*g + 3*a^2*b^3*c^3*i + 6*a^3*b*c^4
*i + 3*a^2*b^6*k - 19*a^3*b^4*c*k + 11*a^4*b^2*c^2*k + 32*a^5*c^3*k)*x^4 -
(3*b^5*c^3*d - 20*a*b^3*c^4*d - 4*a^2*b*c^5*d + a*b^4*c^3*f + 5*a^2*b^2*c^4
*f + 36*a^3*c^5*f - 5*a^2*b^3*c^3*h - 16*a^3*b*c^4*h - 2*a^3*b^3*c^2*j - 28
*a^4*b*c^3*j)*x^3 - 4*(2*a^2*b^2*c^4*e + 10*a^3*c^5*e - a^2*b^3*c^3*g - 5*a
^3*b*c^4*g + 5*a^3*b^2*c^3*i - 2*a^4*c^4*i + 3*a^3*b^5*k - 22*a^4*b^3*c*k +
 31*a^5*b*c^2*k)*x^2 - (5*a*b^4*c^3*d - 37*a^2*b^2*c^4*d + 44*a^3*c^5*d - a
^2*b^3*c^3*f + 16*a^3*b*c^4*f - 3*a^3*b^2*c^3*h - 12*a^4*c^4*h - a^4*b^2*c^
2*j - 20*a^5*c^3*j)*x)/((c*x^4 + b*x^2 + a)^2*(b^2 - 4*a*c)^2*a^2*c^3)
```

Mupad [B] (verification not implemented)

Time = 27.14 (sec) , antiderivative size = 97905, normalized size of antiderivative = 83.18

$$\int \frac{d+ex+fx^2+gx^3+hx^4+ix^5+jx^8+kx^{11}}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5 + j*x^8 + k*x^11)/(a + b*x^2 +
 c*x^4)^3,x)
```

```
[Out] ((x^7*(3*b^3*c^2*d + 20*a^2*c^3*f + a^2*b^3*j - 24*a*b*c^3*d - 16*a^3*b*c*j
 + a*b^2*c^2*f - 12*a^2*b*c^2*h))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) -
(b^3*c^3*e + 8*a^2*c^4*g - 3*a^2*b^4*k - 24*a^4*c^2*k - 10*a*b*c^4*e + a*b^
2*c^3*g - 6*a^2*b*c^3*i + 21*a^3*b^2*c*k)/(4*c^3*(b^4 + 16*a^2*c^2 - 8*a*b^
2*c)) + (x^4*(3*b^6*k - 9*b^2*c^4*g + 3*b^3*c^3*i + 32*a^3*c^3*k + 18*b*c^5
*e + 11*a^2*b^2*c^2*k + 6*a*b*c^4*i - 19*a*b^4*c*k))/(4*c^3*(b^4 + 16*a^2*c
^2 - 8*a*b^2*c)) + (x^2*(2*b^2*c^4*e - b^3*c^3*g - 2*a^2*c^4*i + 10*a*c^5*e
 + 3*a*b^5*k - 5*a*b*c^4*g + 5*a*b^2*c^3*i - 22*a^2*b^3*c*k + 31*a^3*b*c^2*
k))/(2*c^3*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^6*(6*c^5*e + 2*b^5*k + b^2*
c^3*i - 3*b*c^4*g + 2*a*c^4*i - 15*a*b^3*c*k + 25*a^2*b*c^2*k))/(2*c^2*(b^4
 + 16*a^2*c^2 - 8*a*b^2*c)) - (x^3*(2*a^3*b^3*j - 36*a^3*c^3*f - 3*b^5*c*d
- 5*a^2*b^2*c^2*f - a*b^4*c*f + 28*a^4*b*c*j + 20*a*b^3*c^2*d + 4*a^2*b*c^3
*d + 5*a^2*b^3*c*h + 16*a^3*b*c^2*h))/(8*a^2*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*
c)) + (x^5*(28*a^2*c^4*d + 6*b^4*c^2*d + 4*a^3*c^3*h - a^2*b^4*j - 36*a^4*c
^2*j - 19*a^2*b^2*c^2*h - 49*a*b^2*c^3*d + 2*a*b^3*c^2*f + 28*a^2*b*c^3*f -
 5*a^3*b^2*c*j))/(8*a^2*c*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) - (x*(12*a^3*c^2*
h - 44*a^2*c^3*d + a^3*b^2*j - 5*b^4*c*d + 20*a^4*c*j + a*b^3*c*f + 37*a*b^
2*c^2*d - 16*a^2*b*c^2*f + 3*a^2*b^2*c*h))/(8*a*c*(b^4 + 16*a^2*c^2 - 8*a*b
^2*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2*a*b*x^2 + 2*b*c*x^6) + symsu
m(log((10368*a*b^5*c^10*d^3 - 8000*a^5*c^11*f^3 - 567*b^7*c^9*d^3 + 169344*
```

a^3*b*c^12*d^3 + 193536*a^4*c^12*d*e^2 - 141120*a^4*c^12*d^2*f + 1728*a^6*b
*c^9*h^3 + 315*b^8*c^8*d^2*f + 6400*a^9*b*c^6*j^3 + 27648*a^5*c^11*e^2*h +
21504*a^6*c^10*d*i^2 - 135*b^9*c^7*d^2*h + 192*a^2*b^14*d*k^2 - 2880*a^6*c^
10*f*h^2 + 46080*a^6*c^10*e^2*j - 1376256*a^9*c^7*d*k^2 + 9*b^11*c^5*d^2*j
+ 64*a^3*b^13*f*k^2 - 8000*a^8*c^8*f*j^2 + 3072*a^7*c^9*h*i^2 + 192*a^4*b^1
2*h*k^2 + 5120*a^8*c^8*i^2*j - 196608*a^10*c^6*h*k^2 + 2240*a^6*b^10*j*k^2
- 327680*a^11*c^5*j*k^2 - 67824*a^2*b^3*c^11*d^3 + 35*a^2*b^6*c^8*f^3 + 84*
a^3*b^4*c^9*f^3 - 12720*a^4*b^2*c^10*f^3 + 540*a^4*b^5*c^7*h^3 + 4320*a^5*b
^3*c^8*h^3 + 35*a^6*b^7*c^3*j^3 - 1176*a^7*b^5*c^4*j^3 + 9456*a^8*b^3*c^5*j
^3 + 129024*a^5*c^11*d*e*i - 40320*a^5*c^11*d*f*h - 67200*a^6*c^10*d*f*j +
18432*a^6*c^10*e*h*i + 245760*a^7*c^9*e*f*k + 30720*a^7*c^9*e*i*j - 9600*a^
7*c^9*f*h*j + 81920*a^8*c^8*f*i*k - 6237*a*b^6*c^9*d^2*f + 210*a*b^7*c^8*d*
f^2 + 116160*a^4*b*c^11*d*f^2 - 36864*a^4*b*c^11*e^2*f + 2430*a*b^7*c^8*d^2
*h + 133056*a^4*b*c^11*d^2*h + 27648*a^5*b*c^10*d*h^2 - 324*a*b^9*c^6*d^2*j
 + 193536*a^5*b*c^10*d^2*j + 26880*a^5*b*c^10*f^2*h + 63360*a^7*b*c^8*d*j^2
 - 5568*a^3*b^12*c*d*k^2 - 4096*a^6*b*c^9*f*i^2 + 40000*a^6*b*c^9*f^2*j - 2
304*a^4*b^11*c*f*k^2 - 352256*a^9*b*c^6*f*k^2 + 8064*a^7*b*c^8*h^2*j + 1248
0*a^8*b*c^7*h*j^2 - 2112*a^5*b^10*c*h*k^2 - 41664*a^7*b^8*c*j*k^2 + 6912*a^
2*b^4*c^10*d*e^2 - 62208*a^3*b^2*c^11*d*e^2 + 42372*a^2*b^4*c^10*d^2*f - 17
64*a^2*b^5*c^9*d*f^2 - 96048*a^3*b^2*c^11*d^2*f - 4608*a^3*b^3*c^10*d*f^2 +
 1728*a^2*b^6*c^8*d*g^2 + 2304*a^3*b^3*c^10*e^2*f - 15552*a^3*b^4*c^9*d*g^2
 + 48384*a^4*b^2*c^10*d*g^2 - 13716*a^2*b^5*c^9*d^2*h + 405*a^2*b^7*c^7*d*h
^2 + 12096*a^3*b^3*c^10*d^2*h - 5400*a^3*b^5*c^8*d*h^2 + 28944*a^4*b^3*c^9*
d*h^2 + 192*a^2*b^8*c^6*d*i^2 + 576*a^3*b^5*c^8*f*g^2 - 960*a^3*b^6*c^7*d*i
^2 + 6912*a^4*b^2*c^10*e^2*h - 9216*a^4*b^3*c^9*f*g^2 - 768*a^4*b^4*c^8*d*i
^2 + 14592*a^5*b^2*c^9*d*i^2 + 3717*a^2*b^7*c^7*d^2*j - 15*a^2*b^7*c^7*f^2*
h + 3*a^2*b^11*c^3*d*j^2 - 15192*a^3*b^5*c^8*d^2*j - 360*a^3*b^5*c^8*f^2*h
+ 135*a^3*b^6*c^7*f*h^2 - 132*a^3*b^9*c^4*d*j^2 - 7920*a^4*b^3*c^9*d^2*j +
15696*a^4*b^3*c^9*f^2*h - 5580*a^4*b^4*c^8*f*h^2 + 2079*a^4*b^7*c^5*d*j^2 -
 20592*a^5*b^2*c^9*f*h^2 - 14448*a^5*b^5*c^6*d*j^2 + 37104*a^6*b^3*c^7*d*j^
2 + 64*a^3*b^7*c^6*f*i^2 + 1728*a^4*b^4*c^8*g^2*h - 768*a^4*b^5*c^7*f*i^2 +
 70656*a^4*b^10*c^2*d*k^2 + 2304*a^5*b^2*c^9*e^2*j + 6912*a^5*b^2*c^9*g^2*h
 - 3840*a^5*b^3*c^8*f*i^2 - 499008*a^5*b^8*c^3*d*k^2 + 2071104*a^6*b^6*c^4*
d*k^2 - 4853952*a^7*b^4*c^5*d*k^2 + 5399808*a^8*b^2*c^6*d*k^2 + a^2*b^9*c^5
*f^2*j + 20*a^3*b^7*c^6*f^2*j + a^3*b^10*c^3*f*j^2 - 1596*a^4*b^5*c^7*f^2*j
 - 51*a^4*b^8*c^4*f*j^2 + 16736*a^5*b^3*c^8*f^2*j + 875*a^5*b^6*c^5*f*j^2 -
 2716*a^6*b^4*c^6*f*j^2 - 39600*a^7*b^2*c^7*f*j^2 + 192*a^4*b^6*c^6*h*i^2 +
 1536*a^5*b^4*c^7*h*i^2 + 576*a^5*b^4*c^7*g^2*j + 28480*a^5*b^9*c^2*f*k^2 +
 3840*a^6*b^2*c^8*h*i^2 + 11520*a^6*b^2*c^8*g^2*j - 164096*a^6*b^7*c^3*f*k^
2 + 436800*a^7*b^5*c^4*f*k^2 - 338944*a^8*b^3*c^5*f*k^2 - 81*a^4*b^7*c^5*h^
2*j + 3*a^4*b^9*c^3*h*j^2 + 720*a^5*b^5*c^6*h^2*j - 78*a^5*b^7*c^4*h*j^2 +
17136*a^6*b^3*c^7*h^2*j - 900*a^6*b^5*c^5*h*j^2 + 22272*a^7*b^3*c^6*h*j^2 +
 64*a^5*b^6*c^5*i^2*j + 1536*a^6*b^4*c^6*i^2*j - 960*a^6*b^8*c^2*h*k^2 + 53
76*a^7*b^2*c^7*i^2*j + 108672*a^7*b^6*c^3*h*k^2 - 548160*a^8*b^4*c^4*h*k^2
+ 922368*a^9*b^2*c^5*h*k^2 + 305024*a^8*b^6*c^2*j*k^2 - 1042880*a^9*b^4*c^3

*j*k^2 + 1479936*a^10*b^2*c^4*j*k^2 - 193536*a^4*b*c^11*d*e*g - 90*a*b^8*c^
7*d*f*h + 6*a*b^10*c^5*d*f*j - 64512*a^5*b*c^10*d*g*i - 24576*a^5*b*c^10*e*
f*i - 27648*a^5*b*c^10*e*g*h - 1778688*a^6*b*c^9*d*e*k + 84096*a^6*b*c^9*d*
h*j - 46080*a^6*b*c^9*e*g*j - 9216*a^6*b*c^9*g*h*i - 592896*a^7*b*c^8*d*i*k
 - 359424*a^7*b*c^8*e*h*k - 122880*a^7*b*c^8*f*g*k - 15360*a^7*b*c^8*g*i*j
- 549888*a^8*b*c^7*e*j*k - 119808*a^8*b*c^7*h*i*k - 183296*a^9*b*c^6*i*j*k
- 6912*a^2*b^5*c^9*d*e*g + 62208*a^3*b^3*c^10*d*e*g + 2304*a^2*b^6*c^8*d*e*
i - 270*a^2*b^6*c^8*d*f*h - 16128*a^3*b^4*c^9*d*e*i + 16056*a^3*b^4*c^9*d*f
*h - 2304*a^3*b^4*c^9*e*f*g + 23040*a^4*b^2*c^10*d*e*i - 127008*a^4*b^2*c^1
0*d*f*h + 36864*a^4*b^2*c^10*e*f*g - 1152*a^2*b^7*c^7*d*g*i - 48*a^2*b^8*c^
6*d*f*j - 2304*a^2*b^9*c^5*d*e*k + 8064*a^3*b^5*c^8*d*g*i + 768*a^3*b^5*c^8
*e*f*i - 2226*a^3*b^6*c^7*d*f*j + 43776*a^3*b^7*c^6*d*e*k - 11520*a^4*b^3*c
^9*d*g*i - 10752*a^4*b^3*c^9*e*f*i - 6912*a^4*b^3*c^9*e*g*h + 33384*a^4*b^4
*c^8*d*f*j - 340992*a^4*b^5*c^7*d*e*k - 162528*a^5*b^2*c^9*d*f*j + 1241856*
a^5*b^3*c^8*d*e*k - 72*a^2*b^9*c^5*d*h*j + 1152*a^2*b^10*c^4*d*g*k - 384*a^
3*b^6*c^7*f*g*i + 2016*a^3*b^7*c^6*d*h*j - 21888*a^3*b^8*c^5*d*g*k - 768*a^
3*b^8*c^5*e*f*k + 2304*a^4*b^4*c^8*e*h*i + 5376*a^4*b^4*c^8*f*g*i - 18648*a
^4*b^5*c^7*d*h*j + 170496*a^4*b^6*c^6*d*g*k + 19968*a^4*b^6*c^6*e*f*k + 138
24*a^5*b^2*c^9*e*h*i + 12288*a^5*b^2*c^9*f*g*i + 67392*a^5*b^3*c^8*d*h*j -
2304*a^5*b^3*c^8*e*g*j - 620928*a^5*b^4*c^7*d*g*k - 119040*a^5*b^4*c^7*e*f*
k + 889344*a^6*b^2*c^8*d*g*k + 172032*a^6*b^2*c^8*e*f*k - 384*a^2*b^11*c^3*
d*i*k - 24*a^3*b^8*c^5*f*h*j + 6528*a^3*b^9*c^4*d*i*k + 384*a^3*b^9*c^4*f*g
*k - 1152*a^4*b^5*c^7*g*h*i + 1050*a^4*b^6*c^6*f*h*j - 42240*a^4*b^7*c^5*d*
i*k - 2304*a^4*b^7*c^5*e*h*k - 9984*a^4*b^7*c^5*f*g*k - 6912*a^5*b^3*c^8*g*
h*i + 768*a^5*b^4*c^7*e*i*j - 9576*a^5*b^4*c^7*f*h*j + 93312*a^5*b^5*c^6*d*
i*k + 2304*a^5*b^5*c^6*e*h*k + 59520*a^5*b^5*c^6*f*g*k + 16896*a^6*b^2*c^8*
e*i*j - 57504*a^6*b^2*c^8*f*h*j + 117504*a^6*b^3*c^7*d*i*k + 103680*a^6*b^3
*c^7*e*h*k - 86016*a^6*b^3*c^7*f*g*k - 128*a^3*b^10*c^3*f*i*k + 3072*a^4*b^
8*c^4*f*i*k + 1152*a^4*b^8*c^4*g*h*k - 384*a^5*b^5*c^6*g*i*j - 13184*a^5*b^
6*c^5*f*i*k - 1152*a^5*b^6*c^5*g*h*k - 8448*a^6*b^3*c^7*g*i*j - 11008*a^6*b
^4*c^6*f*i*k - 51840*a^6*b^4*c^6*g*h*k - 26880*a^6*b^5*c^5*e*j*k + 98304*a^
7*b^2*c^7*f*i*k + 179712*a^7*b^2*c^7*g*h*k + 231168*a^7*b^3*c^6*e*j*k - 384
*a^4*b^9*c^3*h*i*k - 384*a^5*b^7*c^4*h*i*k + 18048*a^6*b^5*c^5*h*i*k + 1344
0*a^6*b^6*c^4*g*j*k - 25344*a^7*b^3*c^6*h*i*k - 115584*a^7*b^4*c^5*g*j*k +
274944*a^8*b^2*c^6*g*j*k - 4480*a^6*b^7*c^3*i*j*k + 29568*a^7*b^5*c^4*i*j*k
 - 14592*a^8*b^3*c^5*i*j*k)/(512*(4096*a^10*c^10 + a^4*b^12*c^4 - 24*a^5*b^
10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7*b^6*c^7 + 3840*a^8*b^4*c^8 - 6144*a^9*b
^2*c^9)) + root(56371445760*a^11*b^8*c^12*z^4 - 503316480*a^8*b^14*c^9*z^4
+ 47185920*a^7*b^16*c^8*z^4 - 2621440*a^6*b^18*c^7*z^4 + 65536*a^5*b^20*c^6
*z^4 - 171798691840*a^14*b^2*c^15*z^4 + 193273528320*a^13*b^4*c^14*z^4 - 12
8849018880*a^12*b^6*c^13*z^4 - 16911433728*a^10*b^10*c^11*z^4 + 3523215360*
a^9*b^12*c^10*z^4 + 68719476736*a^15*c^16*z^4 - 47185920*a^7*b^16*c^5*k*z^3
 + 2621440*a^6*b^18*c^4*k*z^3 - 65536*a^5*b^20*c^3*k*z^3 + 171798691840*a^1
4*b^2*c^12*k*z^3 - 193273528320*a^13*b^4*c^11*k*z^3 + 128849018880*a^12*b^6
*c^10*k*z^3 + 16911433728*a^10*b^10*c^8*k*z^3 - 3523215360*a^9*b^12*c^7*k*z

^3 - 56371445760*a^11*b^8*c^9*k*z^3 + 503316480*a^8*b^14*c^6*k*z^3 - 687194
76736*a^15*c^13*k*z^3 + 1536*a*b^18*c^6*d*f*z^2 - 2571632640*a^9*b^5*c^11*d
*j*z^2 + 2548039680*a^9*b^3*c^13*d*h*z^2 + 2453667840*a^9*b^7*c^9*e*k*z^2 +
2181038080*a^12*b^3*c^10*i*k*z^2 - 6492782592*a^10*b^5*c^10*e*k*z^2 + 1509
949440*a^9*b^3*c^13*e*g*z^2 - 1401421824*a^8*b^5*c^12*d*h*z^2 - 1226833920*
a^9*b^8*c^8*g*k*z^2 - 1321205760*a^9*b^2*c^14*d*f*z^2 - 2793406464*a^11*b*c
^13*d*j*z^2 + 9563013120*a^11*b^3*c^11*e*k*z^2 + 890634240*a^8*b^7*c^10*d*j
*z^2 - 754974720*a^8*b^5*c^12*e*g*z^2 - 570425344*a^11*b^5*c^9*i*k*z^2 + 73
2168192*a^7*b^6*c^12*d*f*z^2 - 581959680*a^10*b^4*c^11*f*j*z^2 - 603979776*
a^10*b^2*c^13*e*i*z^2 + 534773760*a^11*b^3*c^11*h*j*z^2 - 558366720*a^8*b^9
*c^8*e*k*z^2 - 4781506560*a^11*b^4*c^10*g*k*z^2 - 2013265920*a^13*b*c^11*i*
k*z^2 - 456130560*a^9*b^4*c^12*f*h*z^2 + 384040960*a^9*b^6*c^10*f*j*z^2 - 2
64241152*a^10*b^7*c^8*i*k*z^2 + 390463488*a^7*b^7*c^11*d*h*z^2 + 279183360*
a^8*b^10*c^7*g*k*z^2 + 301989888*a^10*b^3*c^12*g*i*z^2 + 222822400*a^9*b^9*
c^7*i*k*z^2 - 366280704*a^6*b^8*c^11*d*f*z^2 - 330301440*a^8*b^4*c^13*d*f*z
^2 + 254017536*a^8*b^6*c^11*f*h*z^2 - 1887436800*a^10*b*c^14*d*h*z^2 + 1887
43680*a^10*b^2*c^13*f*h*z^2 - 185303040*a^7*b^9*c^9*d*j*z^2 - 117964800*a^1
0*b^5*c^10*h*j*z^2 - 6039797760*a^12*b*c^12*e*k*z^2 - 67502080*a^8*b^11*c^6
*i*k*z^2 + 121634816*a^11*b^2*c^12*f*j*z^2 + 188743680*a^7*b^7*c^11*e*g*z^2
- 115671040*a^8*b^8*c^9*f*j*z^2 + 125829120*a^8*b^6*c^11*e*i*z^2 + 1081344
0*a^7*b^13*c^5*i*k*z^2 + 76677120*a^7*b^11*c^7*e*k*z^2 - 38338560*a^7*b^12*
c^6*g*k*z^2 - 37355520*a^9*b^7*c^9*h*j*z^2 - 917504*a^6*b^15*c^4*i*k*z^2 +
32768*a^5*b^17*c^3*i*k*z^2 - 62914560*a^8*b^7*c^10*g*i*z^2 + 23101440*a^8*b
^9*c^8*h*j*z^2 - 4349952*a^7*b^11*c^7*h*j*z^2 + 2949120*a^6*b^14*c^5*g*k*z^
2 + 337920*a^6*b^13*c^6*h*j*z^2 - 98304*a^5*b^16*c^4*g*k*z^2 - 7680*a^5*b^1
5*c^5*h*j*z^2 - 61931520*a^7*b^8*c^10*f*h*z^2 + 23592960*a^7*b^9*c^9*g*i*z^
2 + 17940480*a^7*b^10*c^8*f*j*z^2 - 47185920*a^7*b^8*c^10*e*i*z^2 - 5898240
*a^6*b^13*c^6*e*k*z^2 - 3538944*a^6*b^11*c^8*g*i*z^2 - 1347584*a^6*b^12*c^7
*f*j*z^2 + 196608*a^5*b^15*c^5*e*k*z^2 + 196608*a^5*b^13*c^7*g*i*z^2 + 3584
0*a^5*b^14*c^6*f*j*z^2 + 96583680*a^5*b^10*c^10*d*f*z^2 + 23371776*a^6*b^11
*c^8*d*j*z^2 - 51609600*a^6*b^9*c^10*d*h*z^2 + 7077888*a^6*b^10*c^9*e*i*z^2
+ 6144000*a^6*b^10*c^9*f*h*z^2 - 1677312*a^5*b^13*c^7*d*j*z^2 - 393216*a^5
*b^12*c^8*e*i*z^2 + 61440*a^5*b^12*c^8*f*h*z^2 + 53760*a^4*b^15*c^6*d*j*z^2
- 46080*a^4*b^14*c^7*f*h*z^2 + 1536*a^3*b^16*c^6*f*h*z^2 - 23592960*a^6*b^
9*c^10*e*g*z^2 + 1179648*a^5*b^11*c^9*e*g*z^2 + 829440*a^4*b^13*c^8*d*h*z^2
+ 368640*a^5*b^11*c^9*d*h*z^2 - 105984*a^3*b^15*c^7*d*h*z^2 + 4608*a^2*b^1
7*c^6*d*h*z^2 - 15175680*a^4*b^12*c^9*d*f*z^2 + 1428480*a^3*b^14*c^8*d*f*z^
2 - 73728*a^2*b^16*c^7*d*f*z^2 + 4108320768*a^10*b^3*c^12*d*j*z^2 - 1207959
552*a^10*b*c^14*e*g*z^2 - 578813952*a^12*b*c^12*h*j*z^2 + 3246391296*a^10*b
^6*c^9*g*k*z^2 - 402653184*a^11*b*c^13*g*i*z^2 + 3019898880*a^12*b^2*c^11*g
*k*z^2 - 440401920*a^10*b*c^14*f^2*z^2 - 188743680*a^11*b*c^13*h^2*z^2 + 17
61607680*a^10*c^15*d*f*z^2 - 655360*a^6*b^18*c*k^2*z^2 - 94464*a*b^17*c^7*d
^2*z^2 + 6936330240*a^8*b^3*c^14*d^2*z^2 + 2464874496*a^6*b^7*c^12*d^2*z^2
- 3963617280*a^9*b*c^15*d^2*z^2 + 58007224320*a^13*b^4*c^8*k^2*z^2 + 149684
22400*a^11*b^8*c^6*k^2*z^2 + 805306368*a^11*c^14*e*i*z^2 - 35966156800*a^12

*b^6*c^7*k^2*z^2 + 419430400*a^12*c^13*f*j*z^2 - 1509949440*a^9*b^2*c^14*e^
2*z^2 + 251658240*a^11*c^14*f*h*z^2 - 56874762240*a^14*b^2*c^9*k^2*z^2 - 54
00428544*a^7*b^5*c^13*d^2*z^2 + 890470400*a^9*b^12*c^4*k^2*z^2 + 754974720*
a^8*b^4*c^13*e^2*z^2 - 730054656*a^5*b^9*c^11*d^2*z^2 + 477102080*a^12*b^3*
c^10*j^2*z^2 + 477102080*a^9*b^3*c^13*f^2*z^2 - 377487360*a^9*b^4*c^12*g^2*
z^2 + 301989888*a^10*b^2*c^13*g^2*z^2 - 174325760*a^11*b^5*c^9*j^2*z^2 - 12
6156800*a^8*b^14*c^3*k^2*z^2 + 188743680*a^8*b^6*c^11*g^2*z^2 + 141557760*a
^10*b^3*c^12*h^2*z^2 - 174325760*a^8*b^5*c^12*f^2*z^2 - 188743680*a^7*b^6*c
^12*e^2*z^2 - 4350935040*a^10*b^10*c^5*k^2*z^2 + 146165760*a^4*b^11*c^10*d^
2*z^2 - 50331648*a^10*b^4*c^11*i^2*z^2 + 11796480*a^7*b^16*c^2*k^2*z^2 - 33
554432*a^11*b^2*c^12*i^2*z^2 + 11206656*a^10*b^7*c^8*j^2*z^2 + 8929280*a^9*
b^9*c^7*j^2*z^2 + 20971520*a^9*b^6*c^10*i^2*z^2 - 2600960*a^8*b^11*c^6*j^2*
z^2 + 291840*a^7*b^13*c^5*j^2*z^2 - 14080*a^6*b^15*c^4*j^2*z^2 + 256*a^5*b^
17*c^3*j^2*z^2 - 47185920*a^7*b^8*c^10*g^2*z^2 - 26542080*a^8*b^7*c^10*h^2*
z^2 - 2752512*a^7*b^10*c^8*i^2*z^2 + 2621440*a^8*b^8*c^9*i^2*z^2 + 524288*a
^6*b^12*c^7*i^2*z^2 - 32768*a^5*b^14*c^6*i^2*z^2 + 9584640*a^7*b^9*c^9*h^2*
z^2 - 2359296*a^9*b^5*c^11*h^2*z^2 - 1290240*a^6*b^11*c^8*h^2*z^2 + 46080*a
^5*b^13*c^7*h^2*z^2 + 2304*a^4*b^15*c^6*h^2*z^2 + 5898240*a^6*b^10*c^9*g^2*
z^2 - 294912*a^5*b^12*c^8*g^2*z^2 + 11206656*a^7*b^7*c^11*f^2*z^2 + 8929280
*a^6*b^9*c^10*f^2*z^2 + 23592960*a^6*b^8*c^11*e^2*z^2 - 2600960*a^5*b^11*c^
9*f^2*z^2 + 291840*a^4*b^13*c^8*f^2*z^2 - 14080*a^3*b^15*c^7*f^2*z^2 + 256*
a^2*b^17*c^6*f^2*z^2 - 19860480*a^3*b^13*c^9*d^2*z^2 - 1179648*a^5*b^10*c^1
0*e^2*z^2 + 1771776*a^2*b^15*c^8*d^2*z^2 - 440401920*a^13*b*c^11*j^2*z^2 +
1207959552*a^10*c^15*e^2*z^2 + 134217728*a^12*c^13*i^2*z^2 + 25769803776*a^
15*c^10*k^2*z^2 + 16384*a^5*b^20*k^2*z^2 + 2304*b^19*c^6*d^2*z^2 + 16515072
0*a^9*b*c^12*d*g*j*z + 23592960*a^10*b*c^11*g*h*j*z + 169869312*a^7*b*c^14*
d*e*f*z + 99090432*a^8*b*c^13*d*g*h*z - 3145728*a^9*b*c^12*f*h*i*z + 566231
04*a^8*b*c^13*d*f*i*z - 1536*a*b^18*c^3*d*f*k*z - 9437184*a^8*b*c^13*e*f*h*
z + 1536*a*b^15*c^6*d*f*i*z - 4608*a*b^14*c^7*d*f*g*z + 9216*a*b^13*c^8*d*e
*f*z + 2173501440*a^9*b^5*c^8*d*j*k*z - 1987706880*a^9*b^3*c^10*d*h*k*z + 1
121255424*a^8*b^5*c^9*d*h*k*z + 861143040*a^8*b^4*c^10*d*f*k*z - 859963392*
a^7*b^6*c^9*d*f*k*z - 780779520*a^8*b^7*c^7*d*j*k*z - 754974720*a^9*b^3*c^1
0*e*g*k*z + 2222456832*a^11*b*c^10*d*j*k*z - 454164480*a^11*b^3*c^8*h*j*k*z
 + 377487360*a^8*b^5*c^9*e*g*k*z + 290979840*a^10*b^4*c^8*f*j*k*z + 3810263
04*a^6*b^8*c^8*d*f*k*z + 412876800*a^8*b^2*c^12*d*e*j*z + 301989888*a^10*b^
2*c^10*e*i*k*z - 320421888*a^7*b^7*c^8*d*h*k*z + 185794560*a^10*b^5*c^7*h*j
*k*z - 192020480*a^9*b^6*c^7*f*j*k*z + 190709760*a^9*b^4*c^9*f*h*k*z - 1509
94944*a^10*b^3*c^9*g*i*k*z + 168990720*a^7*b^9*c^6*d*j*k*z + 235929600*a^9*
b^2*c^11*d*f*k*z - 206438400*a^8*b^3*c^11*d*g*j*z - 206438400*a^7*b^4*c^11*
d*e*j*z - 101646336*a^8*b^6*c^8*f*h*k*z - 29245440*a^9*b^7*c^6*h*j*k*z - 60
817408*a^11*b^2*c^9*f*j*k*z + 57835520*a^8*b^8*c^6*f*j*k*z + 219414528*a^7*
b^2*c^13*d*e*h*z - 70778880*a^10*b^2*c^10*f*h*k*z + 677376*a^7*b^11*c^4*h*j
*k*z - 645120*a^8*b^9*c^5*h*j*k*z - 53760*a^6*b^13*c^3*h*j*k*z + 31457280*a
^8*b^7*c^7*g*i*k*z - 62914560*a^8*b^6*c^8*e*i*k*z - 94371840*a^7*b^7*c^8*e*
g*k*z - 221773824*a^6*b^3*c^13*d*e*f*z + 82575360*a^9*b^2*c^11*d*i*j*z + 11

796480*a^10*b^2*c^10*h*i*j*z - 11796480*a^7*b^9*c^6*g*i*k*z - 8970240*a^7*b
^10*c^5*f*j*k*z + 103219200*a^7*b^5*c^10*d*g*j*z - 2457600*a^8*b^6*c^8*h*i*
j*z + 1769472*a^6*b^11*c^5*g*i*k*z + 921600*a^7*b^8*c^7*h*i*j*z + 673792*a^
6*b^12*c^4*f*j*k*z - 138240*a^6*b^10*c^6*h*i*j*z - 98304*a^5*b^13*c^4*g*i*k
*z - 17920*a^5*b^14*c^3*f*j*k*z + 7680*a^5*b^12*c^5*h*i*j*z - 97136640*a^5*
b^10*c^7*d*f*k*z - 29491200*a^9*b^3*c^10*g*h*j*z + 58982400*a^9*b^2*c^11*e*
h*j*z + 23592960*a^7*b^8*c^7*e*i*k*z - 22169088*a^6*b^11*c^5*d*j*k*z + 2138
1120*a^7*b^8*c^7*f*h*k*z + 14745600*a^8*b^5*c^9*g*h*j*z + 42854400*a^6*b^9*
c^7*d*h*k*z - 109707264*a^7*b^3*c^12*d*g*h*z - 3686400*a^7*b^7*c^8*g*h*j*z
- 3538944*a^6*b^10*c^6*e*i*k*z + 1645056*a^5*b^13*c^4*d*j*k*z - 890880*a^6*
b^10*c^6*f*h*k*z + 460800*a^6*b^9*c^7*g*h*j*z - 330240*a^5*b^12*c^5*f*h*k*z
 + 196608*a^5*b^12*c^5*e*i*k*z - 53760*a^4*b^15*c^3*d*j*k*z + 46080*a^4*b^1
4*c^4*f*h*k*z - 23040*a^5*b^11*c^6*g*h*j*z - 1536*a^3*b^16*c^3*f*h*k*z - 29
491200*a^8*b^4*c^10*e*h*j*z - 17203200*a^7*b^6*c^9*d*i*j*z + 11796480*a^6*b
^9*c^7*e*g*k*z + 110886912*a^6*b^4*c^12*d*f*g*z + 7372800*a^7*b^6*c^9*e*h*j
*z + 40108032*a^8*b^2*c^12*d*h*i*z + 6451200*a^6*b^8*c^8*d*i*j*z + 2359296*
a^8*b^3*c^11*f*h*i*z - 967680*a^5*b^10*c^7*d*i*j*z - 921600*a^6*b^8*c^8*e*h
*j*z - 829440*a^4*b^13*c^5*d*h*k*z - 589824*a^5*b^11*c^6*e*g*k*z - 491520*a
^6*b^7*c^9*f*h*i*z + 184320*a^5*b^9*c^8*f*h*i*z + 105984*a^3*b^15*c^4*d*h*k
*z + 69120*a^5*b^11*c^6*d*h*k*z + 53760*a^4*b^12*c^6*d*i*j*z + 46080*a^5*b^
10*c^7*e*h*j*z - 27648*a^4*b^11*c^7*f*h*i*z - 4608*a^2*b^17*c^3*d*h*k*z + 1
536*a^3*b^13*c^6*f*h*i*z - 25804800*a^6*b^7*c^9*d*g*j*z - 88473600*a^6*b^4*
c^12*d*e*h*z + 51609600*a^6*b^6*c^10*d*e*j*z - 84934656*a^7*b^2*c^13*d*f*g*
z + 117964800*a^5*b^5*c^12*d*e*f*z + 15160320*a^4*b^12*c^6*d*f*k*z - 456130
56*a^7*b^3*c^12*d*f*i*z + 44236800*a^6*b^5*c^11*d*g*h*z - 10321920*a^6*b^6*
c^10*d*h*i*z + 7077888*a^7*b^4*c^11*d*h*i*z - 5898240*a^7*b^4*c^11*f*g*h*z
+ 4718592*a^8*b^2*c^12*f*g*h*z + 3225600*a^5*b^9*c^8*d*g*j*z + 2949120*a^6*
b^6*c^10*f*g*h*z + 2396160*a^5*b^8*c^9*d*h*i*z - 1428480*a^3*b^14*c^5*d*f*k
*z - 737280*a^5*b^8*c^9*f*g*h*z - 161280*a^4*b^11*c^7*d*g*j*z + 92160*a^4*b
^10*c^8*f*g*h*z + 73728*a^2*b^16*c^4*d*f*k*z - 50688*a^3*b^12*c^7*d*h*i*z -
 27648*a^4*b^10*c^8*d*h*i*z - 4608*a^3*b^12*c^7*f*g*h*z + 4608*a^2*b^14*c^6
*d*h*i*z - 58982400*a^5*b^6*c^11*d*f*g*z + 11796480*a^7*b^3*c^12*e*f*h*z +
8847360*a^5*b^7*c^10*d*f*i*z - 6635520*a^5*b^7*c^10*d*g*h*z - 6451200*a^5*b
^8*c^9*d*e*j*z - 5898240*a^6*b^5*c^11*e*f*h*z - 3809280*a^4*b^9*c^9*d*f*i*z
 + 2359296*a^6*b^5*c^11*d*f*i*z + 1474560*a^5*b^7*c^10*e*f*h*z + 681984*a^3
*b^11*c^8*d*f*i*z + 322560*a^4*b^10*c^8*d*e*j*z - 276480*a^4*b^9*c^9*d*g*h*
z - 184320*a^4*b^9*c^9*e*f*h*z + 179712*a^3*b^11*c^8*d*g*h*z - 55296*a^2*b^
13*c^7*d*f*i*z - 13824*a^2*b^13*c^7*d*g*h*z + 9216*a^3*b^11*c^8*e*f*h*z + 1
6220160*a^4*b^8*c^10*d*f*g*z + 13271040*a^5*b^6*c^11*d*e*h*z - 2396160*a^3*
b^10*c^9*d*f*g*z + 552960*a^4*b^8*c^10*d*e*h*z - 359424*a^3*b^10*c^9*d*e*h*
z + 175104*a^2*b^12*c^8*d*f*g*z + 27648*a^2*b^12*c^8*d*e*h*z - 32440320*a^4
*b^7*c^11*d*e*f*z + 4792320*a^3*b^9*c^10*d*e*f*z - 350208*a^2*b^11*c^9*d*e*
f*z + 1439170560*a^10*b*c^11*d*h*k*z - 3361603584*a^10*b^3*c^9*d*j*k*z + 60
3979776*a^10*b*c^11*e*g*k*z + 407371776*a^12*b*c^9*h*j*k*z + 201326592*a^11
*b*c^10*g*i*k*z + 346816512*a^7*b*c^14*d^2*g*z + 129761280*a^11*b*c^10*h^2*

k*z + 121896960*a^10*b*c^11*f^2*k*z + 458752*a^6*b^15*c*i*k^2*z + 19660800*
a^11*b*c^10*g*j^2*z + 49152*a^5*b^16*c*g*k^2*z + 7077888*a^9*b*c^12*g*h^2*z
 + 94464*a*b^17*c^4*d^2*k*z - 19660800*a^8*b*c^13*f^2*g*z - 66816*a*b^14*c^
7*d^2*i*z + 214272*a*b^13*c^8*d^2*g*z - 428544*a*b^12*c^9*d^2*e*z + 2390753
280*a^11*b^4*c^7*g*k^2*z - 2411421696*a^6*b^7*c^9*d^2*k*z - 6603079680*a^8*
b^3*c^11*d^2*k*z + 3715891200*a^9*b*c^12*d^2*k*z - 880803840*a^10*c^12*d*f*
k*z - 1623195648*a^10*b^6*c^6*g*k^2*z - 402653184*a^11*c^11*e*i*k*z - 15099
49440*a^12*b^2*c^8*g*k^2*z - 209715200*a^12*c^10*f*j*k*z - 330301440*a^9*c^
13*d*e*j*z + 3019898880*a^12*b*c^9*e*k^2*z - 125829120*a^11*c^11*f*h*k*z -
110100480*a^10*c^12*d*i*j*z - 198180864*a^8*c^14*d*e*h*z - 15728640*a^11*c^
11*h*i*j*z - 1226833920*a^9*b^7*c^6*e*k^2*z - 47185920*a^10*c^12*e*h*j*z -
66060288*a^9*c^13*d*h*i*z - 1090519040*a^12*b^3*c^7*i*k^2*z + 1022754816*a^
6*b^2*c^14*d^2*e*z + 5216108544*a^7*b^5*c^10*d^2*k*z + 754974720*a^9*b^2*c^
11*e^2*k*z + 721529856*a^5*b^9*c^8*d^2*k*z + 613416960*a^9*b^8*c^5*g*k^2*z
- 642318336*a^5*b^4*c^13*d^2*e*z - 4781506560*a^11*b^3*c^8*e*k^2*z - 398131
200*a^12*b^3*c^7*j^2*k*z - 511377408*a^6*b^3*c^13*d^2*g*z - 377487360*a^8*b
^4*c^10*e^2*k*z + 285212672*a^11*b^5*c^6*i*k^2*z + 199065600*a^11*b^5*c^6*j
^2*k*z + 279183360*a^8*b^9*c^5*e*k^2*z + 321159168*a^5*b^5*c^12*d^2*g*z + 1
88743680*a^9*b^4*c^9*g^2*k*z + 132120576*a^10*b^7*c^5*i*k^2*z - 150994944*a
^10*b^2*c^10*g^2*k*z - 111411200*a^9*b^9*c^4*i*k^2*z - 126812160*a^10*b^3*c
^9*h^2*k*z + 225312768*a^7*b^2*c^13*d^2*i*z - 139591680*a^8*b^10*c^4*g*k^2*
z - 49766400*a^10*b^7*c^5*j^2*k*z - 145463040*a^4*b^11*c^7*d^2*k*z - 943718
40*a^8*b^6*c^8*g^2*k*z + 223395840*a^4*b^6*c^12*d^2*e*z + 33751040*a^8*b^11
*c^3*i*k^2*z - 78970880*a^9*b^3*c^10*f^2*k*z + 94371840*a^7*b^6*c^9*e^2*k*z
 + 25165824*a^10*b^4*c^8*i^2*k*z + 6220800*a^9*b^9*c^4*j^2*k*z + 39223296*a
^9*b^5*c^8*h^2*k*z - 311040*a^8*b^11*c^3*j^2*k*z + 16777216*a^11*b^2*c^9*i^
2*k*z - 10485760*a^9*b^6*c^7*i^2*k*z - 5406720*a^7*b^13*c^2*i*k^2*z + 13762
56*a^7*b^10*c^5*i^2*k*z - 1310720*a^8*b^8*c^6*i^2*k*z - 262144*a^6*b^12*c^4
*i^2*k*z + 16384*a^5*b^14*c^3*i^2*k*z + 10354688*a^11*b^2*c^9*i*j^2*z + 235
92960*a^7*b^8*c^7*g^2*k*z + 38559744*a^7*b^7*c^8*f^2*k*z + 19169280*a^7*b^1
2*c^3*g*k^2*z - 2048000*a^9*b^6*c^7*i*j^2*z - 1520640*a^7*b^9*c^6*h^2*k*z -
 1105920*a^8*b^7*c^7*h^2*k*z + 849920*a^8*b^8*c^6*i*j^2*z - 393216*a^10*b^4
*c^8*i*j^2*z + 195840*a^6*b^11*c^5*h^2*k*z - 145920*a^7*b^10*c^5*i*j^2*z +
11520*a^5*b^13*c^4*h^2*k*z + 11008*a^6*b^12*c^4*i*j^2*z - 2304*a^4*b^15*c^3
*h^2*k*z - 256*a^5*b^14*c^3*i*j^2*z - 25362432*a^10*b^3*c^9*g*j^2*z - 24739
840*a^8*b^5*c^9*f^2*k*z - 38338560*a^7*b^11*c^4*e*k^2*z - 2949120*a^6*b^10*
c^6*g^2*k*z - 1474560*a^6*b^14*c^2*g*k^2*z + 50724864*a^10*b^2*c^10*e*j^2*z
 + 147456*a^5*b^12*c^5*g^2*k*z - 15150080*a^6*b^9*c^7*f^2*k*z + 13271040*a^
9*b^5*c^8*g*j^2*z - 111697920*a^4*b^7*c^11*d^2*g*z - 3563520*a^8*b^7*c^7*g*
j^2*z + 3538944*a^9*b^2*c^11*h^2*i*z + 2912000*a^5*b^11*c^6*f^2*k*z - 73728
0*a^7*b^6*c^9*h^2*i*z + 506880*a^7*b^9*c^6*g*j^2*z - 291840*a^4*b^13*c^5*f^
2*k*z + 276480*a^6*b^8*c^8*h^2*i*z - 41472*a^5*b^10*c^7*h^2*i*z - 34560*a^6
*b^11*c^5*g*j^2*z + 14080*a^3*b^15*c^4*f^2*k*z + 2304*a^4*b^12*c^6*h^2*i*z
+ 768*a^5*b^13*c^4*g*j^2*z - 256*a^2*b^17*c^3*f^2*k*z - 11796480*a^6*b^8*c^
8*e^2*k*z - 26542080*a^9*b^4*c^9*e*j^2*z + 19837440*a^3*b^13*c^6*d^2*k*z +

2949120*a^6*b^13*c^3*e*k^2*z + 589824*a^5*b^10*c^7*e^2*k*z - 98304*a^5*b^15
*c^2*e*k^2*z - 10354688*a^8*b^2*c^12*f^2*i*z - 43646976*a^6*b^4*c^12*d^2*i*
z - 8847360*a^8*b^3*c^11*g*h^2*z + 7127040*a^8*b^6*c^8*e*j^2*z + 4423680*a^
7*b^5*c^10*g*h^2*z + 2048000*a^6*b^6*c^10*f^2*i*z - 1771776*a^2*b^15*c^5*d^
2*k*z - 1105920*a^6*b^7*c^9*g*h^2*z - 1013760*a^7*b^8*c^7*e*j^2*z - 849920*
a^5*b^8*c^9*f^2*i*z + 393216*a^7*b^4*c^11*f^2*i*z + 145920*a^4*b^10*c^8*f^2
*i*z + 138240*a^5*b^9*c^8*g*h^2*z + 69120*a^6*b^10*c^6*e*j^2*z - 11008*a^3*
b^12*c^7*f^2*i*z - 6912*a^4*b^11*c^7*g*h^2*z - 1536*a^5*b^12*c^5*e*j^2*z +
256*a^2*b^14*c^6*f^2*i*z - 32587776*a^5*b^6*c^11*d^2*i*z + 25362432*a^7*b^3
*c^12*f^2*g*z + 21657600*a^4*b^8*c^10*d^2*i*z + 17694720*a^8*b^2*c^12*e*h^2
*z - 50724864*a^7*b^2*c^13*e*f^2*z - 13271040*a^6*b^5*c^11*f^2*g*z - 884736
0*a^7*b^4*c^11*e*h^2*z - 5810688*a^3*b^10*c^9*d^2*i*z + 3563520*a^5*b^7*c^1
0*f^2*g*z + 2211840*a^6*b^6*c^10*e*h^2*z + 845568*a^2*b^12*c^8*d^2*i*z - 50
6880*a^4*b^9*c^9*f^2*g*z - 276480*a^5*b^8*c^9*e*h^2*z + 34560*a^3*b^11*c^8*
f^2*g*z + 13824*a^4*b^10*c^8*e*h^2*z - 768*a^2*b^13*c^7*f^2*g*z + 26542080*
a^6*b^4*c^12*e*f^2*z + 23362560*a^3*b^9*c^10*d^2*g*z - 46725120*a^3*b^8*c^1
1*d^2*e*z - 7127040*a^5*b^6*c^11*e*f^2*z - 2965248*a^2*b^11*c^9*d^2*g*z + 1
013760*a^4*b^8*c^10*e*f^2*z - 69120*a^3*b^10*c^9*e*f^2*z + 1536*a^2*b^12*c^
8*e*f^2*z + 5930496*a^2*b^10*c^10*d^2*e*z + 1006632960*a^13*b*c^8*i*k^2*z +
3246391296*a^10*b^5*c^7*e*k^2*z + 318504960*a^13*b*c^8*j^2*k*z + 61538304*
a^10*b^10*c^2*k^3*z - 603979776*a^10*c^12*e^2*k*z - 693633024*a^7*c^15*d^2*
e*z - 231211008*a^8*c^14*d^2*i*z - 67108864*a^12*c^10*i^2*k*z - 13107200*a^
12*c^10*i*j^2*z - 16384*a^5*b^17*i*k^2*z - 39321600*a^11*c^11*e*j^2*z - 471
8592*a^10*c^12*h^2*i*z - 2304*b^19*c^3*d^2*k*z + 13107200*a^9*c^13*f^2*i*z
+ 2304*b^16*c^6*d^2*i*z - 14155776*a^9*c^13*e*h^2*z + 39321600*a^8*c^14*e*f
^2*z - 4833280*a^9*b^12*c*k^3*z - 6912*b^15*c^7*d^2*g*z + 6962544640*a^14*b
^2*c^6*k^3*z + 13824*b^14*c^8*d^2*e*z + 1876951040*a^12*b^6*c^4*k^3*z - 484
4421120*a^13*b^4*c^5*k^3*z - 437780480*a^11*b^8*c^3*k^3*z - 4294967296*a^15
*c^7*k^3*z + 163840*a^8*b^14*k^3*z + 6144000*a^10*b*c^8*f*i*j*k - 5898240*a
^10*b*c^8*g*h*j*k - 41287680*a^9*b*c^9*d*g*j*k + 4472832*a^9*b*c^9*f*h*i*k
+ 18432000*a^9*b*c^9*e*f*j*k + 3391488*a^8*b*c^10*e*h*i*j + 1228800*a^8*b*c
^10*f*g*i*j - 24772608*a^8*b*c^10*d*g*h*k + 13418496*a^8*b*c^10*e*f*h*k + 1
1649024*a^8*b*c^10*d*f*i*k + 737280*a^7*b*c^11*f*g*h*i - 768*a*b^15*c^3*d*f
*i*k - 19307520*a^7*b*c^11*d*f*h*j + 16367616*a^7*b*c^11*d*e*i*j + 3686400*
a^7*b*c^11*e*f*g*j + 34947072*a^7*b*c^11*d*e*f*k + 2304*a*b^14*c^4*d*f*g*k
- 180*a*b^13*c^5*d*f*h*j + 11059200*a^6*b*c^12*d*e*h*i + 5160960*a^6*b*c^12
*d*f*g*i + 2211840*a^6*b*c^12*e*f*g*h - 4608*a*b^13*c^5*d*e*f*k - 2304*a*b^
11*c^7*d*f*g*i + 4608*a*b^10*c^8*d*e*f*i + 15482880*a^5*b*c^13*d*e*f*g - 13
824*a*b^9*c^9*d*e*f*g - 225976320*a^8*b^2*c^9*d*e*j*k + 112988160*a^8*b^3*c
^8*d*g*j*k - 11427840*a^10*b^2*c^7*h*i*j*k - 4177920*a^9*b^4*c^6*h*i*j*k +
1399296*a^8*b^6*c^5*h*i*j*k - 26880*a^6*b^10*c^3*h*i*j*k + 16128*a^7*b^8*c^
4*h*i*j*k - 61562880*a^9*b^2*c^8*d*i*j*k + 20090880*a^9*b^3*c^7*g*h*j*k + 1
19623680*a^7*b^4*c^8*d*e*j*k + 10485760*a^9*b^3*c^7*f*i*j*k - 40181760*a^9*
b^2*c^8*e*h*j*k - 3778560*a^8*b^5*c^6*g*h*j*k - 137797632*a^7*b^2*c^10*d*e*
h*k - 1248768*a^7*b^7*c^5*f*i*j*k + 229376*a^6*b^9*c^4*f*i*j*k + 220160*a^8

*b^5*c^6*f*i*j*k - 209664*a^7*b^7*c^5*g*h*j*k + 80640*a^6*b^9*c^4*g*h*j*k - 8960*a^5*b^11*c^3*f*i*j*k - 59811840*a^7*b^5*c^7*d*g*j*k + 53084160*a^8*b^2*c^9*e*g*i*k - 11120640*a^8*b^4*c^7*f*g*j*k + 10455552*a^7*b^6*c^6*d*i*j*k - 9216000*a^9*b^2*c^8*f*g*j*k + 7557120*a^8*b^4*c^7*e*h*j*k + 7397376*a^8*b^3*c^8*f*h*i*k + 5230080*a^7*b^6*c^6*f*g*j*k - 37675008*a^8*b^2*c^9*d*h*i*k - 3633408*a^6*b^8*c^5*d*i*j*k + 2211840*a^8*b^4*c^7*d*i*j*k + 68898816*a^7*b^3*c^9*d*g*h*k - 1695744*a^8*b^2*c^9*g*h*i*j - 1400832*a^7*b^4*c^8*g*h*i*j + 967680*a^7*b^5*c^7*f*h*i*k - 783360*a^6*b^7*c^6*f*h*i*k - 741888*a^6*b^8*c^5*f*g*j*k + 499968*a^5*b^10*c^4*d*i*j*k + 419328*a^7*b^6*c^6*e*h*j*k - 253440*a^6*b^6*c^7*g*h*i*j - 161280*a^6*b^8*c^5*e*h*j*k + 42240*a^5*b^9*c^5*f*h*i*k + 26880*a^5*b^10*c^4*f*g*j*k - 26880*a^4*b^12*c^3*d*i*j*k + 13824*a^4*b^11*c^4*f*h*i*k + 11520*a^5*b^8*c^6*g*h*i*j - 768*a^3*b^13*c^3*f*h*i*k + 22241280*a^8*b^3*c^8*e*f*j*k + 14222592*a^6*b^7*c^6*d*g*j*k - 10460160*a^7*b^5*c^7*e*f*j*k + 8847360*a^7*b^4*c^8*e*g*i*k - 7741440*a^7*b^4*c^8*f*g*h*k - 7077888*a^6*b^6*c^7*e*g*i*k + 6935040*a^6*b^6*c^7*d*h*i*k - 6709248*a^8*b^2*c^9*f*g*h*k - 3612672*a^7*b^4*c^8*d*h*i*k + 2801664*a^7*b^3*c^9*e*h*i*j + 2506752*a^7*b^3*c^9*f*g*i*j + 2419200*a^6*b^6*c^7*f*g*h*k - 1661184*a^5*b^9*c^5*d*g*j*k + 1483776*a^6*b^7*c^6*e*f*j*k - 1463040*a^5*b^8*c^6*d*h*i*k + 884736*a^5*b^8*c^6*e*g*i*k + 838656*a^6*b^5*c^8*f*g*i*j + 506880*a^6*b^5*c^8*e*h*i*j + 80640*a^4*b^11*c^4*d*g*j*k - 53760*a^5*b^9*c^5*e*f*j*k - 53760*a^5*b^7*c^7*f*g*i*j - 46080*a^4*b^10*c^5*f*g*h*k - 34560*a^5*b^8*c^6*f*g*h*k + 25344*a^3*b^12*c^4*d*h*i*k - 23040*a^5*b^7*c^7*e*h*i*j + 13824*a^4*b^10*c^5*d*h*i*k + 2304*a^3*b^12*c^4*f*g*h*k - 2304*a^2*b^14*c^3*d*h*i*k - 29030400*a^6*b^5*c^8*d*g*h*k + 28606464*a^7*b^3*c^9*d*f*i*k - 28445184*a^6*b^6*c^7*d*e*j*k + 58060800*a^6*b^4*c^9*d*e*h*k + 15482880*a^7*b^3*c^9*e*f*h*k - 8183808*a^7*b^2*c^10*d*g*i*j - 6718464*a^6*b^5*c^8*d*f*i*k - 5087232*a^7*b^2*c^10*e*g*h*j - 5013504*a^7*b^2*c^10*e*f*i*j - 4838400*a^6*b^5*c^8*e*f*h*k + 4112640*a^5*b^7*c^7*d*g*h*k - 3663360*a^5*b^7*c^7*d*f*i*k + 3322368*a^5*b^8*c^6*d*e*j*k - 2285568*a^6*b^4*c^9*d*g*i*j + 1896960*a^4*b^9*c^6*d*f*i*k + 1843200*a^6*b^3*c^10*f*g*h*i - 1677312*a^6*b^4*c^9*e*f*i*j - 1658880*a^6*b^4*c^9*e*g*h*j + 68345856*a^6*b^3*c^10*d*e*f*k + 783360*a^5*b^5*c^9*f*g*h*i + 741888*a^5*b^6*c^8*d*g*i*j - 34172928*a^6*b^4*c^9*d*f*g*k - 340992*a^3*b^11*c^5*d*f*i*k - 161280*a^4*b^10*c^5*d*e*j*k + 138240*a^4*b^9*c^6*d*g*h*k + 107520*a^5*b^6*c^8*e*f*i*j + 92160*a^4*b^9*c^6*e*f*h*k - 89856*a^3*b^11*c^5*d*g*h*k - 80640*a^4*b^8*c^7*d*g*i*j + 69120*a^5*b^7*c^7*e*f*h*k + 69120*a^5*b^6*c^8*e*g*h*j + 27648*a^2*b^13*c^4*d*f*i*k + 18432*a^4*b^7*c^8*f*g*h*i + 6912*a^2*b^13*c^4*d*g*h*k - 4608*a^3*b^11*c^5*e*f*h*k - 2304*a^3*b^9*c^7*f*g*h*i + 27164160*a^5*b^6*c^8*d*f*g*k - 22164480*a^6*b^3*c^10*d*f*h*j - 54328320*a^5*b^5*c^9*d*e*f*k - 17473536*a^7*b^2*c^10*d*f*g*k - 8225280*a^5*b^6*c^8*d*e*h*k - 8087040*a^4*b^8*c^7*d*f*g*k + 5677056*a^6*b^3*c^10*e*f*g*j - 5529600*a^6*b^2*c^11*d*g*h*i + 4571136*a^6*b^3*c^10*d*e*i*j - 3686400*a^6*b^2*c^11*e*f*h*i + 2805120*a^5*b^5*c^9*d*f*h*j - 2211840*a^5*b^4*c^10*d*g*h*i - 1566720*a^5*b^4*c^10*e*f*h*i - 1483776*a^5*b^5*c^9*d*e*i*j + 1198080*a^3*b^10*c^6*d*f*g*k + 437184*a^4*b^7*c^8*d*f*h*j - 322560*a^5*b^5*c^9*e*f*g*j + 317952*a^4*b^6*c^9*d*g*h*i - 276480*a^4*b^8*c^7*d*e*h*k +

179712*a^3*b^10*c^6*d*e*h*k + 161280*a^4*b^7*c^8*d*e*i*j - 146268*a^3*b^9*c^7*d*f*h*j - 87552*a^2*b^12*c^5*d*f*g*k - 36864*a^4*b^6*c^9*e*f*h*i - 13824*a^2*b^12*c^5*d*e*h*k + 9360*a^2*b^11*c^6*d*f*h*j + 6912*a^3*b^8*c^8*d*g*h*i - 6912*a^2*b^10*c^7*d*g*h*i + 4608*a^3*b^8*c^8*e*f*h*i - 24551424*a^6*b^2*c^11*d*e*g*j + 16174080*a^4*b^7*c^8*d*e*f*k + 5419008*a^5*b^4*c^10*d*e*g*j + 5160960*a^5*b^3*c^11*d*f*g*i + 4423680*a^5*b^3*c^11*e*f*g*h + 4423680*a^5*b^3*c^11*d*e*h*i - 2396160*a^3*b^9*c^7*d*e*f*k - 635904*a^4*b^5*c^10*d*e*h*i - 483840*a^4*b^6*c^9*d*e*g*j - 354816*a^3*b^7*c^9*d*f*g*i + 322560*a^4*b^5*c^10*d*f*g*i + 175104*a^2*b^11*c^6*d*e*f*k + 138240*a^4*b^5*c^10*e*f*g*h + 59904*a^2*b^9*c^8*d*f*g*i - 13824*a^3*b^7*c^9*e*f*g*h - 13824*a^3*b^7*c^9*d*e*h*i + 13824*a^2*b^9*c^8*d*e*h*i - 16588800*a^5*b^2*c^12*d*e*g*h - 10321920*a^5*b^2*c^12*d*e*f*i + 1658880*a^4*b^4*c^11*d*e*g*h + 709632*a^3*b^6*c^10*d*e*f*i - 645120*a^4*b^4*c^11*d*e*f*i + 124416*a^3*b^6*c^10*d*e*g*h - 119808*a^2*b^8*c^9*d*e*f*i - 41472*a^2*b^8*c^9*d*e*g*h + 7741440*a^4*b^3*c^12*d*e*f*g - 2903040*a^3*b^5*c^11*d*e*f*g + 387072*a^2*b^7*c^10*d*e*f*g -
381026304*a^11*b*c^7*d*j*k^2 - 241827840*a^10*b*c^8*d*h*k^2 - 65667072*a^12*b*c^6*h*j*k^2 - 169344*a^7*b^11*c*h*j*k^2 - 25165824*a^11*b*c^7*g*i*k^2 - 4915200*a^11*b*c^7*g*j^2*k - 53084160*a^8*b*c^10*e^2*i*k - 75497472*a^10*b*c^8*e*g*k^2 - 86704128*a^7*b*c^11*d^2*g*k + 565248*a^9*b*c^9*h*i^2*j - 168448*a^6*b^12*c*f*j*k^2 - 24576*a^5*b^13*c*g*i*k^2 - 1769472*a^9*b*c^9*g*h^2*k - 17694720*a^9*b*c^9*e*i^2*k - 411264*a^5*b^13*c*d*j*k^2 - 11520*a^4*b^14*c*f*h*k^2 + 4915200*a^8*b*c^10*f^2*g*k + 2580480*a^9*b*c^9*e*i*j^2 - 2496000*a^9*b*c^9*f*h*j^2 - 1543680*a^8*b*c^10*f*h^2*j + 33408*a*b^14*c^4*d^2*i*k - 59512320*a^6*b*c^12*d^2*f*j + 5087232*a^7*b*c^11*e^2*h*j + 2727936*a^8*b*c^10*d*i^2*j - 26496*a^3*b^15*c*d*h*k^2 + 1105920*a^7*b*c^11*e*h^2*i - 107136*a*b^13*c^5*d^2*g*k + 10260*a*b^12*c^6*d^2*h*j - 10616832*a^6*b*c^12*e^2*g*i - 3538944*a^7*b*c^11*e*g*i^2 + 1843200*a^7*b*c^11*d*h*i^2 - 18432*a^2*b^16*c*d*f*k^2 - 15552000*a^8*b*c^10*d*f*j^2 + 24551424*a^6*b*c^12*d*e^2*j - 37062144*a^5*b*c^13*d^2*f*h + 2580480*a^6*b*c^12*e*f^2*i + 214272*a*b^12*c^6*d^2*e*k + 65664*a*b^10*c^8*d^2*g*i - 25074*a*b^11*c^7*d^2*f*j + 420*a*b^12*c^6*d*f^2*j + 6*a*b^15*c^3*d*f*j^2 + 23224320*a^5*b*c^13*d^2*e*i + 384*a*b^12*c^6*d*f*i^2 - 5985792*a^6*b*c^12*d*f*h^2 + 206010*a*b^9*c^9*d^2*f*h - 131328*a*b^9*c^9*d^2*e*i - 6300*a*b^10*c^8*d*f^2*h + 1350*a*b^11*c^7*d*f*h^2 + 16588800*a^5*b*c^13*d*e^2*h + 3456*a*b^10*c^8*d*f*g^2 + 435456*a*b^8*c^10*d^2*e*g + 13824*a*b^8*c^10*d*e^2*f + 3932160*a^11*c^8*h*i*j*k + 27525120*a^10*c^9*d*i*j*k + 82575360*a^9*c^10*d*e*j*k + 11796480*a^10*c^9*e*h*j*k + 16515072*a^9*c^10*d*h*i*k + 49545216*a^8*c^11*d*e*h*k - 2457600*a^8*c^11*e*f*i*j - 1474560*a^7*c^12*e*f*h*i - 10321920*a^6*c^13*d*e*f*i + 737077248*a^10*b^3*c^6*d*j*k^2 - 518814720*a^9*b^5*c^5*d*j*k^2 + 441354240*a^9*b^3*c^7*d*h*k^2 - 429871104*a^6*b^2*c^11*d^2*e*k - 272212992*a^8*b^5*c^6*d*h*k^2 + 305731584*a^5*b^4*c^10*d^2*e*k + 192412800*a^8*b^7*c^4*d*j*k^2 + 111912960*a^11*b^3*c^5*h*j*k^2 + 214935552*a^6*b^3*c^10*d^2*g*k + 202427136*a^7*b^6*c^6*d*f*k^2 - 49904640*a^10*b^5*c^4*h*j*k^2 - 178513920*a^8*b^4*c^7*d*f*k^2 - 152865792*a^5*b^5*c^9*d^2*g*k - 114388992*a^7*b^2*c^10*d^2*i*k + 94961664*a^10*b^2*c^7*e*i*k^2 - 9039872*a^11*b^2*c^6*i*j^2*k - 56494080*a^10*b

^4*c^5*f*j*k^2 - 2052096*a^10*b^4*c^5*i*j^2*k + 1327360*a^9*b^6*c^4*i*j^2*k
 - 158080*a^8*b^8*c^3*i*j^2*k - 47480832*a^10*b^3*c^6*g*i*k^2 + 45576960*a^
9*b^6*c^4*f*j*k^2 + 7954560*a^9*b^7*c^3*h*j*k^2 - 104693760*a^9*b^3*c^7*e*g
*k^2 + 142080*a^8*b^9*c^2*h*j*k^2 + 16017408*a^10*b^3*c^6*g*j^2*k - 4930560
*a^9*b^5*c^5*g*j^2*k - 3649536*a^9*b^2*c^8*h^2*i*k - 1843200*a^8*b^4*c^7*h^
2*i*k + 85524480*a^8*b^5*c^6*e*g*k^2 + 474240*a^8*b^7*c^4*g*j^2*k + 288000*
a^7*b^6*c^6*h^2*i*k + 63360*a^6*b^8*c^5*h^2*i*k - 8064*a^5*b^10*c^4*h^2*i*k
 - 1152*a^4*b^12*c^3*h^2*i*k - 15437824*a^11*b^2*c^6*f*j*k^2 - 32034816*a^1
0*b^2*c^7*e*j^2*k - 14369280*a^8*b^8*c^3*f*j*k^2 - 13271040*a^8*b^3*c^8*g^2
*i*k + 80267904*a^7*b^7*c^5*d*h*k^2 + 79626240*a^7*b^2*c^10*e^2*g*k + 11059
200*a^9*b^5*c^5*g*i*k^2 + 8847360*a^9*b^2*c^8*g*i^2*k - 42113280*a^7*b^9*c^
3*d*j*k^2 + 6389760*a^8*b^7*c^4*g*i*k^2 + 5898240*a^8*b^4*c^7*g*i^2*k - 376
01280*a^9*b^4*c^6*f*h*k^2 - 2949120*a^7*b^9*c^3*g*i*k^2 + 2242560*a^7*b^10*
c^2*f*j*k^2 - 2211840*a^7*b^5*c^7*g^2*i*k + 1769472*a^6*b^7*c^6*g^2*i*k + 7
49568*a^8*b^3*c^8*h*i^2*j - 442368*a^7*b^6*c^6*g*i^2*k + 442368*a^6*b^11*c^
2*g*i*k^2 - 442368*a^6*b^8*c^5*g*i^2*k + 317952*a^7*b^5*c^7*h*i^2*j - 22118
4*a^5*b^9*c^5*g^2*i*k + 73728*a^5*b^10*c^4*g*i^2*k + 38400*a^6*b^7*c^6*h*i^
2*j - 1920*a^5*b^9*c^5*h*i^2*j + 9861120*a^9*b^4*c^6*e*j^2*k - 110280960*a^
4*b^6*c^9*d^2*e*k - 93330432*a^6*b^8*c^5*d*f*k^2 + 24645888*a^8*b^6*c^5*f*h
*k^2 + 6359040*a^8*b^3*c^8*g*h^2*k - 22118400*a^9*b^4*c^6*e*i*k^2 - 3862528
*a^8*b^2*c^9*f^2*i*k - 2248704*a^7*b^4*c^8*f^2*i*k - 1290240*a^9*b^2*c^8*g*
i*j^2 - 948480*a^8*b^6*c^5*e*j^2*k - 860160*a^8*b^4*c^7*g*i*j^2 - 414720*a^
7*b^5*c^7*g*h^2*k + 303360*a^6*b^6*c^7*f^2*i*k + 266880*a^5*b^8*c^6*f^2*i*k
 - 224640*a^6*b^7*c^6*g*h^2*k - 80640*a^7*b^6*c^6*g*i*j^2 - 72960*a^4*b^10*
c^5*f^2*i*k + 17280*a^5*b^9*c^5*g*h^2*k + 12672*a^6*b^8*c^5*g*i*j^2 + 5504*
a^3*b^12*c^4*f^2*i*k + 3456*a^4*b^11*c^4*g*h^2*k - 384*a^5*b^10*c^4*g*i*j^2
 - 128*a^2*b^14*c^3*f^2*i*k + 30265344*a^6*b^4*c^9*d^2*i*k - 12779520*a^8*b
^6*c^5*e*i*k^2 - 11796480*a^8*b^3*c^8*e*i^2*k - 8847360*a^7*b^3*c^9*e^2*i*k
 - 7925760*a^10*b^2*c^7*f*h*k^2 + 7077888*a^6*b^5*c^8*e^2*i*k - 39813120*a^
7*b^3*c^9*e*g^2*k - 73175040*a^9*b^2*c^8*d*f*k^2 + 5898240*a^7*b^8*c^4*e*i*
k^2 + 5542272*a^6*b^11*c^2*d*j*k^2 - 5420160*a^7*b^8*c^4*f*h*k^2 + 55140480
*a^4*b^7*c^8*d^2*g*k + 1271808*a^7*b^3*c^9*g^2*h*j - 1040384*a^8*b^2*c^9*f*
i^2*j + 884736*a^7*b^5*c^7*e*i^2*k - 884736*a^6*b^10*c^3*e*i*k^2 + 884736*a
^6*b^7*c^6*e*i^2*k - 884736*a^5*b^7*c^7*e^2*i*k - 697344*a^7*b^4*c^8*f*i^2*
j + 414720*a^6*b^5*c^8*g^2*h*j + 226560*a^6*b^10*c^3*f*h*k^2 - 147456*a^5*b
^9*c^5*e*i^2*k - 121856*a^6*b^6*c^7*f*i^2*j + 82560*a^5*b^12*c^2*f*h*k^2 +
49152*a^5*b^12*c^2*e*i*k^2 - 17280*a^5*b^7*c^7*g^2*h*j + 8960*a^5*b^8*c^6*f
*i^2*j + 14194944*a^5*b^6*c^8*d^2*i*k - 12718080*a^8*b^2*c^9*e*h^2*k - 1061
5680*a^4*b^8*c^7*d^2*i*k - 26542080*a^6*b^4*c^9*e^2*g*k - 23592960*a^7*b^7*
c^5*e*g*k^2 - 5142528*a^8*b^3*c^8*f*h*j^2 + 5068800*a^7*b^2*c^10*f^2*h*j -
3755520*a^7*b^3*c^9*f*h^2*j + 3336192*a^7*b^3*c^9*f^2*g*k + 3000960*a^6*b^4
*c^9*f^2*h*j + 2893824*a^3*b^10*c^6*d^2*i*k + 1720320*a^8*b^3*c^8*e*i*j^2 +
 1704960*a^6*b^5*c^8*f^2*g*k - 1307520*a^5*b^7*c^7*f^2*g*k - 1085760*a^6*b^
5*c^8*f*h^2*j - 959040*a^7*b^5*c^7*f*h*j^2 + 829440*a^7*b^4*c^8*e*h^2*k - 5
52960*a^7*b^2*c^10*g*h^2*i - 552960*a^6*b^4*c^9*g*h^2*i + 449280*a^6*b^6*c^

7*e*h^2*k - 422784*a^2*b^12*c^5*d^2*i*k + 253440*a^4*b^9*c^6*f^2*g*k + 1612
80*a^7*b^5*c^7*e*i*j^2 - 145152*a^5*b^6*c^8*g*h^2*i + 103200*a^6*b^7*c^6*f*
h*j^2 + 41280*a^5*b^6*c^8*f^2*h*j - 37188*a^4*b^8*c^7*f^2*h*j - 34560*a^5*b
^8*c^6*e*h^2*k - 25344*a^6*b^7*c^6*e*i*j^2 - 17280*a^3*b^11*c^5*f^2*g*k + 1
3536*a^5*b^7*c^7*f*h^2*j - 6912*a^4*b^10*c^5*e*h^2*k + 5490*a^4*b^9*c^6*f*h
^2*j - 3456*a^4*b^8*c^7*g*h^2*i + 1980*a^3*b^10*c^6*f^2*h*j + 810*a^5*b^9*c
^5*f*h*j^2 + 768*a^5*b^9*c^5*e*i*j^2 + 384*a^2*b^13*c^4*f^2*g*k - 270*a^4*b
^11*c^4*f*h*j^2 - 180*a^3*b^11*c^5*f*h^2*j - 30*a^2*b^12*c^5*f^2*h*j + 6*a^
3*b^13*c^3*f*h*j^2 + 30067200*a^6*b^2*c^11*d^2*h*j + 13271040*a^6*b^5*c^8*e
*g^2*k - 10857600*a^6*b^9*c^4*d*h*k^2 + 2949120*a^6*b^9*c^4*e*g*k^2 + 26542
08*a^5*b^6*c^8*e^2*g*k + 2125824*a^7*b^3*c^9*d*i^2*j + 1658880*a^6*b^3*c^10
*e^2*h*j - 1419264*a^6*b^4*c^9*f*g^2*j - 1327104*a^5*b^7*c^7*e*g^2*k - 9216
00*a^7*b^2*c^10*f*g^2*j - 737280*a^7*b^2*c^10*f*h*i^2 - 568320*a^6*b^4*c^9*
f*h*i^2 + 207360*a^4*b^13*c^2*d*h*k^2 - 147456*a^5*b^11*c^3*e*g*k^2 - 13670
4*a^5*b^6*c^8*f*h*i^2 + 133632*a^6*b^5*c^8*d*i^2*j - 96768*a^5*b^7*c^7*d*i^
2*j + 80640*a^5*b^6*c^8*f*g^2*j - 69120*a^5*b^5*c^9*e^2*h*j + 13440*a^4*b^9
*c^6*d*i^2*j - 5760*a^5*b^11*c^3*d*h*k^2 - 2304*a^4*b^8*c^7*f*h*i^2 + 384*a
^3*b^10*c^6*f*h*i^2 + 11930112*a^8*b^2*c^9*d*h*j^2 - 11646720*a^3*b^9*c^7*d
^2*g*k + 8432640*a^7*b^2*c^10*d*h^2*j + 24140160*a^5*b^10*c^4*d*f*k^2 - 667
2384*a^7*b^2*c^10*e*f^2*k + 4450176*a^7*b^4*c^8*d*h*j^2 + 4337280*a^6*b^4*c
^9*d*h^2*j - 3870720*a^8*b^2*c^9*e*g*j^2 - 3409920*a^6*b^4*c^9*e*f^2*k - 28
85760*a^5*b^4*c^10*d^2*h*j - 2844288*a^4*b^6*c^9*d^2*h*j + 2615040*a^5*b^6*
c^8*e*f^2*k - 1687680*a^6*b^6*c^7*d*h*j^2 + 1482624*a^2*b^11*c^6*d^2*g*k -
1290240*a^6*b^2*c^11*f^2*g*i + 1105920*a^6*b^3*c^10*e*h^2*i + 1019412*a^3*b
^8*c^8*d^2*h*j - 1007424*a^5*b^6*c^8*d*h^2*j - 860160*a^5*b^4*c^10*f^2*g*i
- 645120*a^7*b^4*c^8*e*g*j^2 - 506880*a^4*b^8*c^7*e*f^2*k + 290304*a^5*b^5*
c^9*e*h^2*i + 197460*a^5*b^8*c^6*d*h*j^2 - 143802*a^2*b^10*c^7*d^2*h*j + 80
640*a^6*b^6*c^7*e*g*j^2 - 80640*a^4*b^6*c^9*f^2*g*i + 51948*a^4*b^8*c^7*d*h
^2*j + 34560*a^3*b^10*c^6*e*f^2*k + 12672*a^3*b^8*c^8*f^2*g*i + 10800*a^3*b
^10*c^6*d*h^2*j + 6912*a^4*b^7*c^8*e*h^2*i - 2304*a^5*b^8*c^6*e*g*j^2 - 768
*a^2*b^12*c^5*e*f^2*k - 684*a^3*b^12*c^4*d*h*j^2 - 540*a^2*b^12*c^5*d*h^2*j
 - 384*a^2*b^10*c^7*f^2*g*i - 90*a^4*b^10*c^5*d*h*j^2 + 18*a^2*b^14*c^3*d*h
*j^2 + 23385600*a^6*b^2*c^11*d*f^2*j + 23293440*a^3*b^8*c^8*d^2*e*k + 61378
56*a^6*b^3*c^10*d*g^2*j - 5677056*a^6*b^2*c^11*e^2*f*j + 5308416*a^6*b^2*c^
11*e*g^2*i - 5308416*a^5*b^3*c^11*e^2*g*i - 3786240*a^4*b^12*c^3*d*f*k^2 -
3538944*a^6*b^3*c^10*e*g*i^2 + 2654208*a^5*b^4*c^10*e*g^2*i + 1658880*a^6*b
^3*c^10*d*h*i^2 - 1354752*a^5*b^5*c^9*d*g^2*j - 1105920*a^5*b^4*c^10*f*g^2*
h - 884736*a^5*b^5*c^9*e*g*i^2 - 552960*a^6*b^2*c^11*f*g^2*h + 357120*a^3*b
^14*c^2*d*f*k^2 + 322560*a^5*b^4*c^10*e^2*f*j + 262656*a^5*b^5*c^9*d*h*i^2
+ 120960*a^4*b^7*c^8*d*g^2*j - 55296*a^4*b^7*c^8*d*h*i^2 - 34560*a^4*b^6*c^
9*f*g^2*h + 3456*a^3*b^8*c^8*f*g^2*h + 1152*a^3*b^9*c^7*d*h*i^2 + 1152*a^2*
b^11*c^6*d*h*i^2 - 13149696*a^7*b^3*c^9*d*f*j^2 - 11612160*a^5*b^2*c^12*d^2
*g*i + 10906560*a^4*b^5*c^10*d^2*f*j - 7418880*a^5*b^3*c^11*d^2*f*j + 31489
92*a^6*b^5*c^8*d*f*j^2 - 2985696*a^3*b^7*c^9*d^2*f*j - 2965248*a^2*b^10*c^7
*d^2*e*k + 1720320*a^5*b^3*c^11*e*f^2*i - 1658880*a^6*b^2*c^11*e*g*h^2 + 15

96672*a^3*b^6*c^10*d^2*g*i - 1505280*a^4*b^6*c^9*d*f^2*j - 829440*a^5*b^4*c
^10*e*g*h^2 - 508032*a^2*b^8*c^9*d^2*g*i + 378954*a^2*b^9*c^8*d^2*f*j + 362
880*a^5*b^4*c^10*d*f^2*j + 296964*a^3*b^8*c^8*d*f^2*j + 161280*a^4*b^5*c^10
*e*f^2*i - 77070*a^4*b^9*c^6*d*f*j^2 - 30240*a^5*b^7*c^7*d*f*j^2 - 25344*a^
3*b^7*c^9*e*f^2*i - 20736*a^4*b^6*c^9*e*g*h^2 - 19278*a^2*b^10*c^7*d*f^2*j
+ 8820*a^3*b^11*c^5*d*f*j^2 + 768*a^2*b^9*c^8*e*f^2*i - 378*a^2*b^13*c^4*d*
f*j^2 - 5419008*a^5*b^3*c^11*d*e^2*j - 4423680*a^5*b^2*c^12*e^2*f*h + 41472
00*a^5*b^3*c^11*d*g^2*h - 2580480*a^6*b^2*c^11*d*f*i^2 - 967680*a^5*b^4*c^1
0*d*f*i^2 + 483840*a^4*b^5*c^10*d*e^2*j - 414720*a^4*b^5*c^10*d*g^2*h - 138
240*a^4*b^4*c^11*e^2*f*h + 64512*a^4*b^6*c^9*d*f*i^2 + 39168*a^3*b^8*c^8*d*
f*i^2 - 31104*a^3*b^7*c^9*d*g^2*h + 13824*a^3*b^6*c^10*e^2*f*h + 10368*a^2*
b^9*c^8*d*g^2*h - 9216*a^2*b^10*c^7*d*f*i^2 + 15630336*a^5*b^2*c^12*d*f^2*h
 - 14459904*a^4*b^3*c^12*d^2*f*h + 9630144*a^3*b^5*c^11*d^2*f*h - 8764416*a
^5*b^3*c^11*d*f*h^2 - 3870720*a^5*b^2*c^12*e*f^2*g - 3193344*a^3*b^5*c^11*d
^2*e*i + 2867328*a^4*b^4*c^11*d*f^2*h - 2095200*a^2*b^7*c^10*d^2*f*h - 1414
080*a^3*b^6*c^10*d*f^2*h - 34836480*a^4*b^2*c^13*d^2*e*g + 1016064*a^2*b^7*
c^10*d^2*e*i - 645120*a^4*b^4*c^11*e*f^2*g + 306720*a^3*b^7*c^9*d*f*h^2 + 1
97820*a^2*b^8*c^9*d*f^2*h + 146880*a^4*b^5*c^10*d*f*h^2 + 80640*a^3*b^6*c^1
0*e*f^2*g - 55350*a^2*b^9*c^8*d*f*h^2 - 2304*a^2*b^8*c^9*e*f^2*g - 3870720*
a^5*b^2*c^12*d*f*g^2 - 1935360*a^4*b^4*c^11*d*f*g^2 - 1658880*a^4*b^3*c^12*
d*e^2*h + 725760*a^3*b^6*c^10*d*f*g^2 + 17418240*a^3*b^4*c^12*d^2*e*g - 124
416*a^3*b^5*c^11*d*e^2*h - 96768*a^2*b^8*c^9*d*f*g^2 + 41472*a^2*b^7*c^10*d
*e^2*h - 3919104*a^2*b^6*c^11*d^2*e*g - 7741440*a^4*b^2*c^13*d*e^2*f + 2903
040*a^3*b^4*c^12*d*e^2*f - 387072*a^2*b^6*c^11*d*e^2*f - 681246720*a^9*b*c^
9*d^2*k^2 + 265912320*a^11*b^3*c^5*e*k^3 + 188743680*a^12*b^2*c^5*g*k^3 - 1
32956160*a^11*b^4*c^4*g*k^3 - 52101120*a^13*b*c^5*j^2*k^2 + 25722880*a^12*b
^3*c^4*i*k^3 + 19644416*a^11*b^5*c^3*i*k^3 - 1583680*a^9*b^9*c*j^2*k^2 - 91
42272*a^10*b^7*c^2*i*k^3 - 74022912*a^10*b^5*c^4*e*k^3 - 20643840*a^11*b*c^
7*h^2*k^2 + 37011456*a^10*b^6*c^3*g*k^3 - 2293760*a^9*b^3*c^7*i^3*k - 55705
6*a^8*b^5*c^6*i^3*k + 147456*a^7*b^7*c^5*i^3*k - 65536*a^6*b^12*c*i^2*k^2 +
 32768*a^6*b^9*c^4*i^3*k - 8192*a^5*b^11*c^3*i^3*k + 430080*a^10*b*c^8*i^2*
j^2 - 2880*a^5*b^13*c*h^2*k^2 + 6635520*a^7*b^4*c^8*g^3*k - 4792320*a^9*b^8
*c^2*g*k^3 - 2211840*a^6*b^6*c^7*g^3*k + 1359360*a^10*b^2*c^7*h*j^3 + 11731
20*a^9*b^4*c^6*h*j^3 + 743040*a^7*b^4*c^8*h^3*j + 622080*a^8*b^2*c^9*h^3*j
+ 221184*a^5*b^8*c^6*g^3*k + 107136*a^6*b^6*c^7*h^3*j - 32640*a^8*b^6*c^5*h
*j^3 - 5796*a^7*b^8*c^4*h*j^3 + 540*a^5*b^8*c^6*h^3*j - 270*a^4*b^10*c^5*h^
3*j + 210*a^6*b^10*c^3*h*j^3 - 2949120*a^10*b*c^8*f^2*k^2 + 17694720*a^6*b^
3*c^10*e^3*k + 184320*a^8*b*c^10*h^2*i^2 - 3520*a^3*b^15*c*f^2*k^2 + 958464
0*a^9*b^7*c^3*e*k^3 - 2293760*a^9*b^3*c^7*f*j^3 - 2293760*a^6*b^3*c^10*f^3*
j - 1769472*a^5*b^5*c^9*e^3*k - 884736*a^6*b^3*c^10*g^3*i - 589824*a^7*b^3*
c^9*g*i^3 - 491520*a^8*b^9*c^2*e*k^3 - 442368*a^5*b^5*c^9*g^3*i - 294912*a^
6*b^5*c^8*g*i^3 - 199360*a^8*b^5*c^6*f*j^3 - 199360*a^5*b^5*c^9*f^3*j + 619
20*a^7*b^7*c^5*f*j^3 + 61920*a^4*b^7*c^8*f^3*j - 49152*a^5*b^7*c^7*g*i^3 -
3682*a^6*b^9*c^4*f*j^3 - 3682*a^3*b^9*c^7*f^3*j + 70*a^5*b^11*c^3*f*j^3 + 7
0*a^2*b^11*c^6*f^3*j + 3870720*a^8*b*c^10*e^2*j^2 + 430080*a^7*b*c^11*f^2*i

^2 - 14152320*a^4*b^4*c^11*d^3*j + 10644480*a^5*b^2*c^12*d^3*j + 5483520*a^
9*b^2*c^8*d*j^3 + 4269888*a^3*b^6*c^10*d^3*j + 3538944*a^5*b^2*c^12*e^3*i -
 1648128*a^5*b^3*c^11*f^3*h + 1330560*a^8*b^4*c^7*d*j^3 + 1179648*a^7*b^2*c
^10*e*i^3 - 898560*a^6*b^3*c^10*f*h^3 - 826560*a^7*b^6*c^6*d*j^3 - 607068*a
^2*b^8*c^9*d^3*j + 589824*a^6*b^4*c^9*e*i^3 - 354240*a^5*b^5*c^9*f*h^3 - 35
4240*a^4*b^5*c^10*f^3*h + 145188*a^6*b^8*c^5*d*j^3 + 98304*a^5*b^6*c^8*e*i^
3 + 43680*a^3*b^7*c^9*f^3*h - 21600*a^4*b^7*c^8*f*h^3 - 9576*a^5*b^10*c^4*d
*j^3 + 1350*a^3*b^9*c^7*f*h^3 - 1050*a^2*b^9*c^8*f^3*h - 504*a*b^14*c^4*d^2
*j^2 + 210*a^4*b^12*c^3*d*j^3 + 3870720*a^6*b*c^12*d^2*i^2 + 1658880*a^6*b*
c^12*e^2*h^2 - 9792*a*b^11*c^7*d^2*i^2 + 16547328*a^4*b^2*c^13*d^3*h - 1230
6816*a^3*b^4*c^12*d^3*h + 37310976*a^3*b^3*c^13*d^3*f + 3037824*a^2*b^6*c^1
1*d^3*h - 2654208*a^5*b^3*c^11*e*g^3 + 1949184*a^6*b^2*c^11*d*h^3 + 1296000
*a^5*b^4*c^10*d*h^3 - 155520*a^4*b^6*c^9*d*h^3 - 40500*a*b^10*c^8*d^2*h^2 -
 8100*a^3*b^8*c^8*d*h^3 + 4050*a^2*b^10*c^7*d*h^3 + 3870720*a^5*b*c^13*e^2*
f^2 + 34836480*a^4*b*c^14*d^2*e^2 - 108864*a*b^9*c^9*d^2*g^2 - 8068032*a^2*
b^5*c^12*d^3*f - 5623296*a^4*b^3*c^12*d*f^3 + 1737792*a^3*b^5*c^11*d*f^3 -
260190*a*b^8*c^10*d^2*f^2 - 211680*a^2*b^7*c^10*d*f^3 - 435456*a*b^7*c^11*d
^2*e^2 - 377487360*a^12*b*c^6*e*k^3 + 1434977280*a^8*b^3*c^8*d^2*k^2 + 1734
08256*a^7*c^12*d^2*e*k + 3276800*a^12*c^7*i*j^2*k - 125829120*a^13*b*c^5*i*
k^3 + 26214400*a^12*c^7*f*j*k^2 + 1179648*a^10*c^9*h^2*i*k + 13440*a^6*b^13
*h*j*k^2 + 50331648*a^11*c^8*e*i*k^2 + 110100480*a^10*c^9*d*f*k^2 + 5780275
2*a^8*c^11*d^2*i*k + 9830400*a^11*c^8*e*j^2*k - 3276800*a^9*c^10*f^2*i*k +
4480*a^5*b^14*f*j*k^2 + 15728640*a^11*c^8*f*h*k^2 - 409600*a^9*c^10*f*i^2*j
 - 1152*b^16*c^3*d^2*i*k - 1220516352*a^7*b^5*c^7*d^2*k^2 + 3538944*a^9*c^1
0*e*h^2*k + 384000*a^8*c^11*f^2*h*j + 13440*a^4*b^15*d*j*k^2 + 384*a^3*b^16
*f*h*k^2 + 20321280*a^7*c^12*d^2*h*j - 245760*a^8*c^11*f*h*i^2 + 3456*b^15*
c^4*d^2*g*k - 270*b^14*c^5*d^2*h*j - 9830400*a^8*c^11*e*f^2*k + 4838400*a^9
*c^10*d*h*j^2 + 2903040*a^8*c^11*d*h^2*j - 1966080*a^10*b*c^8*i^3*k + 14336
00*a^9*b^9*c*i*k^3 + 1152*a^2*b^17*d*h*k^2 - 3686400*a^7*c^12*e^2*f*j - 530
84160*a^7*b*c^11*e^3*k - 6912*b^14*c^5*d^2*e*k - 3456*b^12*c^7*d^2*g*i + 63
0*b^13*c^6*d^2*f*j + 2688000*a^7*c^12*d*f^2*j + 245760*a^8*b^10*c*g*k^3 - 2
211840*a^6*c^13*e^2*f*h - 1720320*a^7*c^12*d*f*i^2 - 9450*b^11*c^8*d^2*f*h
+ 6912*b^11*c^8*d^2*e*i + 1612800*a^6*c^13*d*f^2*h - 1344000*a^10*b*c^8*f*j
^3 - 1344000*a^7*b*c^11*f^3*j - 393216*a^8*b*c^10*g*i^3 - 23616*a*b^17*c*d^
2*k^2 - 20736*b^10*c^9*d^2*e*g - 75188736*a^4*b*c^14*d^3*f - 883200*a^6*b*c
^12*f^3*h - 317952*a^7*b*c^11*f*h^3 + 43416*a*b^10*c^8*d^3*j - 15482880*a^5
*c^14*d*e^2*f - 10616832*a^5*b*c^13*e^3*g - 345060*a*b^8*c^10*d^3*h - 42624
00*a^5*b*c^13*d*f^3 + 852768*a*b^7*c^11*d^3*f + 7350*a*b^9*c^9*d*f^3 + 5845
78368*a^6*b^7*c^6*d^2*k^2 + 93905920*a^12*b^3*c^4*j^2*k^2 - 177997248*a^5*b
^9*c^5*d^2*k^2 - 50967040*a^11*b^5*c^3*j^2*k^2 + 104693760*a^9*b^2*c^8*e^2*
k^2 + 12849984*a^10*b^7*c^2*j^2*k^2 + 20021248*a^11*b^2*c^6*i^2*k^2 - 85524
480*a^8*b^4*c^7*e^2*k^2 + 33223680*a^10*b^3*c^6*h^2*k^2 + 4227072*a^10*b^4*
c^5*i^2*k^2 - 3973120*a^9*b^6*c^4*i^2*k^2 + 344064*a^7*b^10*c^2*i^2*k^2 - 8
1920*a^8*b^8*c^3*i^2*k^2 - 11386368*a^9*b^5*c^5*h^2*k^2 + 26173440*a^9*b^4*
c^6*g^2*k^2 - 21381120*a^8*b^6*c^5*g^2*k^2 + 18874368*a^10*b^2*c^7*g^2*k^2

+ 501760*a^9*b^3*c^7*i^2*j^2 + 452160*a^8*b^7*c^4*h^2*k^2 + 385920*a^7*b^9*
c^3*h^2*k^2 + 170240*a^8*b^5*c^6*i^2*j^2 - 48960*a^6*b^11*c^2*h^2*k^2 + 921
6*a^7*b^7*c^5*i^2*j^2 - 1984*a^6*b^9*c^4*i^2*j^2 + 64*a^5*b^11*c^3*i^2*j^2
+ 5898240*a^7*b^8*c^4*g^2*k^2 + 1419840*a^8*b^4*c^7*h^2*j^2 + 1387008*a^9*b
^2*c^8*h^2*j^2 - 737280*a^6*b^10*c^3*g^2*k^2 + 84960*a^7*b^6*c^6*h^2*j^2 +
36864*a^5*b^12*c^2*g^2*k^2 - 8010*a^6*b^8*c^5*h^2*j^2 - 180*a^5*b^10*c^4*h^
2*j^2 + 9*a^4*b^12*c^3*h^2*j^2 + 14115840*a^9*b^3*c^7*f^2*k^2 - 9231552*a^7
*b^7*c^5*f^2*k^2 + 23592960*a^7*b^6*c^6*e^2*k^2 + 4984320*a^8*b^5*c^6*f^2*k
^2 + 3759040*a^6*b^9*c^4*f^2*k^2 + 36190080*a^4*b^11*c^4*d^2*k^2 + 967680*a
^8*b^3*c^8*g^2*j^2 - 727360*a^5*b^11*c^3*f^2*k^2 + 276480*a^7*b^3*c^9*h^2*i
^2 + 161280*a^7*b^5*c^7*g^2*j^2 + 140544*a^6*b^5*c^8*h^2*i^2 + 72960*a^4*b^
13*c^2*f^2*k^2 + 25344*a^5*b^7*c^7*h^2*i^2 - 20160*a^6*b^7*c^6*g^2*j^2 + 57
6*a^5*b^9*c^5*g^2*j^2 + 576*a^4*b^9*c^6*h^2*i^2 + 3808000*a^8*b^2*c^9*f^2*j
^2 - 2949120*a^6*b^8*c^5*e^2*k^2 + 1643712*a^7*b^4*c^8*f^2*j^2 + 884736*a^7
*b^2*c^10*g^2*i^2 + 884736*a^6*b^4*c^9*g^2*i^2 + 221184*a^5*b^6*c^8*g^2*i^2
 + 147456*a^5*b^10*c^4*e^2*k^2 - 125440*a^6*b^6*c^7*f^2*j^2 - 13790*a^5*b^8
*c^6*f^2*j^2 + 1785*a^4*b^10*c^5*f^2*j^2 - 70*a^3*b^12*c^4*f^2*j^2 - 495360
0*a^3*b^13*c^3*d^2*k^2 + 18427392*a^7*b^2*c^10*d^2*j^2 + 645120*a^7*b^3*c^9
*e^2*j^2 + 501760*a^6*b^3*c^10*f^2*i^2 + 442944*a^2*b^15*c^2*d^2*k^2 + 4147
20*a^6*b^3*c^10*g^2*h^2 + 207360*a^5*b^5*c^9*g^2*h^2 + 170240*a^5*b^5*c^9*f
^2*i^2 - 80640*a^6*b^5*c^8*e^2*j^2 + 9216*a^4*b^7*c^8*f^2*i^2 + 5184*a^4*b^
7*c^8*g^2*h^2 + 2304*a^5*b^7*c^7*e^2*j^2 - 1984*a^3*b^9*c^7*f^2*i^2 + 64*a^
2*b^11*c^6*f^2*i^2 - 4148928*a^6*b^4*c^9*d^2*j^2 + 3538944*a^6*b^2*c^11*e^2
*i^2 + 1684224*a^6*b^2*c^11*f^2*h^2 + 1264320*a^5*b^4*c^10*f^2*h^2 - 118339
2*a^5*b^6*c^8*d^2*j^2 + 884736*a^5*b^4*c^10*e^2*i^2 + 645750*a^4*b^8*c^7*d^
2*j^2 + 126720*a^4*b^6*c^9*f^2*h^2 - 115920*a^3*b^10*c^6*d^2*j^2 - 13950*a^
3*b^8*c^8*f^2*h^2 + 10836*a^2*b^12*c^5*d^2*j^2 + 225*a^2*b^10*c^7*f^2*h^2 +
 1935360*a^5*b^3*c^11*d^2*i^2 + 967680*a^5*b^3*c^11*f^2*g^2 + 829440*a^5*b^
3*c^11*e^2*h^2 - 532224*a^4*b^5*c^10*d^2*i^2 + 161280*a^4*b^5*c^10*f^2*g^2
- 96768*a^3*b^7*c^9*d^2*i^2 + 62784*a^2*b^9*c^8*d^2*i^2 + 20736*a^4*b^5*c^1
0*e^2*h^2 - 20160*a^3*b^7*c^9*f^2*g^2 + 576*a^2*b^9*c^8*f^2*g^2 + 11487744*
a^5*b^2*c^12*d^2*h^2 + 7962624*a^5*b^2*c^12*e^2*g^2 + 35525376*a^4*b^2*c^13
*d^2*f^2 - 1412640*a^3*b^6*c^10*d^2*h^2 + 461376*a^4*b^4*c^11*d^2*h^2 + 375
030*a^2*b^8*c^9*d^2*h^2 + 8709120*a^4*b^3*c^12*d^2*g^2 - 4354560*a^3*b^5*c^
11*d^2*g^2 + 979776*a^2*b^7*c^10*d^2*g^2 + 645120*a^4*b^3*c^12*e^2*f^2 - 80
640*a^3*b^5*c^11*e^2*f^2 + 2304*a^2*b^7*c^10*e^2*f^2 - 15269184*a^3*b^4*c^1
2*d^2*f^2 + 2870784*a^2*b^6*c^11*d^2*f^2 - 17418240*a^3*b^3*c^13*d^2*e^2 +
3919104*a^2*b^5*c^12*d^2*e^2 + 384*a*b^18*d*f*k^2 - 199229440*a^14*b^2*c^3*
k^4 + 8388608*a^12*c^7*i^2*k^2 + 75497472*a^10*c^9*e^2*k^2 + 78400*a^8*b^11
*j^2*k^2 + 4096*a^5*b^14*i^2*k^2 + 345600*a^10*c^9*h^2*j^2 + 576*a^4*b^15*h
^2*k^2 + 57937920*a^13*b^4*c^2*k^4 + 320000*a^9*c^10*f^2*j^2 + 64*a^2*b^17*
f^2*k^2 + 16934400*a^8*c^11*d^2*j^2 + 9*b^16*c^3*d^2*j^2 + 3538944*a^7*c^12
*e^2*i^2 + 115200*a^7*c^12*f^2*h^2 + 576*b^13*c^6*d^2*i^2 + 2025*b^12*c^7*d
^2*h^2 + 6096384*a^6*c^13*d^2*h^2 + 492800*a^11*b^2*c^6*j^4 + 351456*a^10*b
^4*c^5*j^4 - 43120*a^9*b^6*c^4*j^4 + 5184*b^11*c^8*d^2*g^2 + 1225*a^8*b^8*c

^3*j^4 + 131072*a^8*b^2*c^9*i^4 + 98304*a^7*b^4*c^8*i^4 + 32768*a^6*b^6*c^7
*i^4 + 11025*b^10*c^9*d^2*f^2 + 4096*a^5*b^8*c^6*i^4 + 5644800*a^5*c^14*d^2
*f^2 + 142560*a^6*b^4*c^9*h^4 + 103680*a^7*b^2*c^10*h^4 + 32400*a^5*b^6*c^8
*h^4 + 20736*b^9*c^10*d^2*e^2 + 2025*a^4*b^8*c^7*h^4 + 331776*a^5*b^4*c^10*
g^4 + 492800*a^5*b^2*c^12*f^4 + 351456*a^4*b^4*c^11*f^4 - 43120*a^3*b^6*c^1
0*f^4 + 1225*a^2*b^8*c^9*f^4 - 27433728*a^3*b^2*c^14*d^4 + 6446304*a^2*b^4*
c^13*d^4 + a^2*b^14*c^3*f^2*j^2 - 81920*a^8*b^11*i*k^3 + 384000*a^11*c^8*h*
j^3 + 138240*a^9*c^10*h^3*j + 47416320*a^6*c^13*d^3*j - 1134*b^12*c^7*d^3*j
 + 7077888*a^6*c^13*e^3*i + 2688000*a^10*c^9*d*j^3 + 786432*a^8*c^11*e*i^3
 + 28449792*a^5*c^14*d^3*h - 7782400*a^12*b^6*c*k^4 + 17010*b^10*c^9*d^3*h +
 580608*a^7*c^12*d*h^3 - 39690*b^9*c^10*d^3*f - 734832*a*b^6*c^12*d^4 + 268
435456*a^15*c^4*k^4 + 576*b^19*d^2*k^2 + 409600*a^11*b^8*k^4 + 160000*a^12*
c^7*j^4 + 65536*a^9*c^10*i^4 + 20736*a^8*c^11*h^4 + 49787136*a^4*c^15*d^4 +
 160000*a^6*c^13*f^4 + 5308416*a^5*c^14*e^4 + 35721*b^8*c^11*d^4, z, n)*((1
1010048*a^9*c^10*d*k - 327680*a^8*c^11*f*i - 983040*a^7*c^12*e*f + 1572864*
a^10*c^9*h*k + 2621440*a^11*c^8*j*k + 3244032*a^6*b*c^12*d*e + 1081344*a^7*
b*c^11*d*i + 884736*a^7*b*c^11*e*h + 491520*a^7*b*c^11*f*g + 1277952*a^8*b*
c^10*e*j + 294912*a^8*b*c^10*h*i + 360448*a^9*b*c^9*f*k + 425984*a^9*b*c^9*
i*j + 4608*a^2*b^9*c^8*d*e - 87552*a^3*b^7*c^9*d*e + 681984*a^4*b^5*c^10*d*
e - 2433024*a^5*b^3*c^11*d*e - 2304*a^2*b^10*c^7*d*g + 43776*a^3*b^8*c^8*d*
g + 1536*a^3*b^8*c^8*e*f - 340992*a^4*b^6*c^9*d*g - 39936*a^4*b^6*c^9*e*f +
 1216512*a^5*b^4*c^10*d*g + 184320*a^5*b^4*c^10*e*f - 1622016*a^6*b^2*c^11*
d*g + 49152*a^6*b^2*c^11*e*f + 768*a^2*b^11*c^6*d*i - 13056*a^3*b^9*c^7*d*i
 - 768*a^3*b^9*c^7*f*g + 84480*a^4*b^7*c^8*d*i + 4608*a^4*b^7*c^8*e*h + 199
68*a^4*b^7*c^8*f*g - 178176*a^5*b^5*c^9*d*i + 18432*a^5*b^5*c^9*e*h - 92160
*a^5*b^5*c^9*f*g - 270336*a^6*b^3*c^10*d*i - 368640*a^6*b^3*c^10*e*h - 2457
6*a^6*b^3*c^10*f*g - 768*a^2*b^14*c^3*d*k + 256*a^3*b^10*c^6*f*i + 22272*a^
3*b^12*c^4*d*k - 6144*a^4*b^8*c^7*f*i - 2304*a^4*b^8*c^7*g*h - 282624*a^4*b
^10*c^5*d*k + 17408*a^5*b^6*c^8*f*i - 9216*a^5*b^6*c^8*g*h - 1536*a^5*b^7*c
^7*e*j + 2003712*a^5*b^8*c^6*d*k + 69632*a^6*b^4*c^9*f*i + 184320*a^6*b^4*c
^9*g*h + 92160*a^6*b^5*c^8*e*j - 8426496*a^6*b^6*c^7*d*k - 147456*a^7*b^2*c
^10*f*i - 442368*a^7*b^2*c^10*g*h - 663552*a^7*b^3*c^9*e*j + 20484096*a^7*b
^4*c^8*d*k - 25411584*a^8*b^2*c^9*d*k - 256*a^3*b^13*c^3*f*k + 768*a^4*b^9*
c^6*h*i + 9216*a^4*b^11*c^4*f*k + 4608*a^5*b^7*c^7*h*i + 768*a^5*b^8*c^6*g*
j - 113920*a^5*b^9*c^5*f*k - 55296*a^6*b^5*c^8*h*i - 46080*a^6*b^6*c^7*g*j
 + 658944*a^6*b^7*c^6*f*k + 24576*a^7*b^3*c^9*h*i + 331776*a^7*b^4*c^8*g*j -
 1812480*a^7*b^5*c^7*f*k - 638976*a^8*b^2*c^9*g*j + 1810432*a^8*b^3*c^8*f*k
 - 768*a^4*b^12*c^3*h*k - 256*a^5*b^9*c^5*i*j + 8448*a^5*b^10*c^4*h*k + 148
48*a^6*b^7*c^6*i*j + 3840*a^6*b^8*c^5*h*k - 79872*a^7*b^5*c^7*i*j - 427008*
a^7*b^6*c^6*h*k - 8192*a^8*b^3*c^8*i*j + 2150400*a^8*b^4*c^7*h*k - 3784704*
a^9*b^2*c^8*h*k - 8960*a^6*b^10*c^3*j*k + 166656*a^7*b^8*c^4*j*k - 1217536*
a^8*b^6*c^5*j*k + 4198400*a^9*b^4*c^6*j*k - 6340608*a^10*b^2*c^7*j*k)/(512*
(4096*a^10*c^10 + a^4*b^12*c^4 - 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a
^7*b^6*c^7 + 3840*a^8*b^4*c^8 - 6144*a^9*b^2*c^9)) + root(56371445760*a^11*
b^8*c^12*z^4 - 503316480*a^8*b^14*c^9*z^4 + 47185920*a^7*b^16*c^8*z^4 - 262

1440*a^6*b^18*c^7*z^4 + 65536*a^5*b^20*c^6*z^4 - 171798691840*a^14*b^2*c^15
*z^4 + 193273528320*a^13*b^4*c^14*z^4 - 128849018880*a^12*b^6*c^13*z^4 - 16
911433728*a^10*b^10*c^11*z^4 + 3523215360*a^9*b^12*c^10*z^4 + 68719476736*a
^15*c^16*z^4 - 47185920*a^7*b^16*c^5*k*z^3 + 2621440*a^6*b^18*c^4*k*z^3 - 6
5536*a^5*b^20*c^3*k*z^3 + 171798691840*a^14*b^2*c^12*k*z^3 - 193273528320*a
^13*b^4*c^11*k*z^3 + 128849018880*a^12*b^6*c^10*k*z^3 + 16911433728*a^10*b^
10*c^8*k*z^3 - 3523215360*a^9*b^12*c^7*k*z^3 - 56371445760*a^11*b^8*c^9*k*z
^3 + 503316480*a^8*b^14*c^6*k*z^3 - 68719476736*a^15*c^13*k*z^3 + 1536*a*b^
18*c^6*d*f*z^2 - 2571632640*a^9*b^5*c^11*d*j*z^2 + 2548039680*a^9*b^3*c^13*
d*h*z^2 + 2453667840*a^9*b^7*c^9*e*k*z^2 + 2181038080*a^12*b^3*c^10*i*k*z^2
 - 6492782592*a^10*b^5*c^10*e*k*z^2 + 1509949440*a^9*b^3*c^13*e*g*z^2 - 140
1421824*a^8*b^5*c^12*d*h*z^2 - 1226833920*a^9*b^8*c^8*g*k*z^2 - 1321205760*
a^9*b^2*c^14*d*f*z^2 - 2793406464*a^11*b*c^13*d*j*z^2 + 9563013120*a^11*b^3
*c^11*e*k*z^2 + 890634240*a^8*b^7*c^10*d*j*z^2 - 754974720*a^8*b^5*c^12*e*g
*z^2 - 570425344*a^11*b^5*c^9*i*k*z^2 + 732168192*a^7*b^6*c^12*d*f*z^2 - 58
1959680*a^10*b^4*c^11*f*j*z^2 - 603979776*a^10*b^2*c^13*e*i*z^2 + 534773760
*a^11*b^3*c^11*h*j*z^2 - 558366720*a^8*b^9*c^8*e*k*z^2 - 4781506560*a^11*b^
4*c^10*g*k*z^2 - 2013265920*a^13*b*c^11*i*k*z^2 - 456130560*a^9*b^4*c^12*f*
h*z^2 + 384040960*a^9*b^6*c^10*f*j*z^2 - 264241152*a^10*b^7*c^8*i*k*z^2 + 3
90463488*a^7*b^7*c^11*d*h*z^2 + 279183360*a^8*b^10*c^7*g*k*z^2 + 301989888*
a^10*b^3*c^12*g*i*z^2 + 222822400*a^9*b^9*c^7*i*k*z^2 - 366280704*a^6*b^8*c
^11*d*f*z^2 - 330301440*a^8*b^4*c^13*d*f*z^2 + 254017536*a^8*b^6*c^11*f*h*z
^2 - 1887436800*a^10*b*c^14*d*h*z^2 + 188743680*a^10*b^2*c^13*f*h*z^2 - 185
303040*a^7*b^9*c^9*d*j*z^2 - 117964800*a^10*b^5*c^10*h*j*z^2 - 6039797760*a
^12*b*c^12*e*k*z^2 - 67502080*a^8*b^11*c^6*i*k*z^2 + 121634816*a^11*b^2*c^1
2*f*j*z^2 + 188743680*a^7*b^7*c^11*e*g*z^2 - 115671040*a^8*b^8*c^9*f*j*z^2
+ 125829120*a^8*b^6*c^11*e*i*z^2 + 10813440*a^7*b^13*c^5*i*k*z^2 + 76677120
*a^7*b^11*c^7*e*k*z^2 - 38338560*a^7*b^12*c^6*g*k*z^2 - 37355520*a^9*b^7*c^
9*h*j*z^2 - 917504*a^6*b^15*c^4*i*k*z^2 + 32768*a^5*b^17*c^3*i*k*z^2 - 6291
4560*a^8*b^7*c^10*g*i*z^2 + 23101440*a^8*b^9*c^8*h*j*z^2 - 4349952*a^7*b^11
*c^7*h*j*z^2 + 2949120*a^6*b^14*c^5*g*k*z^2 + 337920*a^6*b^13*c^6*h*j*z^2 -
 98304*a^5*b^16*c^4*g*k*z^2 - 7680*a^5*b^15*c^5*h*j*z^2 - 61931520*a^7*b^8*
c^10*f*h*z^2 + 23592960*a^7*b^9*c^9*g*i*z^2 + 17940480*a^7*b^10*c^8*f*j*z^2
 - 47185920*a^7*b^8*c^10*e*i*z^2 - 5898240*a^6*b^13*c^6*e*k*z^2 - 3538944*a
^6*b^11*c^8*g*i*z^2 - 1347584*a^6*b^12*c^7*f*j*z^2 + 196608*a^5*b^15*c^5*e*
k*z^2 + 196608*a^5*b^13*c^7*g*i*z^2 + 35840*a^5*b^14*c^6*f*j*z^2 + 96583680
*a^5*b^10*c^10*d*f*z^2 + 23371776*a^6*b^11*c^8*d*j*z^2 - 51609600*a^6*b^9*c
^10*d*h*z^2 + 7077888*a^6*b^10*c^9*e*i*z^2 + 6144000*a^6*b^10*c^9*f*h*z^2 -
 1677312*a^5*b^13*c^7*d*j*z^2 - 393216*a^5*b^12*c^8*e*i*z^2 + 61440*a^5*b^1
2*c^8*f*h*z^2 + 53760*a^4*b^15*c^6*d*j*z^2 - 46080*a^4*b^14*c^7*f*h*z^2 + 1
536*a^3*b^16*c^6*f*h*z^2 - 23592960*a^6*b^9*c^10*e*g*z^2 + 1179648*a^5*b^11
*c^9*e*g*z^2 + 829440*a^4*b^13*c^8*d*h*z^2 + 368640*a^5*b^11*c^9*d*h*z^2 -
105984*a^3*b^15*c^7*d*h*z^2 + 4608*a^2*b^17*c^6*d*h*z^2 - 15175680*a^4*b^12
*c^9*d*f*z^2 + 1428480*a^3*b^14*c^8*d*f*z^2 - 73728*a^2*b^16*c^7*d*f*z^2 +
4108320768*a^10*b^3*c^12*d*j*z^2 - 1207959552*a^10*b*c^14*e*g*z^2 - 5788139

52*a^12*b*c^12*h*j*z^2 + 3246391296*a^10*b^6*c^9*g*k*z^2 - 402653184*a^11*b *c^13*g*i*z^2 + 3019898880*a^12*b^2*c^11*g*k*z^2 - 440401920*a^10*b*c^14*f^ 2*z^2 - 188743680*a^11*b*c^13*h^2*z^2 + 1761607680*a^10*c^15*d*f*z^2 - 6553 60*a^6*b^18*c*k^2*z^2 - 94464*a*b^17*c^7*d^2*z^2 + 6936330240*a^8*b^3*c^14* d^2*z^2 + 2464874496*a^6*b^7*c^12*d^2*z^2 - 3963617280*a^9*b*c^15*d^2*z^2 + 58007224320*a^13*b^4*c^8*k^2*z^2 + 14968422400*a^11*b^8*c^6*k^2*z^2 + 8053 06368*a^11*c^14*e*i*z^2 - 35966156800*a^12*b^6*c^7*k^2*z^2 + 419430400*a^12 *c^13*f*j*z^2 - 1509949440*a^9*b^2*c^14*e^2*z^2 + 251658240*a^11*c^14*f*h*z ^2 - 56874762240*a^14*b^2*c^9*k^2*z^2 - 5400428544*a^7*b^5*c^13*d^2*z^2 + 8 90470400*a^9*b^12*c^4*k^2*z^2 + 754974720*a^8*b^4*c^13*e^2*z^2 - 730054656* a^5*b^9*c^11*d^2*z^2 + 477102080*a^12*b^3*c^10*j^2*z^2 + 477102080*a^9*b^3* c^13*f^2*z^2 - 377487360*a^9*b^4*c^12*g^2*z^2 + 301989888*a^10*b^2*c^13*g^2 *z^2 - 174325760*a^11*b^5*c^9*j^2*z^2 - 126156800*a^8*b^14*c^3*k^2*z^2 + 18 8743680*a^8*b^6*c^11*g^2*z^2 + 141557760*a^10*b^3*c^12*h^2*z^2 - 174325760* a^8*b^5*c^12*f^2*z^2 - 188743680*a^7*b^6*c^12*e^2*z^2 - 4350935040*a^10*b^1 0*c^5*k^2*z^2 + 146165760*a^4*b^11*c^10*d^2*z^2 - 50331648*a^10*b^4*c^11*i^ 2*z^2 + 11796480*a^7*b^16*c^2*k^2*z^2 - 33554432*a^11*b^2*c^12*i^2*z^2 + 11 206656*a^10*b^7*c^8*j^2*z^2 + 8929280*a^9*b^9*c^7*j^2*z^2 + 20971520*a^9*b^ 6*c^10*i^2*z^2 - 2600960*a^8*b^11*c^6*j^2*z^2 + 291840*a^7*b^13*c^5*j^2*z^2 - 14080*a^6*b^15*c^4*j^2*z^2 + 256*a^5*b^17*c^3*j^2*z^2 - 47185920*a^7*b^8 *c^10*g^2*z^2 - 26542080*a^8*b^7*c^10*h^2*z^2 - 2752512*a^7*b^10*c^8*i^2*z^ 2 + 2621440*a^8*b^8*c^9*i^2*z^2 + 524288*a^6*b^12*c^7*i^2*z^2 - 32768*a^5*b ^14*c^6*i^2*z^2 + 9584640*a^7*b^9*c^9*h^2*z^2 - 2359296*a^9*b^5*c^11*h^2*z^ 2 - 1290240*a^6*b^11*c^8*h^2*z^2 + 46080*a^5*b^13*c^7*h^2*z^2 + 2304*a^4*b^ 15*c^6*h^2*z^2 + 5898240*a^6*b^10*c^9*g^2*z^2 - 294912*a^5*b^12*c^8*g^2*z^2 + 11206656*a^7*b^7*c^11*f^2*z^2 + 8929280*a^6*b^9*c^10*f^2*z^2 + 23592960* a^6*b^8*c^11*e^2*z^2 - 2600960*a^5*b^11*c^9*f^2*z^2 + 291840*a^4*b^13*c^8*f ^2*z^2 - 14080*a^3*b^15*c^7*f^2*z^2 + 256*a^2*b^17*c^6*f^2*z^2 - 19860480*a ^3*b^13*c^9*d^2*z^2 - 1179648*a^5*b^10*c^10*e^2*z^2 + 1771776*a^2*b^15*c^8* d^2*z^2 - 440401920*a^13*b*c^11*j^2*z^2 + 1207959552*a^10*c^15*e^2*z^2 + 13 4217728*a^12*c^13*i^2*z^2 + 25769803776*a^15*c^10*k^2*z^2 + 16384*a^5*b^20* k^2*z^2 + 2304*b^19*c^6*d^2*z^2 + 165150720*a^9*b*c^12*d*g*j*z + 23592960*a ^10*b*c^11*g*h*j*z + 169869312*a^7*b*c^14*d*e*f*z + 99090432*a^8*b*c^13*d*g *h*z - 3145728*a^9*b*c^12*f*h*i*z + 56623104*a^8*b*c^13*d*f*i*z - 1536*a*b^ 18*c^3*d*f*k*z - 9437184*a^8*b*c^13*e*f*h*z + 1536*a*b^15*c^6*d*f*i*z - 460 8*a*b^14*c^7*d*f*g*z + 9216*a*b^13*c^8*d*e*f*z + 2173501440*a^9*b^5*c^8*d*j *k*z - 1987706880*a^9*b^3*c^10*d*h*k*z + 1121255424*a^8*b^5*c^9*d*h*k*z + 8 61143040*a^8*b^4*c^10*d*f*k*z - 859963392*a^7*b^6*c^9*d*f*k*z - 780779520*a ^8*b^7*c^7*d*j*k*z - 754974720*a^9*b^3*c^10*e*g*k*z + 2222456832*a^11*b*c^1 0*d*j*k*z - 454164480*a^11*b^3*c^8*h*j*k*z + 377487360*a^8*b^5*c^9*e*g*k*z + 290979840*a^10*b^4*c^8*f*j*k*z + 381026304*a^6*b^8*c^8*d*f*k*z + 41287680 0*a^8*b^2*c^12*d*e*j*z + 301989888*a^10*b^2*c^10*e*i*k*z - 320421888*a^7*b^ 7*c^8*d*h*k*z + 185794560*a^10*b^5*c^7*h*j*k*z - 192020480*a^9*b^6*c^7*f*j* k*z + 190709760*a^9*b^4*c^9*f*h*k*z - 150994944*a^10*b^3*c^9*g*i*k*z + 1689 90720*a^7*b^9*c^6*d*j*k*z + 235929600*a^9*b^2*c^11*d*f*k*z - 206438400*a^8*

b^3*c^11*d*g*j*z - 206438400*a^7*b^4*c^11*d*e*j*z - 101646336*a^8*b^6*c^8*f
*h*k*z - 29245440*a^9*b^7*c^6*h*j*k*z - 60817408*a^11*b^2*c^9*f*j*k*z + 578
35520*a^8*b^8*c^6*f*j*k*z + 219414528*a^7*b^2*c^13*d*e*h*z - 70778880*a^10*
b^2*c^10*f*h*k*z + 677376*a^7*b^11*c^4*h*j*k*z - 645120*a^8*b^9*c^5*h*j*k*z
 - 53760*a^6*b^13*c^3*h*j*k*z + 31457280*a^8*b^7*c^7*g*i*k*z - 62914560*a^8
*b^6*c^8*e*i*k*z - 94371840*a^7*b^7*c^8*e*g*k*z - 221773824*a^6*b^3*c^13*d*
e*f*z + 82575360*a^9*b^2*c^11*d*i*j*z + 11796480*a^10*b^2*c^10*h*i*j*z - 11
796480*a^7*b^9*c^6*g*i*k*z - 8970240*a^7*b^10*c^5*f*j*k*z + 103219200*a^7*b
^5*c^10*d*g*j*z - 2457600*a^8*b^6*c^8*h*i*j*z + 1769472*a^6*b^11*c^5*g*i*k*
z + 921600*a^7*b^8*c^7*h*i*j*z + 673792*a^6*b^12*c^4*f*j*k*z - 138240*a^6*b
^10*c^6*h*i*j*z - 98304*a^5*b^13*c^4*g*i*k*z - 17920*a^5*b^14*c^3*f*j*k*z +
 7680*a^5*b^12*c^5*h*i*j*z - 97136640*a^5*b^10*c^7*d*f*k*z - 29491200*a^9*b
^3*c^10*g*h*j*z + 58982400*a^9*b^2*c^11*e*h*j*z + 23592960*a^7*b^8*c^7*e*i*
k*z - 22169088*a^6*b^11*c^5*d*j*k*z + 21381120*a^7*b^8*c^7*f*h*k*z + 147456
00*a^8*b^5*c^9*g*h*j*z + 42854400*a^6*b^9*c^7*d*h*k*z - 109707264*a^7*b^3*c
^12*d*g*h*z - 3686400*a^7*b^7*c^8*g*h*j*z - 3538944*a^6*b^10*c^6*e*i*k*z +
1645056*a^5*b^13*c^4*d*j*k*z - 890880*a^6*b^10*c^6*f*h*k*z + 460800*a^6*b^9
*c^7*g*h*j*z - 330240*a^5*b^12*c^5*f*h*k*z + 196608*a^5*b^12*c^5*e*i*k*z -
53760*a^4*b^15*c^3*d*j*k*z + 46080*a^4*b^14*c^4*f*h*k*z - 23040*a^5*b^11*c^
6*g*h*j*z - 1536*a^3*b^16*c^3*f*h*k*z - 29491200*a^8*b^4*c^10*e*h*j*z - 172
03200*a^7*b^6*c^9*d*i*j*z + 11796480*a^6*b^9*c^7*e*g*k*z + 110886912*a^6*b^
4*c^12*d*f*g*z + 7372800*a^7*b^6*c^9*e*h*j*z + 40108032*a^8*b^2*c^12*d*h*i*
z + 6451200*a^6*b^8*c^8*d*i*j*z + 2359296*a^8*b^3*c^11*f*h*i*z - 967680*a^5
*b^10*c^7*d*i*j*z - 921600*a^6*b^8*c^8*e*h*j*z - 829440*a^4*b^13*c^5*d*h*k*
z - 589824*a^5*b^11*c^6*e*g*k*z - 491520*a^6*b^7*c^9*f*h*i*z + 184320*a^5*b
^9*c^8*f*h*i*z + 105984*a^3*b^15*c^4*d*h*k*z + 69120*a^5*b^11*c^6*d*h*k*z +
 53760*a^4*b^12*c^6*d*i*j*z + 46080*a^5*b^10*c^7*e*h*j*z - 27648*a^4*b^11*c
^7*f*h*i*z - 4608*a^2*b^17*c^3*d*h*k*z + 1536*a^3*b^13*c^6*f*h*i*z - 258048
00*a^6*b^7*c^9*d*g*j*z - 88473600*a^6*b^4*c^12*d*e*h*z + 51609600*a^6*b^6*c
^10*d*e*j*z - 84934656*a^7*b^2*c^13*d*f*g*z + 117964800*a^5*b^5*c^12*d*e*f*
z + 15160320*a^4*b^12*c^6*d*f*k*z - 45613056*a^7*b^3*c^12*d*f*i*z + 4423680
0*a^6*b^5*c^11*d*g*h*z - 10321920*a^6*b^6*c^10*d*h*i*z + 7077888*a^7*b^4*c^
11*d*h*i*z - 5898240*a^7*b^4*c^11*f*g*h*z + 4718592*a^8*b^2*c^12*f*g*h*z +
3225600*a^5*b^9*c^8*d*g*j*z + 2949120*a^6*b^6*c^10*f*g*h*z + 2396160*a^5*b^
8*c^9*d*h*i*z - 1428480*a^3*b^14*c^5*d*f*k*z - 737280*a^5*b^8*c^9*f*g*h*z -
 161280*a^4*b^11*c^7*d*g*j*z + 92160*a^4*b^10*c^8*f*g*h*z + 73728*a^2*b^16*
c^4*d*f*k*z - 50688*a^3*b^12*c^7*d*h*i*z - 27648*a^4*b^10*c^8*d*h*i*z - 460
8*a^3*b^12*c^7*f*g*h*z + 4608*a^2*b^14*c^6*d*h*i*z - 58982400*a^5*b^6*c^11*
d*f*g*z + 11796480*a^7*b^3*c^12*e*f*h*z + 8847360*a^5*b^7*c^10*d*f*i*z - 66
35520*a^5*b^7*c^10*d*g*h*z - 6451200*a^5*b^8*c^9*d*e*j*z - 5898240*a^6*b^5*
c^11*e*f*h*z - 3809280*a^4*b^9*c^9*d*f*i*z + 2359296*a^6*b^5*c^11*d*f*i*z +
 1474560*a^5*b^7*c^10*e*f*h*z + 681984*a^3*b^11*c^8*d*f*i*z + 322560*a^4*b^
10*c^8*d*e*j*z - 276480*a^4*b^9*c^9*d*g*h*z - 184320*a^4*b^9*c^9*e*f*h*z +
179712*a^3*b^11*c^8*d*g*h*z - 55296*a^2*b^13*c^7*d*f*i*z - 13824*a^2*b^13*c
^7*d*g*h*z + 9216*a^3*b^11*c^8*e*f*h*z + 16220160*a^4*b^8*c^10*d*f*g*z + 13

271040*a^5*b^6*c^11*d*e*h*z - 2396160*a^3*b^10*c^9*d*f*g*z + 552960*a^4*b^8
*c^10*d*e*h*z - 359424*a^3*b^10*c^9*d*e*h*z + 175104*a^2*b^12*c^8*d*f*g*z +
 27648*a^2*b^12*c^8*d*e*h*z - 32440320*a^4*b^7*c^11*d*e*f*z + 4792320*a^3*b
^9*c^10*d*e*f*z - 350208*a^2*b^11*c^9*d*e*f*z + 1439170560*a^10*b*c^11*d*h*
k*z - 3361603584*a^10*b^3*c^9*d*j*k*z + 603979776*a^10*b*c^11*e*g*k*z + 407
371776*a^12*b*c^9*h*j*k*z + 201326592*a^11*b*c^10*g*i*k*z + 346816512*a^7*b
*c^14*d^2*g*z + 129761280*a^11*b*c^10*h^2*k*z + 121896960*a^10*b*c^11*f^2*k
*z + 458752*a^6*b^15*c*i*k^2*z + 19660800*a^11*b*c^10*g*j^2*z + 49152*a^5*b
^16*c*g*k^2*z + 7077888*a^9*b*c^12*g*h^2*z + 94464*a*b^17*c^4*d^2*k*z - 196
60800*a^8*b*c^13*f^2*g*z - 66816*a*b^14*c^7*d^2*i*z + 214272*a*b^13*c^8*d^2
*g*z - 428544*a*b^12*c^9*d^2*e*z + 2390753280*a^11*b^4*c^7*g*k^2*z - 241142
1696*a^6*b^7*c^9*d^2*k*z - 6603079680*a^8*b^3*c^11*d^2*k*z + 3715891200*a^9
*b*c^12*d^2*k*z - 880803840*a^10*c^12*d*f*k*z - 1623195648*a^10*b^6*c^6*g*k
^2*z - 402653184*a^11*c^11*e*i*k*z - 1509949440*a^12*b^2*c^8*g*k^2*z - 2097
15200*a^12*c^10*f*j*k*z - 330301440*a^9*c^13*d*e*j*z + 3019898880*a^12*b*c^
9*e*k^2*z - 125829120*a^11*c^11*f*h*k*z - 110100480*a^10*c^12*d*i*j*z - 198
180864*a^8*c^14*d*e*h*z - 15728640*a^11*c^11*h*i*j*z - 1226833920*a^9*b^7*c
^6*e*k^2*z - 47185920*a^10*c^12*e*h*j*z - 66060288*a^9*c^13*d*h*i*z - 10905
19040*a^12*b^3*c^7*i*k^2*z + 1022754816*a^6*b^2*c^14*d^2*e*z + 5216108544*a
^7*b^5*c^10*d^2*k*z + 754974720*a^9*b^2*c^11*e^2*k*z + 721529856*a^5*b^9*c^
8*d^2*k*z + 613416960*a^9*b^8*c^5*g*k^2*z - 642318336*a^5*b^4*c^13*d^2*e*z
 - 4781506560*a^11*b^3*c^8*e*k^2*z - 398131200*a^12*b^3*c^7*j^2*k*z - 511377
408*a^6*b^3*c^13*d^2*g*z - 377487360*a^8*b^4*c^10*e^2*k*z + 285212672*a^11*
b^5*c^6*i*k^2*z + 199065600*a^11*b^5*c^6*j^2*k*z + 279183360*a^8*b^9*c^5*e*
k^2*z + 321159168*a^5*b^5*c^12*d^2*g*z + 188743680*a^9*b^4*c^9*g^2*k*z + 13
2120576*a^10*b^7*c^5*i*k^2*z - 150994944*a^10*b^2*c^10*g^2*k*z - 111411200*
a^9*b^9*c^4*i*k^2*z - 126812160*a^10*b^3*c^9*h^2*k*z + 225312768*a^7*b^2*c^
13*d^2*i*z - 139591680*a^8*b^10*c^4*g*k^2*z - 49766400*a^10*b^7*c^5*j^2*k*z
 - 145463040*a^4*b^11*c^7*d^2*k*z - 94371840*a^8*b^6*c^8*g^2*k*z + 22339584
0*a^4*b^6*c^12*d^2*e*z + 33751040*a^8*b^11*c^3*i*k^2*z - 78970880*a^9*b^3*c
^10*f^2*k*z + 94371840*a^7*b^6*c^9*e^2*k*z + 25165824*a^10*b^4*c^8*i^2*k*z
 + 6220800*a^9*b^9*c^4*j^2*k*z + 39223296*a^9*b^5*c^8*h^2*k*z - 311040*a^8*b
^11*c^3*j^2*k*z + 16777216*a^11*b^2*c^9*i^2*k*z - 10485760*a^9*b^6*c^7*i^2*
k*z - 5406720*a^7*b^13*c^2*i*k^2*z + 1376256*a^7*b^10*c^5*i^2*k*z - 1310720
*a^8*b^8*c^6*i^2*k*z - 262144*a^6*b^12*c^4*i^2*k*z + 16384*a^5*b^14*c^3*i^2
*k*z + 10354688*a^11*b^2*c^9*i*j^2*z + 23592960*a^7*b^8*c^7*g^2*k*z + 38559
744*a^7*b^7*c^8*f^2*k*z + 19169280*a^7*b^12*c^3*g*k^2*z - 2048000*a^9*b^6*c
^7*i*j^2*z - 1520640*a^7*b^9*c^6*h^2*k*z - 1105920*a^8*b^7*c^7*h^2*k*z + 84
9920*a^8*b^8*c^6*i*j^2*z - 393216*a^10*b^4*c^8*i*j^2*z + 195840*a^6*b^11*c^
5*h^2*k*z - 145920*a^7*b^10*c^5*i*j^2*z + 11520*a^5*b^13*c^4*h^2*k*z + 1100
8*a^6*b^12*c^4*i*j^2*z - 2304*a^4*b^15*c^3*h^2*k*z - 256*a^5*b^14*c^3*i*j^2
*z - 25362432*a^10*b^3*c^9*g*j^2*z - 24739840*a^8*b^5*c^9*f^2*k*z - 3833856
0*a^7*b^11*c^4*e*k^2*z - 2949120*a^6*b^10*c^6*g^2*k*z - 1474560*a^6*b^14*c^
2*g*k^2*z + 50724864*a^10*b^2*c^10*e*j^2*z + 147456*a^5*b^12*c^5*g^2*k*z -
15150080*a^6*b^9*c^7*f^2*k*z + 13271040*a^9*b^5*c^8*g*j^2*z - 111697920*a^4

*b^7*c^11*d^2*g*z - 3563520*a^8*b^7*c^7*g*j^2*z + 3538944*a^9*b^2*c^11*h^2*i*z + 2912000*a^5*b^11*c^6*f^2*k*z - 737280*a^7*b^6*c^9*h^2*i*z + 506880*a^7*b^9*c^6*g*j^2*z - 291840*a^4*b^13*c^5*f^2*k*z + 276480*a^6*b^8*c^8*h^2*i*z - 41472*a^5*b^10*c^7*h^2*i*z - 34560*a^6*b^11*c^5*g*j^2*z + 14080*a^3*b^15*c^4*f^2*k*z + 2304*a^4*b^12*c^6*h^2*i*z + 768*a^5*b^13*c^4*g*j^2*z - 256*a^2*b^17*c^3*f^2*k*z - 11796480*a^6*b^8*c^8*e^2*k*z - 26542080*a^9*b^4*c^9*e*j^2*z + 19837440*a^3*b^13*c^6*d^2*k*z + 2949120*a^6*b^13*c^3*e*k^2*z + 589824*a^5*b^10*c^7*e^2*k*z - 98304*a^5*b^15*c^2*e*k^2*z - 10354688*a^8*b^2*c^12*f^2*i*z - 43646976*a^6*b^4*c^12*d^2*i*z - 8847360*a^8*b^3*c^11*g*h^2*z + 7127040*a^8*b^6*c^8*e*j^2*z + 4423680*a^7*b^5*c^10*g*h^2*z + 2048000*a^6*b^6*c^10*f^2*i*z - 1771776*a^2*b^15*c^5*d^2*k*z - 1105920*a^6*b^7*c^9*g*h^2*z - 1013760*a^7*b^8*c^7*e*j^2*z - 849920*a^5*b^8*c^9*f^2*i*z + 393216*a^7*b^4*c^11*f^2*i*z + 145920*a^4*b^10*c^8*f^2*i*z + 138240*a^5*b^9*c^8*g*h^2*z + 69120*a^6*b^10*c^6*e*j^2*z - 11008*a^3*b^12*c^7*f^2*i*z - 6912*a^4*b^11*c^7*g*h^2*z - 1536*a^5*b^12*c^5*e*j^2*z + 256*a^2*b^14*c^6*f^2*i*z - 32587776*a^5*b^6*c^11*d^2*i*z + 25362432*a^7*b^3*c^12*f^2*g*z + 21657600*a^4*b^8*c^10*d^2*i*z + 17694720*a^8*b^2*c^12*e*h^2*z - 50724864*a^7*b^2*c^13*e*f^2*z - 13271040*a^6*b^5*c^11*f^2*g*z - 8847360*a^7*b^4*c^11*e*h^2*z - 5810688*a^3*b^10*c^9*d^2*i*z + 3563520*a^5*b^7*c^10*f^2*g*z + 2211840*a^6*b^6*c^10*e*h^2*z + 845568*a^2*b^12*c^8*d^2*i*z - 506880*a^4*b^9*c^9*f^2*g*z - 276480*a^5*b^8*c^9*e*h^2*z + 34560*a^3*b^11*c^8*f^2*g*z + 13824*a^4*b^10*c^8*e*h^2*z - 768*a^2*b^13*c^7*f^2*g*z + 26542080*a^6*b^4*c^12*e*f^2*z + 23362560*a^3*b^9*c^10*d^2*g*z - 46725120*a^3*b^8*c^11*d^2*e*z - 7127040*a^5*b^6*c^11*e*f^2*z - 2965248*a^2*b^11*c^9*d^2*g*z + 1013760*a^4*b^8*c^10*e*f^2*z - 69120*a^3*b^10*c^9*e*f^2*z + 1536*a^2*b^12*c^8*e*f^2*z + 5930496*a^2*b^10*c^10*d^2*e*z + 1006632960*a^13*b*c^8*i*k^2*z + 3246391296*a^10*b^5*c^7*e*k^2*z + 318504960*a^13*b*c^8*j^2*k*z + 61538304*a^10*b^10*c^2*k^3*z - 603979776*a^10*c^12*e^2*k*z - 693633024*a^7*c^15*d^2*e*z - 231211008*a^8*c^14*d^2*i*z - 67108864*a^12*c^10*i^2*k*z - 13107200*a^12*c^10*i*j^2*z - 16384*a^5*b^17*i*k^2*z - 39321600*a^11*c^11*e*j^2*z - 4718592*a^10*c^12*h^2*i*z - 2304*b^19*c^3*d^2*k*z + 13107200*a^9*c^13*f^2*i*z + 2304*b^16*c^6*d^2*i*z - 14155776*a^9*c^13*e*h^2*z + 39321600*a^8*c^14*e*f^2*z - 4833280*a^9*b^12*c*k^3*z - 6912*b^15*c^7*d^2*g*z + 6962544640*a^14*b^2*c^6*k^3*z + 13824*b^14*c^8*d^2*e*z + 1876951040*a^12*b^6*c^4*k^3*z - 4844421120*a^13*b^4*c^5*k^3*z - 437780480*a^11*b^8*c^3*k^3*z - 4294967296*a^15*c^7*k^3*z + 163840*a^8*b^14*k^3*z + 6144000*a^10*b*c^8*f*i*j*k - 5898240*a^10*b*c^8*g*h*j*k - 41287680*a^9*b*c^9*d*g*j*k + 4472832*a^9*b*c^9*f*h*i*k + 18432000*a^9*b*c^9*e*f*j*k + 3391488*a^8*b*c^10*e*h*i*j + 1228800*a^8*b*c^10*f*g*i*j - 24772608*a^8*b*c^10*d*g*h*k + 13418496*a^8*b*c^10*e*f*h*k + 11649024*a^8*b*c^10*d*f*i*k + 737280*a^7*b*c^11*f*g*h*i - 768*a*b^15*c^3*d*f*i*k - 19307520*a^7*b*c^11*d*f*h*j + 16367616*a^7*b*c^11*d*e*i*j + 3686400*a^7*b*c^11*e*f*g*j + 34947072*a^7*b*c^11*d*e*f*k + 2304*a*b^14*c^4*d*f*g*k - 180*a*b^13*c^5*d*f*h*j + 11059200*a^6*b*c^12*d*e*h*i + 5160960*a^6*b*c^12*d*f*g*i + 2211840*a^6*b*c^12*e*f*g*h - 4608*a*b^13*c^5*d*e*f*k - 2304*a*b^11*c^7*d*f*g*i + 4608*a*b^10*c^8*d*e*f*i + 15482880*a^5*b*c^13*d*e*f*g - 13824*a*b^9*c^9*d*e*f*g - 225976320

*a^8*b^2*c^9*d*e*j*k + 112988160*a^8*b^3*c^8*d*g*j*k - 11427840*a^10*b^2*c^
7*h*i*j*k - 4177920*a^9*b^4*c^6*h*i*j*k + 1399296*a^8*b^6*c^5*h*i*j*k - 268
80*a^6*b^10*c^3*h*i*j*k + 16128*a^7*b^8*c^4*h*i*j*k - 61562880*a^9*b^2*c^8*
d*i*j*k + 20090880*a^9*b^3*c^7*g*h*j*k + 119623680*a^7*b^4*c^8*d*e*j*k + 10
485760*a^9*b^3*c^7*f*i*j*k - 40181760*a^9*b^2*c^8*e*h*j*k - 3778560*a^8*b^5
*c^6*g*h*j*k - 137797632*a^7*b^2*c^10*d*e*h*k - 1248768*a^7*b^7*c^5*f*i*j*k
 + 229376*a^6*b^9*c^4*f*i*j*k + 220160*a^8*b^5*c^6*f*i*j*k - 209664*a^7*b^7
*c^5*g*h*j*k + 80640*a^6*b^9*c^4*g*h*j*k - 8960*a^5*b^11*c^3*f*i*j*k - 5981
1840*a^7*b^5*c^7*d*g*j*k + 53084160*a^8*b^2*c^9*e*g*i*k - 11120640*a^8*b^4*
c^7*f*g*j*k + 10455552*a^7*b^6*c^6*d*i*j*k - 9216000*a^9*b^2*c^8*f*g*j*k +
7557120*a^8*b^4*c^7*e*h*j*k + 7397376*a^8*b^3*c^8*f*h*i*k + 5230080*a^7*b^6
*c^6*f*g*j*k - 37675008*a^8*b^2*c^9*d*h*i*k - 3633408*a^6*b^8*c^5*d*i*j*k +
 2211840*a^8*b^4*c^7*d*i*j*k + 68898816*a^7*b^3*c^9*d*g*h*k - 1695744*a^8*b
^2*c^9*g*h*i*j - 1400832*a^7*b^4*c^8*g*h*i*j + 967680*a^7*b^5*c^7*f*h*i*k -
 783360*a^6*b^7*c^6*f*h*i*k - 741888*a^6*b^8*c^5*f*g*j*k + 499968*a^5*b^10*
c^4*d*i*j*k + 419328*a^7*b^6*c^6*e*h*j*k - 253440*a^6*b^6*c^7*g*h*i*j - 161
280*a^6*b^8*c^5*e*h*j*k + 42240*a^5*b^9*c^5*f*h*i*k + 26880*a^5*b^10*c^4*f*
g*j*k - 26880*a^4*b^12*c^3*d*i*j*k + 13824*a^4*b^11*c^4*f*h*i*k + 11520*a^5
*b^8*c^6*g*h*i*j - 768*a^3*b^13*c^3*f*h*i*k + 22241280*a^8*b^3*c^8*e*f*j*k
+ 14222592*a^6*b^7*c^6*d*g*j*k - 10460160*a^7*b^5*c^7*e*f*j*k + 8847360*a^7
*b^4*c^8*e*g*i*k - 7741440*a^7*b^4*c^8*f*g*h*k - 7077888*a^6*b^6*c^7*e*g*i*
k + 6935040*a^6*b^6*c^7*d*h*i*k - 6709248*a^8*b^2*c^9*f*g*h*k - 3612672*a^7
*b^4*c^8*d*h*i*k + 2801664*a^7*b^3*c^9*e*h*i*j + 2506752*a^7*b^3*c^9*f*g*i*
j + 2419200*a^6*b^6*c^7*f*g*h*k - 1661184*a^5*b^9*c^5*d*g*j*k + 1483776*a^6
*b^7*c^6*e*f*j*k - 1463040*a^5*b^8*c^6*d*h*i*k + 884736*a^5*b^8*c^6*e*g*i*k
 + 838656*a^6*b^5*c^8*f*g*i*j + 506880*a^6*b^5*c^8*e*h*i*j + 80640*a^4*b^11
*c^4*d*g*j*k - 53760*a^5*b^9*c^5*e*f*j*k - 53760*a^5*b^7*c^7*f*g*i*j - 4608
0*a^4*b^10*c^5*f*g*h*k - 34560*a^5*b^8*c^6*f*g*h*k + 25344*a^3*b^12*c^4*d*h
*i*k - 23040*a^5*b^7*c^7*e*h*i*j + 13824*a^4*b^10*c^5*d*h*i*k + 2304*a^3*b^
12*c^4*f*g*h*k - 2304*a^2*b^14*c^3*d*h*i*k - 29030400*a^6*b^5*c^8*d*g*h*k +
 28606464*a^7*b^3*c^9*d*f*i*k - 28445184*a^6*b^6*c^7*d*e*j*k + 58060800*a^6
*b^4*c^9*d*e*h*k + 15482880*a^7*b^3*c^9*e*f*h*k - 8183808*a^7*b^2*c^10*d*g*
i*j - 6718464*a^6*b^5*c^8*d*f*i*k - 5087232*a^7*b^2*c^10*e*g*h*j - 5013504*
a^7*b^2*c^10*e*f*i*j - 4838400*a^6*b^5*c^8*e*f*h*k + 4112640*a^5*b^7*c^7*d*
g*h*k - 3663360*a^5*b^7*c^7*d*f*i*k + 3322368*a^5*b^8*c^6*d*e*j*k - 2285568
*a^6*b^4*c^9*d*g*i*j + 1896960*a^4*b^9*c^6*d*f*i*k + 1843200*a^6*b^3*c^10*f
*g*h*i - 1677312*a^6*b^4*c^9*e*f*i*j - 1658880*a^6*b^4*c^9*e*g*h*j + 683458
56*a^6*b^3*c^10*d*e*f*k + 783360*a^5*b^5*c^9*f*g*h*i + 741888*a^5*b^6*c^8*d
*g*i*j - 34172928*a^6*b^4*c^9*d*f*g*k - 340992*a^3*b^11*c^5*d*f*i*k - 16128
0*a^4*b^10*c^5*d*e*j*k + 138240*a^4*b^9*c^6*d*g*h*k + 107520*a^5*b^6*c^8*e*
f*i*j + 92160*a^4*b^9*c^6*e*f*h*k - 89856*a^3*b^11*c^5*d*g*h*k - 80640*a^4*
b^8*c^7*d*g*i*j + 69120*a^5*b^7*c^7*e*f*h*k + 69120*a^5*b^6*c^8*e*g*h*j + 2
7648*a^2*b^13*c^4*d*f*i*k + 18432*a^4*b^7*c^8*f*g*h*i + 6912*a^2*b^13*c^4*d
*g*h*k - 4608*a^3*b^11*c^5*e*f*h*k - 2304*a^3*b^9*c^7*f*g*h*i + 27164160*a^
5*b^6*c^8*d*f*g*k - 22164480*a^6*b^3*c^10*d*f*h*j - 54328320*a^5*b^5*c^9*d*

e*f*k - 17473536*a^7*b^2*c^10*d*f*g*k - 8225280*a^5*b^6*c^8*d*e*h*k - 80870
40*a^4*b^8*c^7*d*f*g*k + 5677056*a^6*b^3*c^10*e*f*g*j - 5529600*a^6*b^2*c^1
1*d*g*h*i + 4571136*a^6*b^3*c^10*d*e*i*j - 3686400*a^6*b^2*c^11*e*f*h*i + 2
805120*a^5*b^5*c^9*d*f*h*j - 2211840*a^5*b^4*c^10*d*g*h*i - 1566720*a^5*b^4
*c^10*e*f*h*i - 1483776*a^5*b^5*c^9*d*e*i*j + 1198080*a^3*b^10*c^6*d*f*g*k
+ 437184*a^4*b^7*c^8*d*f*h*j - 322560*a^5*b^5*c^9*e*f*g*j + 317952*a^4*b^6*
c^9*d*g*h*i - 276480*a^4*b^8*c^7*d*e*h*k + 179712*a^3*b^10*c^6*d*e*h*k + 16
1280*a^4*b^7*c^8*d*e*i*j - 146268*a^3*b^9*c^7*d*f*h*j - 87552*a^2*b^12*c^5*
d*f*g*k - 36864*a^4*b^6*c^9*e*f*h*i - 13824*a^2*b^12*c^5*d*e*h*k + 9360*a^2
*b^11*c^6*d*f*h*j + 6912*a^3*b^8*c^8*d*g*h*i - 6912*a^2*b^10*c^7*d*g*h*i +
4608*a^3*b^8*c^8*e*f*h*i - 24551424*a^6*b^2*c^11*d*e*g*j + 16174080*a^4*b^7
*c^8*d*e*f*k + 5419008*a^5*b^4*c^10*d*e*g*j + 5160960*a^5*b^3*c^11*d*f*g*i
+ 4423680*a^5*b^3*c^11*e*f*g*h + 4423680*a^5*b^3*c^11*d*e*h*i - 2396160*a^3
*b^9*c^7*d*e*f*k - 635904*a^4*b^5*c^10*d*e*h*i - 483840*a^4*b^6*c^9*d*e*g*j
 - 354816*a^3*b^7*c^9*d*f*g*i + 322560*a^4*b^5*c^10*d*f*g*i + 175104*a^2*b^
11*c^6*d*e*f*k + 138240*a^4*b^5*c^10*e*f*g*h + 59904*a^2*b^9*c^8*d*f*g*i -
13824*a^3*b^7*c^9*e*f*g*h - 13824*a^3*b^7*c^9*d*e*h*i + 13824*a^2*b^9*c^8*d
*e*h*i - 16588800*a^5*b^2*c^12*d*e*g*h - 10321920*a^5*b^2*c^12*d*e*f*i + 16
58880*a^4*b^4*c^11*d*e*g*h + 709632*a^3*b^6*c^10*d*e*f*i - 645120*a^4*b^4*c
^11*d*e*f*i + 124416*a^3*b^6*c^10*d*e*g*h - 119808*a^2*b^8*c^9*d*e*f*i - 41
472*a^2*b^8*c^9*d*e*g*h + 7741440*a^4*b^3*c^12*d*e*f*g - 2903040*a^3*b^5*c^
11*d*e*f*g + 387072*a^2*b^7*c^10*d*e*f*g - 381026304*a^11*b*c^7*d*j*k^2 - 2
41827840*a^10*b*c^8*d*h*k^2 - 65667072*a^12*b*c^6*h*j*k^2 - 169344*a^7*b^11
*c*h*j*k^2 - 25165824*a^11*b*c^7*g*i*k^2 - 4915200*a^11*b*c^7*g*j^2*k - 530
84160*a^8*b*c^10*e^2*i*k - 75497472*a^10*b*c^8*e*g*k^2 - 86704128*a^7*b*c^1
1*d^2*g*k + 565248*a^9*b*c^9*h*i^2*j - 168448*a^6*b^12*c*f*j*k^2 - 24576*a^
5*b^13*c*g*i*k^2 - 1769472*a^9*b*c^9*g*h^2*k - 17694720*a^9*b*c^9*e*i^2*k -
 411264*a^5*b^13*c*d*j*k^2 - 11520*a^4*b^14*c*f*h*k^2 + 4915200*a^8*b*c^10*
f^2*g*k + 2580480*a^9*b*c^9*e*i*j^2 - 2496000*a^9*b*c^9*f*h*j^2 - 1543680*a
^8*b*c^10*f*h^2*j + 33408*a*b^14*c^4*d^2*i*k - 59512320*a^6*b*c^12*d^2*f*j
+ 5087232*a^7*b*c^11*e^2*h*j + 2727936*a^8*b*c^10*d*i^2*j - 26496*a^3*b^15*
c*d*h*k^2 + 1105920*a^7*b*c^11*e*h^2*i - 107136*a*b^13*c^5*d^2*g*k + 10260*
a*b^12*c^6*d^2*h*j - 10616832*a^6*b*c^12*e^2*g*i - 3538944*a^7*b*c^11*e*g*i
^2 + 1843200*a^7*b*c^11*d*h*i^2 - 18432*a^2*b^16*c*d*f*k^2 - 15552000*a^8*b
*c^10*d*f*j^2 + 24551424*a^6*b*c^12*d*e^2*j - 37062144*a^5*b*c^13*d^2*f*h +
 2580480*a^6*b*c^12*e*f^2*i + 214272*a*b^12*c^6*d^2*e*k + 65664*a*b^10*c^8*
d^2*g*i - 25074*a*b^11*c^7*d^2*f*j + 420*a*b^12*c^6*d*f^2*j + 6*a*b^15*c^3*
d*f*j^2 + 23224320*a^5*b*c^13*d^2*e*i + 384*a*b^12*c^6*d*f*i^2 - 5985792*a^
6*b*c^12*d*f*h^2 + 206010*a*b^9*c^9*d^2*f*h - 131328*a*b^9*c^9*d^2*e*i - 63
00*a*b^10*c^8*d*f^2*h + 1350*a*b^11*c^7*d*f*h^2 + 16588800*a^5*b*c^13*d*e^2
*h + 3456*a*b^10*c^8*d*f*g^2 + 435456*a*b^8*c^10*d^2*e*g + 13824*a*b^8*c^10
*d*e^2*f + 3932160*a^11*c^8*h*i*j*k + 27525120*a^10*c^9*d*i*j*k + 82575360*
a^9*c^10*d*e*j*k + 11796480*a^10*c^9*e*h*j*k + 16515072*a^9*c^10*d*h*i*k +
49545216*a^8*c^11*d*e*h*k - 2457600*a^8*c^11*e*f*i*j - 1474560*a^7*c^12*e*f
*h*i - 10321920*a^6*c^13*d*e*f*i + 737077248*a^10*b^3*c^6*d*j*k^2 - 5188147

20*a^9*b^5*c^5*d*j*k^2 + 441354240*a^9*b^3*c^7*d*h*k^2 - 429871104*a^6*b^2*
c^11*d^2*e*k - 272212992*a^8*b^5*c^6*d*h*k^2 + 305731584*a^5*b^4*c^10*d^2*e
*k + 192412800*a^8*b^7*c^4*d*j*k^2 + 111912960*a^11*b^3*c^5*h*j*k^2 + 21493
5552*a^6*b^3*c^10*d^2*g*k + 202427136*a^7*b^6*c^6*d*f*k^2 - 49904640*a^10*b
^5*c^4*h*j*k^2 - 178513920*a^8*b^4*c^7*d*f*k^2 - 152865792*a^5*b^5*c^9*d^2*
g*k - 114388992*a^7*b^2*c^10*d^2*i*k + 94961664*a^10*b^2*c^7*e*i*k^2 - 9039
872*a^11*b^2*c^6*i*j^2*k - 56494080*a^10*b^4*c^5*f*j*k^2 - 2052096*a^10*b^4
*c^5*i*j^2*k + 1327360*a^9*b^6*c^4*i*j^2*k - 158080*a^8*b^8*c^3*i*j^2*k - 4
7480832*a^10*b^3*c^6*g*i*k^2 + 45576960*a^9*b^6*c^4*f*j*k^2 + 7954560*a^9*b
^7*c^3*h*j*k^2 - 104693760*a^9*b^3*c^7*e*g*k^2 + 142080*a^8*b^9*c^2*h*j*k^2
 + 16017408*a^10*b^3*c^6*g*j^2*k - 4930560*a^9*b^5*c^5*g*j^2*k - 3649536*a^
9*b^2*c^8*h^2*i*k - 1843200*a^8*b^4*c^7*h^2*i*k + 85524480*a^8*b^5*c^6*e*g*
k^2 + 474240*a^8*b^7*c^4*g*j^2*k + 288000*a^7*b^6*c^6*h^2*i*k + 63360*a^6*b
^8*c^5*h^2*i*k - 8064*a^5*b^10*c^4*h^2*i*k - 1152*a^4*b^12*c^3*h^2*i*k - 15
437824*a^11*b^2*c^6*f*j*k^2 - 32034816*a^10*b^2*c^7*e*j^2*k - 14369280*a^8*
b^8*c^3*f*j*k^2 - 13271040*a^8*b^3*c^8*g^2*i*k + 80267904*a^7*b^7*c^5*d*h*k
^2 + 79626240*a^7*b^2*c^10*e^2*g*k + 11059200*a^9*b^5*c^5*g*i*k^2 + 8847360
*a^9*b^2*c^8*g*i^2*k - 42113280*a^7*b^9*c^3*d*j*k^2 + 6389760*a^8*b^7*c^4*g
*i*k^2 + 5898240*a^8*b^4*c^7*g*i^2*k - 37601280*a^9*b^4*c^6*f*h*k^2 - 29491
20*a^7*b^9*c^3*g*i*k^2 + 2242560*a^7*b^10*c^2*f*j*k^2 - 2211840*a^7*b^5*c^7
*g^2*i*k + 1769472*a^6*b^7*c^6*g^2*i*k + 749568*a^8*b^3*c^8*h*i^2*j - 44236
8*a^7*b^6*c^6*g*i^2*k + 442368*a^6*b^11*c^2*g*i*k^2 - 442368*a^6*b^8*c^5*g*
i^2*k + 317952*a^7*b^5*c^7*h*i^2*j - 221184*a^5*b^9*c^5*g^2*i*k + 73728*a^5
*b^10*c^4*g*i^2*k + 38400*a^6*b^7*c^6*h*i^2*j - 1920*a^5*b^9*c^5*h*i^2*j +
9861120*a^9*b^4*c^6*e*j^2*k - 110280960*a^4*b^6*c^9*d^2*e*k - 93330432*a^6*
b^8*c^5*d*f*k^2 + 24645888*a^8*b^6*c^5*f*h*k^2 + 6359040*a^8*b^3*c^8*g*h^2*
k - 22118400*a^9*b^4*c^6*e*i*k^2 - 3862528*a^8*b^2*c^9*f^2*i*k - 2248704*a^
7*b^4*c^8*f^2*i*k - 1290240*a^9*b^2*c^8*g*i*j^2 - 948480*a^8*b^6*c^5*e*j^2*
k - 860160*a^8*b^4*c^7*g*i*j^2 - 414720*a^7*b^5*c^7*g*h^2*k + 303360*a^6*b^
6*c^7*f^2*i*k + 266880*a^5*b^8*c^6*f^2*i*k - 224640*a^6*b^7*c^6*g*h^2*k - 8
0640*a^7*b^6*c^6*g*i*j^2 - 72960*a^4*b^10*c^5*f^2*i*k + 17280*a^5*b^9*c^5*g
*h^2*k + 12672*a^6*b^8*c^5*g*i*j^2 + 5504*a^3*b^12*c^4*f^2*i*k + 3456*a^4*b
^11*c^4*g*h^2*k - 384*a^5*b^10*c^4*g*i*j^2 - 128*a^2*b^14*c^3*f^2*i*k + 302
65344*a^6*b^4*c^9*d^2*i*k - 12779520*a^8*b^6*c^5*e*i*k^2 - 11796480*a^8*b^3
*c^8*e*i^2*k - 8847360*a^7*b^3*c^9*e^2*i*k - 7925760*a^10*b^2*c^7*f*h*k^2 +
 7077888*a^6*b^5*c^8*e^2*i*k - 39813120*a^7*b^3*c^9*e*g^2*k - 73175040*a^9*
b^2*c^8*d*f*k^2 + 5898240*a^7*b^8*c^4*e*i*k^2 + 5542272*a^6*b^11*c^2*d*j*k^
2 - 5420160*a^7*b^8*c^4*f*h*k^2 + 55140480*a^4*b^7*c^8*d^2*g*k + 1271808*a^
7*b^3*c^9*g^2*h*j - 1040384*a^8*b^2*c^9*f*i^2*j + 884736*a^7*b^5*c^7*e*i^2*
k - 884736*a^6*b^10*c^3*e*i*k^2 + 884736*a^6*b^7*c^6*e*i^2*k - 884736*a^5*b
^7*c^7*e^2*i*k - 697344*a^7*b^4*c^8*f*i^2*j + 414720*a^6*b^5*c^8*g^2*h*j +
226560*a^6*b^10*c^3*f*h*k^2 - 147456*a^5*b^9*c^5*e*i^2*k - 121856*a^6*b^6*c
^7*f*i^2*j + 82560*a^5*b^12*c^2*f*h*k^2 + 49152*a^5*b^12*c^2*e*i*k^2 - 1728
0*a^5*b^7*c^7*g^2*h*j + 8960*a^5*b^8*c^6*f*i^2*j + 14194944*a^5*b^6*c^8*d^2
*i*k - 12718080*a^8*b^2*c^9*e*h^2*k - 10615680*a^4*b^8*c^7*d^2*i*k - 265420

80*a^6*b^4*c^9*e^2*g*k - 23592960*a^7*b^7*c^5*e*g*k^2 - 5142528*a^8*b^3*c^8
*f*h*j^2 + 5068800*a^7*b^2*c^10*f^2*h*j - 3755520*a^7*b^3*c^9*f*h^2*j + 333
6192*a^7*b^3*c^9*f^2*g*k + 3000960*a^6*b^4*c^9*f^2*h*j + 2893824*a^3*b^10*c
^6*d^2*i*k + 1720320*a^8*b^3*c^8*e*i*j^2 + 1704960*a^6*b^5*c^8*f^2*g*k - 13
07520*a^5*b^7*c^7*f^2*g*k - 1085760*a^6*b^5*c^8*f*h^2*j - 959040*a^7*b^5*c^
7*f*h*j^2 + 829440*a^7*b^4*c^8*e*h^2*k - 552960*a^7*b^2*c^10*g*h^2*i - 5529
60*a^6*b^4*c^9*g*h^2*i + 449280*a^6*b^6*c^7*e*h^2*k - 422784*a^2*b^12*c^5*d
^2*i*k + 253440*a^4*b^9*c^6*f^2*g*k + 161280*a^7*b^5*c^7*e*i*j^2 - 145152*a
^5*b^6*c^8*g*h^2*i + 103200*a^6*b^7*c^6*f*h*j^2 + 41280*a^5*b^6*c^8*f^2*h*j
 - 37188*a^4*b^8*c^7*f^2*h*j - 34560*a^5*b^8*c^6*e*h^2*k - 25344*a^6*b^7*c^
6*e*i*j^2 - 17280*a^3*b^11*c^5*f^2*g*k + 13536*a^5*b^7*c^7*f*h^2*j - 6912*a
^4*b^10*c^5*e*h^2*k + 5490*a^4*b^9*c^6*f*h^2*j - 3456*a^4*b^8*c^7*g*h^2*i +
 1980*a^3*b^10*c^6*f^2*h*j + 810*a^5*b^9*c^5*f*h*j^2 + 768*a^5*b^9*c^5*e*i*
j^2 + 384*a^2*b^13*c^4*f^2*g*k - 270*a^4*b^11*c^4*f*h*j^2 - 180*a^3*b^11*c^
5*f*h^2*j - 30*a^2*b^12*c^5*f^2*h*j + 6*a^3*b^13*c^3*f*h*j^2 + 30067200*a^6
*b^2*c^11*d^2*h*j + 13271040*a^6*b^5*c^8*e*g^2*k - 10857600*a^6*b^9*c^4*d*h
*k^2 + 2949120*a^6*b^9*c^4*e*g*k^2 + 2654208*a^5*b^6*c^8*e^2*g*k + 2125824*
a^7*b^3*c^9*d*i^2*j + 1658880*a^6*b^3*c^10*e^2*h*j - 1419264*a^6*b^4*c^9*f*
g^2*j - 1327104*a^5*b^7*c^7*e*g^2*k - 921600*a^7*b^2*c^10*f*g^2*j - 737280*
a^7*b^2*c^10*f*h*i^2 - 568320*a^6*b^4*c^9*f*h*i^2 + 207360*a^4*b^13*c^2*d*h
*k^2 - 147456*a^5*b^11*c^3*e*g*k^2 - 136704*a^5*b^6*c^8*f*h*i^2 + 133632*a^
6*b^5*c^8*d*i^2*j - 96768*a^5*b^7*c^7*d*i^2*j + 80640*a^5*b^6*c^8*f*g^2*j -
 69120*a^5*b^5*c^9*e^2*h*j + 13440*a^4*b^9*c^6*d*i^2*j - 5760*a^5*b^11*c^3*
d*h*k^2 - 2304*a^4*b^8*c^7*f*h*i^2 + 384*a^3*b^10*c^6*f*h*i^2 + 11930112*a^
8*b^2*c^9*d*h*j^2 - 11646720*a^3*b^9*c^7*d^2*g*k + 8432640*a^7*b^2*c^10*d*h
^2*j + 24140160*a^5*b^10*c^4*d*f*k^2 - 6672384*a^7*b^2*c^10*e*f^2*k + 44501
76*a^7*b^4*c^8*d*h*j^2 + 4337280*a^6*b^4*c^9*d*h^2*j - 3870720*a^8*b^2*c^9*
e*g*j^2 - 3409920*a^6*b^4*c^9*e*f^2*k - 2885760*a^5*b^4*c^10*d^2*h*j - 2844
288*a^4*b^6*c^9*d^2*h*j + 2615040*a^5*b^6*c^8*e*f^2*k - 1687680*a^6*b^6*c^7
*d*h*j^2 + 1482624*a^2*b^11*c^6*d^2*g*k - 1290240*a^6*b^2*c^11*f^2*g*i + 11
05920*a^6*b^3*c^10*e*h^2*i + 1019412*a^3*b^8*c^8*d^2*h*j - 1007424*a^5*b^6*
c^8*d*h^2*j - 860160*a^5*b^4*c^10*f^2*g*i - 645120*a^7*b^4*c^8*e*g*j^2 - 50
6880*a^4*b^8*c^7*e*f^2*k + 290304*a^5*b^5*c^9*e*h^2*i + 197460*a^5*b^8*c^6*
d*h*j^2 - 143802*a^2*b^10*c^7*d^2*h*j + 80640*a^6*b^6*c^7*e*g*j^2 - 80640*a
^4*b^6*c^9*f^2*g*i + 51948*a^4*b^8*c^7*d*h^2*j + 34560*a^3*b^10*c^6*e*f^2*k
 + 12672*a^3*b^8*c^8*f^2*g*i + 10800*a^3*b^10*c^6*d*h^2*j + 6912*a^4*b^7*c^
8*e*h^2*i - 2304*a^5*b^8*c^6*e*g*j^2 - 768*a^2*b^12*c^5*e*f^2*k - 684*a^3*b
^12*c^4*d*h*j^2 - 540*a^2*b^12*c^5*d*h^2*j - 384*a^2*b^10*c^7*f^2*g*i - 90*
a^4*b^10*c^5*d*h*j^2 + 18*a^2*b^14*c^3*d*h*j^2 + 23385600*a^6*b^2*c^11*d*f^
2*j + 23293440*a^3*b^8*c^8*d^2*e*k + 6137856*a^6*b^3*c^10*d*g^2*j - 5677056
*a^6*b^2*c^11*e^2*f*j + 5308416*a^6*b^2*c^11*e*g^2*i - 5308416*a^5*b^3*c^11
*e^2*g*i - 3786240*a^4*b^12*c^3*d*f*k^2 - 3538944*a^6*b^3*c^10*e*g*i^2 + 26
54208*a^5*b^4*c^10*e*g^2*i + 1658880*a^6*b^3*c^10*d*h*i^2 - 1354752*a^5*b^5
*c^9*d*g^2*j - 1105920*a^5*b^4*c^10*f*g^2*h - 884736*a^5*b^5*c^9*e*g*i^2 -
552960*a^6*b^2*c^11*f*g^2*h + 357120*a^3*b^14*c^2*d*f*k^2 + 322560*a^5*b^4*

c^10*e^2*f*j + 262656*a^5*b^5*c^9*d*h*i^2 + 120960*a^4*b^7*c^8*d*g^2*j - 55
296*a^4*b^7*c^8*d*h*i^2 - 34560*a^4*b^6*c^9*f*g^2*h + 3456*a^3*b^8*c^8*f*g^
2*h + 1152*a^3*b^9*c^7*d*h*i^2 + 1152*a^2*b^11*c^6*d*h*i^2 - 13149696*a^7*b
^3*c^9*d*f*j^2 - 11612160*a^5*b^2*c^12*d^2*g*i + 10906560*a^4*b^5*c^10*d^2*
f*j - 7418880*a^5*b^3*c^11*d^2*f*j + 3148992*a^6*b^5*c^8*d*f*j^2 - 2985696*
a^3*b^7*c^9*d^2*f*j - 2965248*a^2*b^10*c^7*d^2*e*k + 1720320*a^5*b^3*c^11*e
*f^2*i - 1658880*a^6*b^2*c^11*e*g*h^2 + 1596672*a^3*b^6*c^10*d^2*g*i - 1505
280*a^4*b^6*c^9*d*f^2*j - 829440*a^5*b^4*c^10*e*g*h^2 - 508032*a^2*b^8*c^9*
d^2*g*i + 378954*a^2*b^9*c^8*d^2*f*j + 362880*a^5*b^4*c^10*d*f^2*j + 296964
*a^3*b^8*c^8*d*f^2*j + 161280*a^4*b^5*c^10*e*f^2*i - 77070*a^4*b^9*c^6*d*f*
j^2 - 30240*a^5*b^7*c^7*d*f*j^2 - 25344*a^3*b^7*c^9*e*f^2*i - 20736*a^4*b^6
*c^9*e*g*h^2 - 19278*a^2*b^10*c^7*d*f^2*j + 8820*a^3*b^11*c^5*d*f*j^2 + 768
*a^2*b^9*c^8*e*f^2*i - 378*a^2*b^13*c^4*d*f*j^2 - 5419008*a^5*b^3*c^11*d*e^
2*j - 4423680*a^5*b^2*c^12*e^2*f*h + 4147200*a^5*b^3*c^11*d*g^2*h - 2580480
*a^6*b^2*c^11*d*f*i^2 - 967680*a^5*b^4*c^10*d*f*i^2 + 483840*a^4*b^5*c^10*d
*e^2*j - 414720*a^4*b^5*c^10*d*g^2*h - 138240*a^4*b^4*c^11*e^2*f*h + 64512*
a^4*b^6*c^9*d*f*i^2 + 39168*a^3*b^8*c^8*d*f*i^2 - 31104*a^3*b^7*c^9*d*g^2*h
 + 13824*a^3*b^6*c^10*e^2*f*h + 10368*a^2*b^9*c^8*d*g^2*h - 9216*a^2*b^10*c
^7*d*f*i^2 + 15630336*a^5*b^2*c^12*d*f^2*h - 14459904*a^4*b^3*c^12*d^2*f*h
+ 9630144*a^3*b^5*c^11*d^2*f*h - 8764416*a^5*b^3*c^11*d*f*h^2 - 3870720*a^5
*b^2*c^12*e*f^2*g - 3193344*a^3*b^5*c^11*d^2*e*i + 2867328*a^4*b^4*c^11*d*f
^2*h - 2095200*a^2*b^7*c^10*d^2*f*h - 1414080*a^3*b^6*c^10*d*f^2*h - 348364
80*a^4*b^2*c^13*d^2*e*g + 1016064*a^2*b^7*c^10*d^2*e*i - 645120*a^4*b^4*c^1
1*e*f^2*g + 306720*a^3*b^7*c^9*d*f*h^2 + 197820*a^2*b^8*c^9*d*f^2*h + 14688
0*a^4*b^5*c^10*d*f*h^2 + 80640*a^3*b^6*c^10*e*f^2*g - 55350*a^2*b^9*c^8*d*f
*h^2 - 2304*a^2*b^8*c^9*e*f^2*g - 3870720*a^5*b^2*c^12*d*f*g^2 - 1935360*a^
4*b^4*c^11*d*f*g^2 - 1658880*a^4*b^3*c^12*d*e^2*h + 725760*a^3*b^6*c^10*d*f
*g^2 + 17418240*a^3*b^4*c^12*d^2*e*g - 124416*a^3*b^5*c^11*d*e^2*h - 96768*
a^2*b^8*c^9*d*f*g^2 + 41472*a^2*b^7*c^10*d*e^2*h - 3919104*a^2*b^6*c^11*d^2
*e*g - 7741440*a^4*b^2*c^13*d*e^2*f + 2903040*a^3*b^4*c^12*d*e^2*f - 387072
*a^2*b^6*c^11*d*e^2*f - 681246720*a^9*b*c^9*d^2*k^2 + 265912320*a^11*b^3*c^
5*e*k^3 + 188743680*a^12*b^2*c^5*g*k^3 - 132956160*a^11*b^4*c^4*g*k^3 - 521
01120*a^13*b*c^5*j^2*k^2 + 25722880*a^12*b^3*c^4*i*k^3 + 19644416*a^11*b^5*
c^3*i*k^3 - 1583680*a^9*b^9*c*j^2*k^2 - 9142272*a^10*b^7*c^2*i*k^3 - 740229
12*a^10*b^5*c^4*e*k^3 - 20643840*a^11*b*c^7*h^2*k^2 + 37011456*a^10*b^6*c^3
*g*k^3 - 2293760*a^9*b^3*c^7*i^3*k - 557056*a^8*b^5*c^6*i^3*k + 147456*a^7*
b^7*c^5*i^3*k - 65536*a^6*b^12*c*i^2*k^2 + 32768*a^6*b^9*c^4*i^3*k - 8192*a
^5*b^11*c^3*i^3*k + 430080*a^10*b*c^8*i^2*j^2 - 2880*a^5*b^13*c*h^2*k^2 + 6
635520*a^7*b^4*c^8*g^3*k - 4792320*a^9*b^8*c^2*g*k^3 - 2211840*a^6*b^6*c^7*
g^3*k + 1359360*a^10*b^2*c^7*h*j^3 + 1173120*a^9*b^4*c^6*h*j^3 + 743040*a^7
*b^4*c^8*h^3*j + 622080*a^8*b^2*c^9*h^3*j + 221184*a^5*b^8*c^6*g^3*k + 1071
36*a^6*b^6*c^7*h^3*j - 32640*a^8*b^6*c^5*h*j^3 - 5796*a^7*b^8*c^4*h*j^3 + 5
40*a^5*b^8*c^6*h^3*j - 270*a^4*b^10*c^5*h^3*j + 210*a^6*b^10*c^3*h*j^3 - 29
49120*a^10*b*c^8*f^2*k^2 + 17694720*a^6*b^3*c^10*e^3*k + 184320*a^8*b*c^10*
h^2*i^2 - 3520*a^3*b^15*c*f^2*k^2 + 9584640*a^9*b^7*c^3*e*k^3 - 2293760*a^9

*b^3*c^7*f*j^3 - 2293760*a^6*b^3*c^10*f^3*j - 1769472*a^5*b^5*c^9*e^3*k - 8
84736*a^6*b^3*c^10*g^3*i - 589824*a^7*b^3*c^9*g*i^3 - 491520*a^8*b^9*c^2*e*
k^3 - 442368*a^5*b^5*c^9*g^3*i - 294912*a^6*b^5*c^8*g*i^3 - 199360*a^8*b^5*
c^6*f*j^3 - 199360*a^5*b^5*c^9*f^3*j + 61920*a^7*b^7*c^5*f*j^3 + 61920*a^4*
b^7*c^8*f^3*j - 49152*a^5*b^7*c^7*g*i^3 - 3682*a^6*b^9*c^4*f*j^3 - 3682*a^3
*b^9*c^7*f^3*j + 70*a^5*b^11*c^3*f*j^3 + 70*a^2*b^11*c^6*f^3*j + 3870720*a^
8*b*c^10*e^2*j^2 + 430080*a^7*b*c^11*f^2*i^2 - 14152320*a^4*b^4*c^11*d^3*j
+ 10644480*a^5*b^2*c^12*d^3*j + 5483520*a^9*b^2*c^8*d*j^3 + 4269888*a^3*b^6
*c^10*d^3*j + 3538944*a^5*b^2*c^12*e^3*i - 1648128*a^5*b^3*c^11*f^3*h + 133
0560*a^8*b^4*c^7*d*j^3 + 1179648*a^7*b^2*c^10*e*i^3 - 898560*a^6*b^3*c^10*f
*h^3 - 826560*a^7*b^6*c^6*d*j^3 - 607068*a^2*b^8*c^9*d^3*j + 589824*a^6*b^4
*c^9*e*i^3 - 354240*a^5*b^5*c^9*f*h^3 - 354240*a^4*b^5*c^10*f^3*h + 145188*
a^6*b^8*c^5*d*j^3 + 98304*a^5*b^6*c^8*e*i^3 + 43680*a^3*b^7*c^9*f^3*h - 216
00*a^4*b^7*c^8*f*h^3 - 9576*a^5*b^10*c^4*d*j^3 + 1350*a^3*b^9*c^7*f*h^3 - 1
050*a^2*b^9*c^8*f^3*h - 504*a*b^14*c^4*d^2*j^2 + 210*a^4*b^12*c^3*d*j^3 + 3
870720*a^6*b*c^12*d^2*i^2 + 1658880*a^6*b*c^12*e^2*h^2 - 9792*a*b^11*c^7*d^
2*i^2 + 16547328*a^4*b^2*c^13*d^3*h - 12306816*a^3*b^4*c^12*d^3*h + 3731097
6*a^3*b^3*c^13*d^3*f + 3037824*a^2*b^6*c^11*d^3*h - 2654208*a^5*b^3*c^11*e*
g^3 + 1949184*a^6*b^2*c^11*d*h^3 + 1296000*a^5*b^4*c^10*d*h^3 - 155520*a^4*
b^6*c^9*d*h^3 - 40500*a*b^10*c^8*d^2*h^2 - 8100*a^3*b^8*c^8*d*h^3 + 4050*a^
2*b^10*c^7*d*h^3 + 3870720*a^5*b*c^13*e^2*f^2 + 34836480*a^4*b*c^14*d^2*e^2
 - 108864*a*b^9*c^9*d^2*g^2 - 8068032*a^2*b^5*c^12*d^3*f - 5623296*a^4*b^3*
c^12*d*f^3 + 1737792*a^3*b^5*c^11*d*f^3 - 260190*a*b^8*c^10*d^2*f^2 - 21168
0*a^2*b^7*c^10*d*f^3 - 435456*a*b^7*c^11*d^2*e^2 - 377487360*a^12*b*c^6*e*k
^3 + 1434977280*a^8*b^3*c^8*d^2*k^2 + 173408256*a^7*c^12*d^2*e*k + 3276800*
a^12*c^7*i*j^2*k - 125829120*a^13*b*c^5*i*k^3 + 26214400*a^12*c^7*f*j*k^2 +
 1179648*a^10*c^9*h^2*i*k + 13440*a^6*b^13*h*j*k^2 + 50331648*a^11*c^8*e*i*
k^2 + 110100480*a^10*c^9*d*f*k^2 + 57802752*a^8*c^11*d^2*i*k + 9830400*a^11
*c^8*e*j^2*k - 3276800*a^9*c^10*f^2*i*k + 4480*a^5*b^14*f*j*k^2 + 15728640*
a^11*c^8*f*h*k^2 - 409600*a^9*c^10*f*i^2*j - 1152*b^16*c^3*d^2*i*k - 122051
6352*a^7*b^5*c^7*d^2*k^2 + 3538944*a^9*c^10*e*h^2*k + 384000*a^8*c^11*f^2*h
*j + 13440*a^4*b^15*d*j*k^2 + 384*a^3*b^16*f*h*k^2 + 20321280*a^7*c^12*d^2*
h*j - 245760*a^8*c^11*f*h*i^2 + 3456*b^15*c^4*d^2*g*k - 270*b^14*c^5*d^2*h*
j - 9830400*a^8*c^11*e*f^2*k + 4838400*a^9*c^10*d*h*j^2 + 2903040*a^8*c^11*
d*h^2*j - 1966080*a^10*b*c^8*i^3*k + 1433600*a^9*b^9*c*i*k^3 + 1152*a^2*b^1
7*d*h*k^2 - 3686400*a^7*c^12*e^2*f*j - 53084160*a^7*b*c^11*e^3*k - 6912*b^1
4*c^5*d^2*e*k - 3456*b^12*c^7*d^2*g*i + 630*b^13*c^6*d^2*f*j + 2688000*a^7*
c^12*d*f^2*j + 245760*a^8*b^10*c*g*k^3 - 2211840*a^6*c^13*e^2*f*h - 1720320
*a^7*c^12*d*f*i^2 - 9450*b^11*c^8*d^2*f*h + 6912*b^11*c^8*d^2*e*i + 1612800
*a^6*c^13*d*f^2*h - 1344000*a^10*b*c^8*f*j^3 - 1344000*a^7*b*c^11*f^3*j - 3
93216*a^8*b*c^10*g*i^3 - 23616*a*b^17*c*d^2*k^2 - 20736*b^10*c^9*d^2*e*g -
75188736*a^4*b*c^14*d^3*f - 883200*a^6*b*c^12*f^3*h - 317952*a^7*b*c^11*f*h
^3 + 43416*a*b^10*c^8*d^3*j - 15482880*a^5*c^14*d*e^2*f - 10616832*a^5*b*c^
13*e^3*g - 345060*a*b^8*c^10*d^3*h - 4262400*a^5*b*c^13*d*f^3 + 852768*a*b^
7*c^11*d^3*f + 7350*a*b^9*c^9*d*f^3 + 584578368*a^6*b^7*c^6*d^2*k^2 + 93905

920*a^12*b^3*c^4*j^2*k^2 - 177997248*a^5*b^9*c^5*d^2*k^2 - 50967040*a^11*b^
5*c^3*j^2*k^2 + 104693760*a^9*b^2*c^8*e^2*k^2 + 12849984*a^10*b^7*c^2*j^2*k
^2 + 20021248*a^11*b^2*c^6*i^2*k^2 - 85524480*a^8*b^4*c^7*e^2*k^2 + 3322368
0*a^10*b^3*c^6*h^2*k^2 + 4227072*a^10*b^4*c^5*i^2*k^2 - 3973120*a^9*b^6*c^4
*i^2*k^2 + 344064*a^7*b^10*c^2*i^2*k^2 - 81920*a^8*b^8*c^3*i^2*k^2 - 113863
68*a^9*b^5*c^5*h^2*k^2 + 26173440*a^9*b^4*c^6*g^2*k^2 - 21381120*a^8*b^6*c^
5*g^2*k^2 + 18874368*a^10*b^2*c^7*g^2*k^2 + 501760*a^9*b^3*c^7*i^2*j^2 + 45
2160*a^8*b^7*c^4*h^2*k^2 + 385920*a^7*b^9*c^3*h^2*k^2 + 170240*a^8*b^5*c^6*
i^2*j^2 - 48960*a^6*b^11*c^2*h^2*k^2 + 9216*a^7*b^7*c^5*i^2*j^2 - 1984*a^6*
b^9*c^4*i^2*j^2 + 64*a^5*b^11*c^3*i^2*j^2 + 5898240*a^7*b^8*c^4*g^2*k^2 + 1
419840*a^8*b^4*c^7*h^2*j^2 + 1387008*a^9*b^2*c^8*h^2*j^2 - 737280*a^6*b^10*
c^3*g^2*k^2 + 84960*a^7*b^6*c^6*h^2*j^2 + 36864*a^5*b^12*c^2*g^2*k^2 - 8010
*a^6*b^8*c^5*h^2*j^2 - 180*a^5*b^10*c^4*h^2*j^2 + 9*a^4*b^12*c^3*h^2*j^2 +
14115840*a^9*b^3*c^7*f^2*k^2 - 9231552*a^7*b^7*c^5*f^2*k^2 + 23592960*a^7*b
^6*c^6*e^2*k^2 + 4984320*a^8*b^5*c^6*f^2*k^2 + 3759040*a^6*b^9*c^4*f^2*k^2
+ 36190080*a^4*b^11*c^4*d^2*k^2 + 967680*a^8*b^3*c^8*g^2*j^2 - 727360*a^5*b
^11*c^3*f^2*k^2 + 276480*a^7*b^3*c^9*h^2*i^2 + 161280*a^7*b^5*c^7*g^2*j^2 +
 140544*a^6*b^5*c^8*h^2*i^2 + 72960*a^4*b^13*c^2*f^2*k^2 + 25344*a^5*b^7*c^
7*h^2*i^2 - 20160*a^6*b^7*c^6*g^2*j^2 + 576*a^5*b^9*c^5*g^2*j^2 + 576*a^4*b
^9*c^6*h^2*i^2 + 3808000*a^8*b^2*c^9*f^2*j^2 - 2949120*a^6*b^8*c^5*e^2*k^2
+ 1643712*a^7*b^4*c^8*f^2*j^2 + 884736*a^7*b^2*c^10*g^2*i^2 + 884736*a^6*b^
4*c^9*g^2*i^2 + 221184*a^5*b^6*c^8*g^2*i^2 + 147456*a^5*b^10*c^4*e^2*k^2 -
125440*a^6*b^6*c^7*f^2*j^2 - 13790*a^5*b^8*c^6*f^2*j^2 + 1785*a^4*b^10*c^5*
f^2*j^2 - 70*a^3*b^12*c^4*f^2*j^2 - 4953600*a^3*b^13*c^3*d^2*k^2 + 18427392
*a^7*b^2*c^10*d^2*j^2 + 645120*a^7*b^3*c^9*e^2*j^2 + 501760*a^6*b^3*c^10*f^
2*i^2 + 442944*a^2*b^15*c^2*d^2*k^2 + 414720*a^6*b^3*c^10*g^2*h^2 + 207360*
a^5*b^5*c^9*g^2*h^2 + 170240*a^5*b^5*c^9*f^2*i^2 - 80640*a^6*b^5*c^8*e^2*j^
2 + 9216*a^4*b^7*c^8*f^2*i^2 + 5184*a^4*b^7*c^8*g^2*h^2 + 2304*a^5*b^7*c^7*
e^2*j^2 - 1984*a^3*b^9*c^7*f^2*i^2 + 64*a^2*b^11*c^6*f^2*i^2 - 4148928*a^6*
b^4*c^9*d^2*j^2 + 3538944*a^6*b^2*c^11*e^2*i^2 + 1684224*a^6*b^2*c^11*f^2*h
^2 + 1264320*a^5*b^4*c^10*f^2*h^2 - 1183392*a^5*b^6*c^8*d^2*j^2 + 884736*a^
5*b^4*c^10*e^2*i^2 + 645750*a^4*b^8*c^7*d^2*j^2 + 126720*a^4*b^6*c^9*f^2*h^
2 - 115920*a^3*b^10*c^6*d^2*j^2 - 13950*a^3*b^8*c^8*f^2*h^2 + 10836*a^2*b^1
2*c^5*d^2*j^2 + 225*a^2*b^10*c^7*f^2*h^2 + 1935360*a^5*b^3*c^11*d^2*i^2 + 9
67680*a^5*b^3*c^11*f^2*g^2 + 829440*a^5*b^3*c^11*e^2*h^2 - 532224*a^4*b^5*c
^10*d^2*i^2 + 161280*a^4*b^5*c^10*f^2*g^2 - 96768*a^3*b^7*c^9*d^2*i^2 + 627
84*a^2*b^9*c^8*d^2*i^2 + 20736*a^4*b^5*c^10*e^2*h^2 - 20160*a^3*b^7*c^9*f^2
*g^2 + 576*a^2*b^9*c^8*f^2*g^2 + 11487744*a^5*b^2*c^12*d^2*h^2 + 7962624*a^
5*b^2*c^12*e^2*g^2 + 35525376*a^4*b^2*c^13*d^2*f^2 - 1412640*a^3*b^6*c^10*d
^2*h^2 + 461376*a^4*b^4*c^11*d^2*h^2 + 375030*a^2*b^8*c^9*d^2*h^2 + 8709120
*a^4*b^3*c^12*d^2*g^2 - 4354560*a^3*b^5*c^11*d^2*g^2 + 979776*a^2*b^7*c^10*
d^2*g^2 + 645120*a^4*b^3*c^12*e^2*f^2 - 80640*a^3*b^5*c^11*e^2*f^2 + 2304*a
^2*b^7*c^10*e^2*f^2 - 15269184*a^3*b^4*c^12*d^2*f^2 + 2870784*a^2*b^6*c^11*
d^2*f^2 - 17418240*a^3*b^3*c^13*d^2*e^2 + 3919104*a^2*b^5*c^12*d^2*e^2 + 38
4*a*b^18*d*f*k^2 - 199229440*a^14*b^2*c^3*k^4 + 8388608*a^12*c^7*i^2*k^2 +

75497472*a^10*c^9*e^2*k^2 + 78400*a^8*b^11*j^2*k^2 + 4096*a^5*b^14*i^2*k^2
+ 345600*a^10*c^9*h^2*j^2 + 576*a^4*b^15*h^2*k^2 + 57937920*a^13*b^4*c^2*k^
4 + 320000*a^9*c^10*f^2*j^2 + 64*a^2*b^17*f^2*k^2 + 16934400*a^8*c^11*d^2*j
^2 + 9*b^16*c^3*d^2*j^2 + 3538944*a^7*c^12*e^2*i^2 + 115200*a^7*c^12*f^2*h^
2 + 576*b^13*c^6*d^2*i^2 + 2025*b^12*c^7*d^2*h^2 + 6096384*a^6*c^13*d^2*h^2
 + 492800*a^11*b^2*c^6*j^4 + 351456*a^10*b^4*c^5*j^4 - 43120*a^9*b^6*c^4*j^
4 + 5184*b^11*c^8*d^2*g^2 + 1225*a^8*b^8*c^3*j^4 + 131072*a^8*b^2*c^9*i^4 +
 98304*a^7*b^4*c^8*i^4 + 32768*a^6*b^6*c^7*i^4 + 11025*b^10*c^9*d^2*f^2 + 4
096*a^5*b^8*c^6*i^4 + 5644800*a^5*c^14*d^2*f^2 + 142560*a^6*b^4*c^9*h^4 + 1
03680*a^7*b^2*c^10*h^4 + 32400*a^5*b^6*c^8*h^4 + 20736*b^9*c^10*d^2*e^2 + 2
025*a^4*b^8*c^7*h^4 + 331776*a^5*b^4*c^10*g^4 + 492800*a^5*b^2*c^12*f^4 + 3
51456*a^4*b^4*c^11*f^4 - 43120*a^3*b^6*c^10*f^4 + 1225*a^2*b^8*c^9*f^4 - 27
433728*a^3*b^2*c^14*d^4 + 6446304*a^2*b^4*c^13*d^4 + a^2*b^14*c^3*f^2*j^2 -
 81920*a^8*b^11*i*k^3 + 384000*a^11*c^8*h*j^3 + 138240*a^9*c^10*h^3*j + 474
16320*a^6*c^13*d^3*j - 1134*b^12*c^7*d^3*j + 7077888*a^6*c^13*e^3*i + 26880
00*a^10*c^9*d*j^3 + 786432*a^8*c^11*e*i^3 + 28449792*a^5*c^14*d^3*h - 77824
00*a^12*b^6*c*k^4 + 17010*b^10*c^9*d^3*h + 580608*a^7*c^12*d*h^3 - 39690*b^
9*c^10*d^3*f - 734832*a*b^6*c^12*d^4 + 268435456*a^15*c^4*k^4 + 576*b^19*d^
2*k^2 + 409600*a^11*b^8*k^4 + 160000*a^12*c^7*j^4 + 65536*a^9*c^10*i^4 + 20
736*a^8*c^11*h^4 + 49787136*a^4*c^15*d^4 + 160000*a^6*c^13*f^4 + 5308416*a^
5*c^14*e^4 + 35721*b^8*c^11*d^4, z, n)*((768*a^2*b^14*c^6*d - 3145728*a^10*
c^12*h - 5242880*a^11*c^11*j - 22020096*a^9*c^13*d - 22272*a^3*b^12*c^7*d +
 282624*a^4*b^10*c^8*d - 2027520*a^5*b^8*c^9*d + 8847360*a^6*b^6*c^10*d - 2
3396352*a^7*b^4*c^11*d + 34603008*a^8*b^2*c^12*d + 256*a^3*b^13*c^6*f - 921
6*a^4*b^11*c^7*f + 122880*a^5*b^9*c^8*f - 819200*a^6*b^7*c^9*f + 2949120*a^
7*b^5*c^10*f - 5505024*a^8*b^3*c^11*f + 768*a^4*b^12*c^6*h - 12288*a^5*b^10
*c^7*h + 61440*a^6*b^8*c^8*h - 983040*a^8*b^4*c^10*h + 3145728*a^9*b^2*c^11
*h + 256*a^5*b^12*c^5*j - 61440*a^7*b^8*c^7*j + 655360*a^8*b^6*c^8*j - 2949
120*a^9*b^4*c^9*j + 6291456*a^10*b^2*c^10*j + 4194304*a^9*b*c^12*f)/(512*(4
096*a^10*c^10 + a^4*b^12*c^4 - 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7
*b^6*c^7 + 3840*a^8*b^4*c^8 - 6144*a^9*b^2*c^9)) + (x*(1572864*a^9*c^13*e +
 524288*a^10*c^12*i - 1536*a^4*b^10*c^8*e + 30720*a^5*b^8*c^9*e - 245760*a^
6*b^6*c^10*e + 983040*a^7*b^4*c^11*e - 1966080*a^8*b^2*c^12*e + 768*a^4*b^1
1*c^7*g - 15360*a^5*b^9*c^8*g + 122880*a^6*b^7*c^9*g - 491520*a^7*b^5*c^10*
g + 983040*a^8*b^3*c^11*g - 256*a^4*b^12*c^6*i + 4608*a^5*b^10*c^7*i - 3072
0*a^6*b^8*c^8*i + 81920*a^7*b^6*c^9*i - 393216*a^9*b^2*c^11*i + 512*a^4*b^1
5*c^3*k - 14592*a^5*b^13*c^4*k + 178944*a^6*b^11*c^5*k - 1223680*a^7*b^9*c^
6*k + 5038080*a^8*b^7*c^7*k - 12484608*a^9*b^5*c^8*k + 17235968*a^10*b^3*c^
9*k - 786432*a^9*b*c^12*g - 10223616*a^11*b*c^10*k))/(64*(4096*a^10*c^10 +
a^4*b^12*c^4 - 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7*b^6*c^7 + 3840*
a^8*b^4*c^8 - 6144*a^9*b^2*c^9)) + (root(56371445760*a^11*b^8*c^12*z^4 - 50
3316480*a^8*b^14*c^9*z^4 + 47185920*a^7*b^16*c^8*z^4 - 2621440*a^6*b^18*c^7
*z^4 + 65536*a^5*b^20*c^6*z^4 - 171798691840*a^14*b^2*c^15*z^4 + 1932735283
20*a^13*b^4*c^14*z^4 - 128849018880*a^12*b^6*c^13*z^4 - 16911433728*a^10*b^
10*c^11*z^4 + 3523215360*a^9*b^12*c^10*z^4 + 68719476736*a^15*c^16*z^4 - 47

185920*a^7*b^16*c^5*k*z^3 + 2621440*a^6*b^18*c^4*k*z^3 - 65536*a^5*b^20*c^3
*k*z^3 + 171798691840*a^14*b^2*c^12*k*z^3 - 193273528320*a^13*b^4*c^11*k*z^
3 + 128849018880*a^12*b^6*c^10*k*z^3 + 16911433728*a^10*b^10*c^8*k*z^3 - 35
23215360*a^9*b^12*c^7*k*z^3 - 56371445760*a^11*b^8*c^9*k*z^3 + 503316480*a^
8*b^14*c^6*k*z^3 - 68719476736*a^15*c^13*k*z^3 + 1536*a*b^18*c^6*d*f*z^2 -
2571632640*a^9*b^5*c^11*d*j*z^2 + 2548039680*a^9*b^3*c^13*d*h*z^2 + 2453667
840*a^9*b^7*c^9*e*k*z^2 + 2181038080*a^12*b^3*c^10*i*k*z^2 - 6492782592*a^1
0*b^5*c^10*e*k*z^2 + 1509949440*a^9*b^3*c^13*e*g*z^2 - 1401421824*a^8*b^5*c
^12*d*h*z^2 - 1226833920*a^9*b^8*c^8*g*k*z^2 - 1321205760*a^9*b^2*c^14*d*f*
z^2 - 2793406464*a^11*b*c^13*d*j*z^2 + 9563013120*a^11*b^3*c^11*e*k*z^2 + 8
90634240*a^8*b^7*c^10*d*j*z^2 - 754974720*a^8*b^5*c^12*e*g*z^2 - 570425344*
a^11*b^5*c^9*i*k*z^2 + 732168192*a^7*b^6*c^12*d*f*z^2 - 581959680*a^10*b^4*
c^11*f*j*z^2 - 603979776*a^10*b^2*c^13*e*i*z^2 + 534773760*a^11*b^3*c^11*h*
j*z^2 - 558366720*a^8*b^9*c^8*e*k*z^2 - 4781506560*a^11*b^4*c^10*g*k*z^2 -
2013265920*a^13*b*c^11*i*k*z^2 - 456130560*a^9*b^4*c^12*f*h*z^2 + 384040960
*a^9*b^6*c^10*f*j*z^2 - 264241152*a^10*b^7*c^8*i*k*z^2 + 390463488*a^7*b^7*
c^11*d*h*z^2 + 279183360*a^8*b^10*c^7*g*k*z^2 + 301989888*a^10*b^3*c^12*g*i
*z^2 + 222822400*a^9*b^9*c^7*i*k*z^2 - 366280704*a^6*b^8*c^11*d*f*z^2 - 330
301440*a^8*b^4*c^13*d*f*z^2 + 254017536*a^8*b^6*c^11*f*h*z^2 - 1887436800*a
^10*b*c^14*d*h*z^2 + 188743680*a^10*b^2*c^13*f*h*z^2 - 185303040*a^7*b^9*c^
9*d*j*z^2 - 117964800*a^10*b^5*c^10*h*j*z^2 - 6039797760*a^12*b*c^12*e*k*z^
2 - 67502080*a^8*b^11*c^6*i*k*z^2 + 121634816*a^11*b^2*c^12*f*j*z^2 + 18874
3680*a^7*b^7*c^11*e*g*z^2 - 115671040*a^8*b^8*c^9*f*j*z^2 + 125829120*a^8*b
^6*c^11*e*i*z^2 + 10813440*a^7*b^13*c^5*i*k*z^2 + 76677120*a^7*b^11*c^7*e*k
*z^2 - 38338560*a^7*b^12*c^6*g*k*z^2 - 37355520*a^9*b^7*c^9*h*j*z^2 - 91750
4*a^6*b^15*c^4*i*k*z^2 + 32768*a^5*b^17*c^3*i*k*z^2 - 62914560*a^8*b^7*c^10
*g*i*z^2 + 23101440*a^8*b^9*c^8*h*j*z^2 - 4349952*a^7*b^11*c^7*h*j*z^2 + 29
49120*a^6*b^14*c^5*g*k*z^2 + 337920*a^6*b^13*c^6*h*j*z^2 - 98304*a^5*b^16*c
^4*g*k*z^2 - 7680*a^5*b^15*c^5*h*j*z^2 - 61931520*a^7*b^8*c^10*f*h*z^2 + 23
592960*a^7*b^9*c^9*g*i*z^2 + 17940480*a^7*b^10*c^8*f*j*z^2 - 47185920*a^7*b
^8*c^10*e*i*z^2 - 5898240*a^6*b^13*c^6*e*k*z^2 - 3538944*a^6*b^11*c^8*g*i*z
^2 - 1347584*a^6*b^12*c^7*f*j*z^2 + 196608*a^5*b^15*c^5*e*k*z^2 + 196608*a^
5*b^13*c^7*g*i*z^2 + 35840*a^5*b^14*c^6*f*j*z^2 + 96583680*a^5*b^10*c^10*d*
f*z^2 + 23371776*a^6*b^11*c^8*d*j*z^2 - 51609600*a^6*b^9*c^10*d*h*z^2 + 707
7888*a^6*b^10*c^9*e*i*z^2 + 6144000*a^6*b^10*c^9*f*h*z^2 - 1677312*a^5*b^13
*c^7*d*j*z^2 - 393216*a^5*b^12*c^8*e*i*z^2 + 61440*a^5*b^12*c^8*f*h*z^2 + 5
3760*a^4*b^15*c^6*d*j*z^2 - 46080*a^4*b^14*c^7*f*h*z^2 + 1536*a^3*b^16*c^6*
f*h*z^2 - 23592960*a^6*b^9*c^10*e*g*z^2 + 1179648*a^5*b^11*c^9*e*g*z^2 + 82
9440*a^4*b^13*c^8*d*h*z^2 + 368640*a^5*b^11*c^9*d*h*z^2 - 105984*a^3*b^15*c
^7*d*h*z^2 + 4608*a^2*b^17*c^6*d*h*z^2 - 15175680*a^4*b^12*c^9*d*f*z^2 + 14
28480*a^3*b^14*c^8*d*f*z^2 - 73728*a^2*b^16*c^7*d*f*z^2 + 4108320768*a^10*b
^3*c^12*d*j*z^2 - 1207959552*a^10*b*c^14*e*g*z^2 - 578813952*a^12*b*c^12*h*
j*z^2 + 3246391296*a^10*b^6*c^9*g*k*z^2 - 402653184*a^11*b*c^13*g*i*z^2 + 3
019898880*a^12*b^2*c^11*g*k*z^2 - 440401920*a^10*b*c^14*f^2*z^2 - 188743680
*a^11*b*c^13*h^2*z^2 + 1761607680*a^10*c^15*d*f*z^2 - 655360*a^6*b^18*c*k^2

*z^2 - 94464*a*b^17*c^7*d^2*z^2 + 6936330240*a^8*b^3*c^14*d^2*z^2 + 2464874
496*a^6*b^7*c^12*d^2*z^2 - 3963617280*a^9*b*c^15*d^2*z^2 + 58007224320*a^13
*b^4*c^8*k^2*z^2 + 14968422400*a^11*b^8*c^6*k^2*z^2 + 805306368*a^11*c^14*e
*i*z^2 - 35966156800*a^12*b^6*c^7*k^2*z^2 + 419430400*a^12*c^13*f*j*z^2 - 1
509949440*a^9*b^2*c^14*e^2*z^2 + 251658240*a^11*c^14*f*h*z^2 - 56874762240*
a^14*b^2*c^9*k^2*z^2 - 5400428544*a^7*b^5*c^13*d^2*z^2 + 890470400*a^9*b^12
*c^4*k^2*z^2 + 754974720*a^8*b^4*c^13*e^2*z^2 - 730054656*a^5*b^9*c^11*d^2*
z^2 + 477102080*a^12*b^3*c^10*j^2*z^2 + 477102080*a^9*b^3*c^13*f^2*z^2 - 37
7487360*a^9*b^4*c^12*g^2*z^2 + 301989888*a^10*b^2*c^13*g^2*z^2 - 174325760*
a^11*b^5*c^9*j^2*z^2 - 126156800*a^8*b^14*c^3*k^2*z^2 + 188743680*a^8*b^6*c
^11*g^2*z^2 + 141557760*a^10*b^3*c^12*h^2*z^2 - 174325760*a^8*b^5*c^12*f^2*
z^2 - 188743680*a^7*b^6*c^12*e^2*z^2 - 4350935040*a^10*b^10*c^5*k^2*z^2 + 1
46165760*a^4*b^11*c^10*d^2*z^2 - 50331648*a^10*b^4*c^11*i^2*z^2 + 11796480*
a^7*b^16*c^2*k^2*z^2 - 33554432*a^11*b^2*c^12*i^2*z^2 + 11206656*a^10*b^7*c
^8*j^2*z^2 + 8929280*a^9*b^9*c^7*j^2*z^2 + 20971520*a^9*b^6*c^10*i^2*z^2 -
2600960*a^8*b^11*c^6*j^2*z^2 + 291840*a^7*b^13*c^5*j^2*z^2 - 14080*a^6*b^15
*c^4*j^2*z^2 + 256*a^5*b^17*c^3*j^2*z^2 - 47185920*a^7*b^8*c^10*g^2*z^2 - 2
6542080*a^8*b^7*c^10*h^2*z^2 - 2752512*a^7*b^10*c^8*i^2*z^2 + 2621440*a^8*b
^8*c^9*i^2*z^2 + 524288*a^6*b^12*c^7*i^2*z^2 - 32768*a^5*b^14*c^6*i^2*z^2 +
9584640*a^7*b^9*c^9*h^2*z^2 - 2359296*a^9*b^5*c^11*h^2*z^2 - 1290240*a^6*b
^11*c^8*h^2*z^2 + 46080*a^5*b^13*c^7*h^2*z^2 + 2304*a^4*b^15*c^6*h^2*z^2 +
5898240*a^6*b^10*c^9*g^2*z^2 - 294912*a^5*b^12*c^8*g^2*z^2 + 11206656*a^7*b
^7*c^11*f^2*z^2 + 8929280*a^6*b^9*c^10*f^2*z^2 + 23592960*a^6*b^8*c^11*e^2*
z^2 - 2600960*a^5*b^11*c^9*f^2*z^2 + 291840*a^4*b^13*c^8*f^2*z^2 - 14080*a^
3*b^15*c^7*f^2*z^2 + 256*a^2*b^17*c^6*f^2*z^2 - 19860480*a^3*b^13*c^9*d^2*z
^2 - 1179648*a^5*b^10*c^10*e^2*z^2 + 1771776*a^2*b^15*c^8*d^2*z^2 - 4404019
20*a^13*b*c^11*j^2*z^2 + 1207959552*a^10*c^15*e^2*z^2 + 134217728*a^12*c^13
*i^2*z^2 + 25769803776*a^15*c^10*k^2*z^2 + 16384*a^5*b^20*k^2*z^2 + 2304*b^
19*c^6*d^2*z^2 + 165150720*a^9*b*c^12*d*g*j*z + 23592960*a^10*b*c^11*g*h*j*
z + 169869312*a^7*b*c^14*d*e*f*z + 99090432*a^8*b*c^13*d*g*h*z - 3145728*a^
9*b*c^12*f*h*i*z + 56623104*a^8*b*c^13*d*f*i*z - 1536*a*b^18*c^3*d*f*k*z -
9437184*a^8*b*c^13*e*f*h*z + 1536*a*b^15*c^6*d*f*i*z - 4608*a*b^14*c^7*d*f*
g*z + 9216*a*b^13*c^8*d*e*f*z + 2173501440*a^9*b^5*c^8*d*j*k*z - 1987706880
*a^9*b^3*c^10*d*h*k*z + 1121255424*a^8*b^5*c^9*d*h*k*z + 861143040*a^8*b^4*
c^10*d*f*k*z - 859963392*a^7*b^6*c^9*d*f*k*z - 780779520*a^8*b^7*c^7*d*j*k*
z - 754974720*a^9*b^3*c^10*e*g*k*z + 2222456832*a^11*b*c^10*d*j*k*z - 45416
4480*a^11*b^3*c^8*h*j*k*z + 377487360*a^8*b^5*c^9*e*g*k*z + 290979840*a^10*
b^4*c^8*f*j*k*z + 381026304*a^6*b^8*c^8*d*f*k*z + 412876800*a^8*b^2*c^12*d*
e*j*z + 301989888*a^10*b^2*c^10*e*i*k*z - 320421888*a^7*b^7*c^8*d*h*k*z + 1
85794560*a^10*b^5*c^7*h*j*k*z - 192020480*a^9*b^6*c^7*f*j*k*z + 190709760*a
^9*b^4*c^9*f*h*k*z - 150994944*a^10*b^3*c^9*g*i*k*z + 168990720*a^7*b^9*c^6
*d*j*k*z + 235929600*a^9*b^2*c^11*d*f*k*z - 206438400*a^8*b^3*c^11*d*g*j*z
- 206438400*a^7*b^4*c^11*d*e*j*z - 101646336*a^8*b^6*c^8*f*h*k*z - 29245440
*a^9*b^7*c^6*h*j*k*z - 60817408*a^11*b^2*c^9*f*j*k*z + 57835520*a^8*b^8*c^6
*f*j*k*z + 219414528*a^7*b^2*c^13*d*e*h*z - 70778880*a^10*b^2*c^10*f*h*k*z

+ 677376*a^7*b^11*c^4*h*j*k*z - 645120*a^8*b^9*c^5*h*j*k*z - 53760*a^6*b^13
*c^3*h*j*k*z + 31457280*a^8*b^7*c^7*g*i*k*z - 62914560*a^8*b^6*c^8*e*i*k*z
- 94371840*a^7*b^7*c^8*e*g*k*z - 221773824*a^6*b^3*c^13*d*e*f*z + 82575360*
a^9*b^2*c^11*d*i*j*z + 11796480*a^10*b^2*c^10*h*i*j*z - 11796480*a^7*b^9*c^
6*g*i*k*z - 8970240*a^7*b^10*c^5*f*j*k*z + 103219200*a^7*b^5*c^10*d*g*j*z -
2457600*a^8*b^6*c^8*h*i*j*z + 1769472*a^6*b^11*c^5*g*i*k*z + 921600*a^7*b^
8*c^7*h*i*j*z + 673792*a^6*b^12*c^4*f*j*k*z - 138240*a^6*b^10*c^6*h*i*j*z -
98304*a^5*b^13*c^4*g*i*k*z - 17920*a^5*b^14*c^3*f*j*k*z + 7680*a^5*b^12*c^
5*h*i*j*z - 97136640*a^5*b^10*c^7*d*f*k*z - 29491200*a^9*b^3*c^10*g*h*j*z +
58982400*a^9*b^2*c^11*e*h*j*z + 23592960*a^7*b^8*c^7*e*i*k*z - 22169088*a^
6*b^11*c^5*d*j*k*z + 21381120*a^7*b^8*c^7*f*h*k*z + 14745600*a^8*b^5*c^9*g*
h*j*z + 42854400*a^6*b^9*c^7*d*h*k*z - 109707264*a^7*b^3*c^12*d*g*h*z - 368
6400*a^7*b^7*c^8*g*h*j*z - 3538944*a^6*b^10*c^6*e*i*k*z + 1645056*a^5*b^13*
c^4*d*j*k*z - 890880*a^6*b^10*c^6*f*h*k*z + 460800*a^6*b^9*c^7*g*h*j*z - 33
0240*a^5*b^12*c^5*f*h*k*z + 196608*a^5*b^12*c^5*e*i*k*z - 53760*a^4*b^15*c^
3*d*j*k*z + 46080*a^4*b^14*c^4*f*h*k*z - 23040*a^5*b^11*c^6*g*h*j*z - 1536*
a^3*b^16*c^3*f*h*k*z - 29491200*a^8*b^4*c^10*e*h*j*z - 17203200*a^7*b^6*c^9
*d*i*j*z + 11796480*a^6*b^9*c^7*e*g*k*z + 110886912*a^6*b^4*c^12*d*f*g*z +
7372800*a^7*b^6*c^9*e*h*j*z + 40108032*a^8*b^2*c^12*d*h*i*z + 6451200*a^6*b
^8*c^8*d*i*j*z + 2359296*a^8*b^3*c^11*f*h*i*z - 967680*a^5*b^10*c^7*d*i*j*z
- 921600*a^6*b^8*c^8*e*h*j*z - 829440*a^4*b^13*c^5*d*h*k*z - 589824*a^5*b^
11*c^6*e*g*k*z - 491520*a^6*b^7*c^9*f*h*i*z + 184320*a^5*b^9*c^8*f*h*i*z +
105984*a^3*b^15*c^4*d*h*k*z + 69120*a^5*b^11*c^6*d*h*k*z + 53760*a^4*b^12*c
^6*d*i*j*z + 46080*a^5*b^10*c^7*e*h*j*z - 27648*a^4*b^11*c^7*f*h*i*z - 4608
*a^2*b^17*c^3*d*h*k*z + 1536*a^3*b^13*c^6*f*h*i*z - 25804800*a^6*b^7*c^9*d*
g*j*z - 88473600*a^6*b^4*c^12*d*e*h*z + 51609600*a^6*b^6*c^10*d*e*j*z - 849
34656*a^7*b^2*c^13*d*f*g*z + 117964800*a^5*b^5*c^12*d*e*f*z + 15160320*a^4*
b^12*c^6*d*f*k*z - 45613056*a^7*b^3*c^12*d*f*i*z + 44236800*a^6*b^5*c^11*d*
g*h*z - 10321920*a^6*b^6*c^10*d*h*i*z + 7077888*a^7*b^4*c^11*d*h*i*z - 5898
240*a^7*b^4*c^11*f*g*h*z + 4718592*a^8*b^2*c^12*f*g*h*z + 3225600*a^5*b^9*c
^8*d*g*j*z + 2949120*a^6*b^6*c^10*f*g*h*z + 2396160*a^5*b^8*c^9*d*h*i*z - 1
428480*a^3*b^14*c^5*d*f*k*z - 737280*a^5*b^8*c^9*f*g*h*z - 161280*a^4*b^11*
c^7*d*g*j*z + 92160*a^4*b^10*c^8*f*g*h*z + 73728*a^2*b^16*c^4*d*f*k*z - 506
88*a^3*b^12*c^7*d*h*i*z - 27648*a^4*b^10*c^8*d*h*i*z - 4608*a^3*b^12*c^7*f*
g*h*z + 4608*a^2*b^14*c^6*d*h*i*z - 58982400*a^5*b^6*c^11*d*f*g*z + 1179648
0*a^7*b^3*c^12*e*f*h*z + 8847360*a^5*b^7*c^10*d*f*i*z - 6635520*a^5*b^7*c^1
0*d*g*h*z - 6451200*a^5*b^8*c^9*d*e*j*z - 5898240*a^6*b^5*c^11*e*f*h*z - 38
09280*a^4*b^9*c^9*d*f*i*z + 2359296*a^6*b^5*c^11*d*f*i*z + 1474560*a^5*b^7*
c^10*e*f*h*z + 681984*a^3*b^11*c^8*d*f*i*z + 322560*a^4*b^10*c^8*d*e*j*z -
276480*a^4*b^9*c^9*d*g*h*z - 184320*a^4*b^9*c^9*e*f*h*z + 179712*a^3*b^11*c
^8*d*g*h*z - 55296*a^2*b^13*c^7*d*f*i*z - 13824*a^2*b^13*c^7*d*g*h*z + 9216
*a^3*b^11*c^8*e*f*h*z + 16220160*a^4*b^8*c^10*d*f*g*z + 13271040*a^5*b^6*c^
11*d*e*h*z - 2396160*a^3*b^10*c^9*d*f*g*z + 552960*a^4*b^8*c^10*d*e*h*z - 3
59424*a^3*b^10*c^9*d*e*h*z + 175104*a^2*b^12*c^8*d*f*g*z + 27648*a^2*b^12*c
^8*d*e*h*z - 32440320*a^4*b^7*c^11*d*e*f*z + 4792320*a^3*b^9*c^10*d*e*f*z -

350208*a^2*b^11*c^9*d*e*f*z + 1439170560*a^10*b*c^11*d*h*k*z - 3361603584*
a^10*b^3*c^9*d*j*k*z + 603979776*a^10*b*c^11*e*g*k*z + 407371776*a^12*b*c^9
*h*j*k*z + 201326592*a^11*b*c^10*g*i*k*z + 346816512*a^7*b*c^14*d^2*g*z + 1
29761280*a^11*b*c^10*h^2*k*z + 121896960*a^10*b*c^11*f^2*k*z + 458752*a^6*b
^15*c*i*k^2*z + 19660800*a^11*b*c^10*g*j^2*z + 49152*a^5*b^16*c*g*k^2*z + 7
077888*a^9*b*c^12*g*h^2*z + 94464*a*b^17*c^4*d^2*k*z - 19660800*a^8*b*c^13*
f^2*g*z - 66816*a*b^14*c^7*d^2*i*z + 214272*a*b^13*c^8*d^2*g*z - 428544*a*b
^12*c^9*d^2*e*z + 2390753280*a^11*b^4*c^7*g*k^2*z - 2411421696*a^6*b^7*c^9*
d^2*k*z - 6603079680*a^8*b^3*c^11*d^2*k*z + 3715891200*a^9*b*c^12*d^2*k*z -
880803840*a^10*c^12*d*f*k*z - 1623195648*a^10*b^6*c^6*g*k^2*z - 402653184*
a^11*c^11*e*i*k*z - 1509949440*a^12*b^2*c^8*g*k^2*z - 209715200*a^12*c^10*f
*j*k*z - 330301440*a^9*c^13*d*e*j*z + 3019898880*a^12*b*c^9*e*k^2*z - 12582
9120*a^11*c^11*f*h*k*z - 110100480*a^10*c^12*d*i*j*z - 198180864*a^8*c^14*d
*e*h*z - 15728640*a^11*c^11*h*i*j*z - 1226833920*a^9*b^7*c^6*e*k^2*z - 4718
5920*a^10*c^12*e*h*j*z - 66060288*a^9*c^13*d*h*i*z - 1090519040*a^12*b^3*c^
7*i*k^2*z + 1022754816*a^6*b^2*c^14*d^2*e*z + 5216108544*a^7*b^5*c^10*d^2*k
*z + 754974720*a^9*b^2*c^11*e^2*k*z + 721529856*a^5*b^9*c^8*d^2*k*z + 61341
6960*a^9*b^8*c^5*g*k^2*z - 642318336*a^5*b^4*c^13*d^2*e*z - 4781506560*a^11
*b^3*c^8*e*k^2*z - 398131200*a^12*b^3*c^7*j^2*k*z - 511377408*a^6*b^3*c^13*
d^2*g*z - 377487360*a^8*b^4*c^10*e^2*k*z + 285212672*a^11*b^5*c^6*i*k^2*z +
199065600*a^11*b^5*c^6*j^2*k*z + 279183360*a^8*b^9*c^5*e*k^2*z + 321159168
*a^5*b^5*c^12*d^2*g*z + 188743680*a^9*b^4*c^9*g^2*k*z + 132120576*a^10*b^7*
c^5*i*k^2*z - 150994944*a^10*b^2*c^10*g^2*k*z - 111411200*a^9*b^9*c^4*i*k^2
*z - 126812160*a^10*b^3*c^9*h^2*k*z + 225312768*a^7*b^2*c^13*d^2*i*z - 1395
91680*a^8*b^10*c^4*g*k^2*z - 49766400*a^10*b^7*c^5*j^2*k*z - 145463040*a^4*
b^11*c^7*d^2*k*z - 94371840*a^8*b^6*c^8*g^2*k*z + 223395840*a^4*b^6*c^12*d^
2*e*z + 33751040*a^8*b^11*c^3*i*k^2*z - 78970880*a^9*b^3*c^10*f^2*k*z + 943
71840*a^7*b^6*c^9*e^2*k*z + 25165824*a^10*b^4*c^8*i^2*k*z + 6220800*a^9*b^9
*c^4*j^2*k*z + 39223296*a^9*b^5*c^8*h^2*k*z - 311040*a^8*b^11*c^3*j^2*k*z +
16777216*a^11*b^2*c^9*i^2*k*z - 10485760*a^9*b^6*c^7*i^2*k*z - 5406720*a^7
*b^13*c^2*i*k^2*z + 1376256*a^7*b^10*c^5*i^2*k*z - 1310720*a^8*b^8*c^6*i^2*
k*z - 262144*a^6*b^12*c^4*i^2*k*z + 16384*a^5*b^14*c^3*i^2*k*z + 10354688*a
^11*b^2*c^9*i*j^2*z + 23592960*a^7*b^8*c^7*g^2*k*z + 38559744*a^7*b^7*c^8*f
^2*k*z + 19169280*a^7*b^12*c^3*g*k^2*z - 2048000*a^9*b^6*c^7*i*j^2*z - 1520
640*a^7*b^9*c^6*h^2*k*z - 1105920*a^8*b^7*c^7*h^2*k*z + 849920*a^8*b^8*c^6*
i*j^2*z - 393216*a^10*b^4*c^8*i*j^2*z + 195840*a^6*b^11*c^5*h^2*k*z - 14592
0*a^7*b^10*c^5*i*j^2*z + 11520*a^5*b^13*c^4*h^2*k*z + 11008*a^6*b^12*c^4*i*
j^2*z - 2304*a^4*b^15*c^3*h^2*k*z - 256*a^5*b^14*c^3*i*j^2*z - 25362432*a^1
0*b^3*c^9*g*j^2*z - 24739840*a^8*b^5*c^9*f^2*k*z - 38338560*a^7*b^11*c^4*e*
k^2*z - 2949120*a^6*b^10*c^6*g^2*k*z - 1474560*a^6*b^14*c^2*g*k^2*z + 50724
864*a^10*b^2*c^10*e*j^2*z + 147456*a^5*b^12*c^5*g^2*k*z - 15150080*a^6*b^9*
c^7*f^2*k*z + 13271040*a^9*b^5*c^8*g*j^2*z - 111697920*a^4*b^7*c^11*d^2*g*z
- 3563520*a^8*b^7*c^7*g*j^2*z + 3538944*a^9*b^2*c^11*h^2*i*z + 2912000*a^5
*b^11*c^6*f^2*k*z - 737280*a^7*b^6*c^9*h^2*i*z + 506880*a^7*b^9*c^6*g*j^2*z
- 291840*a^4*b^13*c^5*f^2*k*z + 276480*a^6*b^8*c^8*h^2*i*z - 41472*a^5*b^1

0*c^7*h^2*i*z - 34560*a^6*b^11*c^5*g*j^2*z + 14080*a^3*b^15*c^4*f^2*k*z + 2
304*a^4*b^12*c^6*h^2*i*z + 768*a^5*b^13*c^4*g*j^2*z - 256*a^2*b^17*c^3*f^2*
k*z - 11796480*a^6*b^8*c^8*e^2*k*z - 26542080*a^9*b^4*c^9*e*j^2*z + 1983744
0*a^3*b^13*c^6*d^2*k*z + 2949120*a^6*b^13*c^3*e*k^2*z + 589824*a^5*b^10*c^7
*e^2*k*z - 98304*a^5*b^15*c^2*e*k^2*z - 10354688*a^8*b^2*c^12*f^2*i*z - 436
46976*a^6*b^4*c^12*d^2*i*z - 8847360*a^8*b^3*c^11*g*h^2*z + 7127040*a^8*b^6
*c^8*e*j^2*z + 4423680*a^7*b^5*c^10*g*h^2*z + 2048000*a^6*b^6*c^10*f^2*i*z
- 1771776*a^2*b^15*c^5*d^2*k*z - 1105920*a^6*b^7*c^9*g*h^2*z - 1013760*a^7*
b^8*c^7*e*j^2*z - 849920*a^5*b^8*c^9*f^2*i*z + 393216*a^7*b^4*c^11*f^2*i*z
+ 145920*a^4*b^10*c^8*f^2*i*z + 138240*a^5*b^9*c^8*g*h^2*z + 69120*a^6*b^10
*c^6*e*j^2*z - 11008*a^3*b^12*c^7*f^2*i*z - 6912*a^4*b^11*c^7*g*h^2*z - 153
6*a^5*b^12*c^5*e*j^2*z + 256*a^2*b^14*c^6*f^2*i*z - 32587776*a^5*b^6*c^11*d
^2*i*z + 25362432*a^7*b^3*c^12*f^2*g*z + 21657600*a^4*b^8*c^10*d^2*i*z + 17
694720*a^8*b^2*c^12*e*h^2*z - 50724864*a^7*b^2*c^13*e*f^2*z - 13271040*a^6*
b^5*c^11*f^2*g*z - 8847360*a^7*b^4*c^11*e*h^2*z - 5810688*a^3*b^10*c^9*d^2*
i*z + 3563520*a^5*b^7*c^10*f^2*g*z + 2211840*a^6*b^6*c^10*e*h^2*z + 845568*
a^2*b^12*c^8*d^2*i*z - 506880*a^4*b^9*c^9*f^2*g*z - 276480*a^5*b^8*c^9*e*h^
2*z + 34560*a^3*b^11*c^8*f^2*g*z + 13824*a^4*b^10*c^8*e*h^2*z - 768*a^2*b^1
3*c^7*f^2*g*z + 26542080*a^6*b^4*c^12*e*f^2*z + 23362560*a^3*b^9*c^10*d^2*g
*z - 46725120*a^3*b^8*c^11*d^2*e*z - 7127040*a^5*b^6*c^11*e*f^2*z - 2965248
*a^2*b^11*c^9*d^2*g*z + 1013760*a^4*b^8*c^10*e*f^2*z - 69120*a^3*b^10*c^9*e
*f^2*z + 1536*a^2*b^12*c^8*e*f^2*z + 5930496*a^2*b^10*c^10*d^2*e*z + 100663
2960*a^13*b*c^8*i*k^2*z + 3246391296*a^10*b^5*c^7*e*k^2*z + 318504960*a^13*
b*c^8*j^2*k*z + 61538304*a^10*b^10*c^2*k^3*z - 603979776*a^10*c^12*e^2*k*z
- 693633024*a^7*c^15*d^2*e*z - 231211008*a^8*c^14*d^2*i*z - 67108864*a^12*c
^10*i^2*k*z - 13107200*a^12*c^10*i*j^2*z - 16384*a^5*b^17*i*k^2*z - 3932160
0*a^11*c^11*e*j^2*z - 4718592*a^10*c^12*h^2*i*z - 2304*b^19*c^3*d^2*k*z + 1
3107200*a^9*c^13*f^2*i*z + 2304*b^16*c^6*d^2*i*z - 14155776*a^9*c^13*e*h^2*
z + 39321600*a^8*c^14*e*f^2*z - 4833280*a^9*b^12*c*k^3*z - 6912*b^15*c^7*d^
2*g*z + 6962544640*a^14*b^2*c^6*k^3*z + 13824*b^14*c^8*d^2*e*z + 1876951040
*a^12*b^6*c^4*k^3*z - 4844421120*a^13*b^4*c^5*k^3*z - 437780480*a^11*b^8*c^
3*k^3*z - 4294967296*a^15*c^7*k^3*z + 163840*a^8*b^14*k^3*z + 6144000*a^10*
b*c^8*f*i*j*k - 5898240*a^10*b*c^8*g*h*j*k - 41287680*a^9*b*c^9*d*g*j*k + 4
472832*a^9*b*c^9*f*h*i*k + 18432000*a^9*b*c^9*e*f*j*k + 3391488*a^8*b*c^10*
e*h*i*j + 1228800*a^8*b*c^10*f*g*i*j - 24772608*a^8*b*c^10*d*g*h*k + 134184
96*a^8*b*c^10*e*f*h*k + 11649024*a^8*b*c^10*d*f*i*k + 737280*a^7*b*c^11*f*g
*h*i - 768*a*b^15*c^3*d*f*i*k - 19307520*a^7*b*c^11*d*f*h*j + 16367616*a^7*
b*c^11*d*e*i*j + 3686400*a^7*b*c^11*e*f*g*j + 34947072*a^7*b*c^11*d*e*f*k +
2304*a*b^14*c^4*d*f*g*k - 180*a*b^13*c^5*d*f*h*j + 11059200*a^6*b*c^12*d*e
*h*i + 5160960*a^6*b*c^12*d*f*g*i + 2211840*a^6*b*c^12*e*f*g*h - 4608*a*b^1
3*c^5*d*e*f*k - 2304*a*b^11*c^7*d*f*g*i + 4608*a*b^10*c^8*d*e*f*i + 1548288
0*a^5*b*c^13*d*e*f*g - 13824*a*b^9*c^9*d*e*f*g - 225976320*a^8*b^2*c^9*d*e*
j*k + 112988160*a^8*b^3*c^8*d*g*j*k - 11427840*a^10*b^2*c^7*h*i*j*k - 41779
20*a^9*b^4*c^6*h*i*j*k + 1399296*a^8*b^6*c^5*h*i*j*k - 26880*a^6*b^10*c^3*h
*i*j*k + 16128*a^7*b^8*c^4*h*i*j*k - 61562880*a^9*b^2*c^8*d*i*j*k + 2009088

0*a^9*b^3*c^7*g*h*j*k + 119623680*a^7*b^4*c^8*d*e*j*k + 10485760*a^9*b^3*c^
7*f*i*j*k - 40181760*a^9*b^2*c^8*e*h*j*k - 3778560*a^8*b^5*c^6*g*h*j*k - 13
7797632*a^7*b^2*c^10*d*e*h*k - 1248768*a^7*b^7*c^5*f*i*j*k + 229376*a^6*b^9
*c^4*f*i*j*k + 220160*a^8*b^5*c^6*f*i*j*k - 209664*a^7*b^7*c^5*g*h*j*k + 80
640*a^6*b^9*c^4*g*h*j*k - 8960*a^5*b^11*c^3*f*i*j*k - 59811840*a^7*b^5*c^7*
d*g*j*k + 53084160*a^8*b^2*c^9*e*g*i*k - 11120640*a^8*b^4*c^7*f*g*j*k + 104
55552*a^7*b^6*c^6*d*i*j*k - 9216000*a^9*b^2*c^8*f*g*j*k + 7557120*a^8*b^4*c
^7*e*h*j*k + 7397376*a^8*b^3*c^8*f*h*i*k + 5230080*a^7*b^6*c^6*f*g*j*k - 37
675008*a^8*b^2*c^9*d*h*i*k - 3633408*a^6*b^8*c^5*d*i*j*k + 2211840*a^8*b^4*
c^7*d*i*j*k + 68898816*a^7*b^3*c^9*d*g*h*k - 1695744*a^8*b^2*c^9*g*h*i*j -
1400832*a^7*b^4*c^8*g*h*i*j + 967680*a^7*b^5*c^7*f*h*i*k - 783360*a^6*b^7*c
^6*f*h*i*k - 741888*a^6*b^8*c^5*f*g*j*k + 499968*a^5*b^10*c^4*d*i*j*k + 419
328*a^7*b^6*c^6*e*h*j*k - 253440*a^6*b^6*c^7*g*h*i*j - 161280*a^6*b^8*c^5*e
*h*j*k + 42240*a^5*b^9*c^5*f*h*i*k + 26880*a^5*b^10*c^4*f*g*j*k - 26880*a^4
*b^12*c^3*d*i*j*k + 13824*a^4*b^11*c^4*f*h*i*k + 11520*a^5*b^8*c^6*g*h*i*j
- 768*a^3*b^13*c^3*f*h*i*k + 22241280*a^8*b^3*c^8*e*f*j*k + 14222592*a^6*b^
7*c^6*d*g*j*k - 10460160*a^7*b^5*c^7*e*f*j*k + 8847360*a^7*b^4*c^8*e*g*i*k
- 7741440*a^7*b^4*c^8*f*g*h*k - 7077888*a^6*b^6*c^7*e*g*i*k + 6935040*a^6*b
^6*c^7*d*h*i*k - 6709248*a^8*b^2*c^9*f*g*h*k - 3612672*a^7*b^4*c^8*d*h*i*k
+ 2801664*a^7*b^3*c^9*e*h*i*j + 2506752*a^7*b^3*c^9*f*g*i*j + 2419200*a^6*b
^6*c^7*f*g*h*k - 1661184*a^5*b^9*c^5*d*g*j*k + 1483776*a^6*b^7*c^6*e*f*j*k
- 1463040*a^5*b^8*c^6*d*h*i*k + 884736*a^5*b^8*c^6*e*g*i*k + 838656*a^6*b^5
*c^8*f*g*i*j + 506880*a^6*b^5*c^8*e*h*i*j + 80640*a^4*b^11*c^4*d*g*j*k - 53
760*a^5*b^9*c^5*e*f*j*k - 53760*a^5*b^7*c^7*f*g*i*j - 46080*a^4*b^10*c^5*f*
g*h*k - 34560*a^5*b^8*c^6*f*g*h*k + 25344*a^3*b^12*c^4*d*h*i*k - 23040*a^5*
b^7*c^7*e*h*i*j + 13824*a^4*b^10*c^5*d*h*i*k + 2304*a^3*b^12*c^4*f*g*h*k -
2304*a^2*b^14*c^3*d*h*i*k - 29030400*a^6*b^5*c^8*d*g*h*k + 28606464*a^7*b^3
*c^9*d*f*i*k - 28445184*a^6*b^6*c^7*d*e*j*k + 58060800*a^6*b^4*c^9*d*e*h*k
+ 15482880*a^7*b^3*c^9*e*f*h*k - 8183808*a^7*b^2*c^10*d*g*i*j - 6718464*a^6
*b^5*c^8*d*f*i*k - 5087232*a^7*b^2*c^10*e*g*h*j - 5013504*a^7*b^2*c^10*e*f*
i*j - 4838400*a^6*b^5*c^8*e*f*h*k + 4112640*a^5*b^7*c^7*d*g*h*k - 3663360*a
^5*b^7*c^7*d*f*i*k + 3322368*a^5*b^8*c^6*d*e*j*k - 2285568*a^6*b^4*c^9*d*g*
i*j + 1896960*a^4*b^9*c^6*d*f*i*k + 1843200*a^6*b^3*c^10*f*g*h*i - 1677312*
a^6*b^4*c^9*e*f*i*j - 1658880*a^6*b^4*c^9*e*g*h*j + 68345856*a^6*b^3*c^10*d
*e*f*k + 783360*a^5*b^5*c^9*f*g*h*i + 741888*a^5*b^6*c^8*d*g*i*j - 34172928
*a^6*b^4*c^9*d*f*g*k - 340992*a^3*b^11*c^5*d*f*i*k - 161280*a^4*b^10*c^5*d*
e*j*k + 138240*a^4*b^9*c^6*d*g*h*k + 107520*a^5*b^6*c^8*e*f*i*j + 92160*a^4
*b^9*c^6*e*f*h*k - 89856*a^3*b^11*c^5*d*g*h*k - 80640*a^4*b^8*c^7*d*g*i*j +
69120*a^5*b^7*c^7*e*f*h*k + 69120*a^5*b^6*c^8*e*g*h*j + 27648*a^2*b^13*c^4
*d*f*i*k + 18432*a^4*b^7*c^8*f*g*h*i + 6912*a^2*b^13*c^4*d*g*h*k - 4608*a^3
*b^11*c^5*e*f*h*k - 2304*a^3*b^9*c^7*f*g*h*i + 27164160*a^5*b^6*c^8*d*f*g*k
- 22164480*a^6*b^3*c^10*d*f*h*j - 54328320*a^5*b^5*c^9*d*e*f*k - 17473536*
a^7*b^2*c^10*d*f*g*k - 8225280*a^5*b^6*c^8*d*e*h*k - 8087040*a^4*b^8*c^7*d*
f*g*k + 5677056*a^6*b^3*c^10*e*f*g*j - 5529600*a^6*b^2*c^11*d*g*h*i + 45711
36*a^6*b^3*c^10*d*e*i*j - 3686400*a^6*b^2*c^11*e*f*h*i + 2805120*a^5*b^5*c^

9*d*f*h*j - 2211840*a^5*b^4*c^10*d*g*h*i - 1566720*a^5*b^4*c^10*e*f*h*i - 1
483776*a^5*b^5*c^9*d*e*i*j + 1198080*a^3*b^10*c^6*d*f*g*k + 437184*a^4*b^7*
c^8*d*f*h*j - 322560*a^5*b^5*c^9*e*f*g*j + 317952*a^4*b^6*c^9*d*g*h*i - 276
480*a^4*b^8*c^7*d*e*h*k + 179712*a^3*b^10*c^6*d*e*h*k + 161280*a^4*b^7*c^8*
d*e*i*j - 146268*a^3*b^9*c^7*d*f*h*j - 87552*a^2*b^12*c^5*d*f*g*k - 36864*a
^4*b^6*c^9*e*f*h*i - 13824*a^2*b^12*c^5*d*e*h*k + 9360*a^2*b^11*c^6*d*f*h*j
 + 6912*a^3*b^8*c^8*d*g*h*i - 6912*a^2*b^10*c^7*d*g*h*i + 4608*a^3*b^8*c^8*
e*f*h*i - 24551424*a^6*b^2*c^11*d*e*g*j + 16174080*a^4*b^7*c^8*d*e*f*k + 54
19008*a^5*b^4*c^10*d*e*g*j + 5160960*a^5*b^3*c^11*d*f*g*i + 4423680*a^5*b^3
*c^11*e*f*g*h + 4423680*a^5*b^3*c^11*d*e*h*i - 2396160*a^3*b^9*c^7*d*e*f*k
- 635904*a^4*b^5*c^10*d*e*h*i - 483840*a^4*b^6*c^9*d*e*g*j - 354816*a^3*b^7
*c^9*d*f*g*i + 322560*a^4*b^5*c^10*d*f*g*i + 175104*a^2*b^11*c^6*d*e*f*k +
138240*a^4*b^5*c^10*e*f*g*h + 59904*a^2*b^9*c^8*d*f*g*i - 13824*a^3*b^7*c^9
*e*f*g*h - 13824*a^3*b^7*c^9*d*e*h*i + 13824*a^2*b^9*c^8*d*e*h*i - 16588800
*a^5*b^2*c^12*d*e*g*h - 10321920*a^5*b^2*c^12*d*e*f*i + 1658880*a^4*b^4*c^1
1*d*e*g*h + 709632*a^3*b^6*c^10*d*e*f*i - 645120*a^4*b^4*c^11*d*e*f*i + 124
416*a^3*b^6*c^10*d*e*g*h - 119808*a^2*b^8*c^9*d*e*f*i - 41472*a^2*b^8*c^9*d
*e*g*h + 7741440*a^4*b^3*c^12*d*e*f*g - 2903040*a^3*b^5*c^11*d*e*f*g + 3870
72*a^2*b^7*c^10*d*e*f*g - 381026304*a^11*b*c^7*d*j*k^2 - 241827840*a^10*b*c
^8*d*h*k^2 - 65667072*a^12*b*c^6*h*j*k^2 - 169344*a^7*b^11*c*h*j*k^2 - 2516
5824*a^11*b*c^7*g*i*k^2 - 4915200*a^11*b*c^7*g*j^2*k - 53084160*a^8*b*c^10*
e^2*i*k - 75497472*a^10*b*c^8*e*g*k^2 - 86704128*a^7*b*c^11*d^2*g*k + 56524
8*a^9*b*c^9*h*i^2*j - 168448*a^6*b^12*c*f*j*k^2 - 24576*a^5*b^13*c*g*i*k^2
- 1769472*a^9*b*c^9*g*h^2*k - 17694720*a^9*b*c^9*e*i^2*k - 411264*a^5*b^13*
c*d*j*k^2 - 11520*a^4*b^14*c*f*h*k^2 + 4915200*a^8*b*c^10*f^2*g*k + 2580480
*a^9*b*c^9*e*i*j^2 - 2496000*a^9*b*c^9*f*h*j^2 - 1543680*a^8*b*c^10*f*h^2*j
 + 33408*a*b^14*c^4*d^2*i*k - 59512320*a^6*b*c^12*d^2*f*j + 5087232*a^7*b*c
^11*e^2*h*j + 2727936*a^8*b*c^10*d*i^2*j - 26496*a^3*b^15*c*d*h*k^2 + 11059
20*a^7*b*c^11*e*h^2*i - 107136*a*b^13*c^5*d^2*g*k + 10260*a*b^12*c^6*d^2*h*
j - 10616832*a^6*b*c^12*e^2*g*i - 3538944*a^7*b*c^11*e*g*i^2 + 1843200*a^7*
b*c^11*d*h*i^2 - 18432*a^2*b^16*c*d*f*k^2 - 15552000*a^8*b*c^10*d*f*j^2 + 2
4551424*a^6*b*c^12*d*e^2*j - 37062144*a^5*b*c^13*d^2*f*h + 2580480*a^6*b*c^
12*e*f^2*i + 214272*a*b^12*c^6*d^2*e*k + 65664*a*b^10*c^8*d^2*g*i - 25074*a
*b^11*c^7*d^2*f*j + 420*a*b^12*c^6*d*f^2*j + 6*a*b^15*c^3*d*f*j^2 + 2322432
0*a^5*b*c^13*d^2*e*i + 384*a*b^12*c^6*d*f*i^2 - 5985792*a^6*b*c^12*d*f*h^2
+ 206010*a*b^9*c^9*d^2*f*h - 131328*a*b^9*c^9*d^2*e*i - 6300*a*b^10*c^8*d*f
^2*h + 1350*a*b^11*c^7*d*f*h^2 + 16588800*a^5*b*c^13*d*e^2*h + 3456*a*b^10*
c^8*d*f*g^2 + 435456*a*b^8*c^10*d^2*e*g + 13824*a*b^8*c^10*d*e^2*f + 393216
0*a^11*c^8*h*i*j*k + 27525120*a^10*c^9*d*i*j*k + 82575360*a^9*c^10*d*e*j*k
+ 11796480*a^10*c^9*e*h*j*k + 16515072*a^9*c^10*d*h*i*k + 49545216*a^8*c^11
*d*e*h*k - 2457600*a^8*c^11*e*f*i*j - 1474560*a^7*c^12*e*f*h*i - 10321920*a
^6*c^13*d*e*f*i + 737077248*a^10*b^3*c^6*d*j*k^2 - 518814720*a^9*b^5*c^5*d*
j*k^2 + 441354240*a^9*b^3*c^7*d*h*k^2 - 429871104*a^6*b^2*c^11*d^2*e*k - 27
2212992*a^8*b^5*c^6*d*h*k^2 + 305731584*a^5*b^4*c^10*d^2*e*k + 192412800*a^
8*b^7*c^4*d*j*k^2 + 111912960*a^11*b^3*c^5*h*j*k^2 + 214935552*a^6*b^3*c^10

*d^2*g*k + 202427136*a^7*b^6*c^6*d*f*k^2 - 49904640*a^10*b^5*c^4*h*j*k^2 - 178513920*a^8*b^4*c^7*d*f*k^2 - 152865792*a^5*b^5*c^9*d^2*g*k - 114388992*a^7*b^2*c^10*d^2*i*k + 94961664*a^10*b^2*c^7*e*i*k^2 - 9039872*a^11*b^2*c^6*i*j^2*k - 56494080*a^10*b^4*c^5*f*j*k^2 - 2052096*a^10*b^4*c^5*i*j^2*k + 1327360*a^9*b^6*c^4*i*j^2*k - 158080*a^8*b^8*c^3*i*j^2*k - 47480832*a^10*b^3*c^6*g*i*k^2 + 45576960*a^9*b^6*c^4*f*j*k^2 + 7954560*a^9*b^7*c^3*h*j*k^2 - 104693760*a^9*b^3*c^7*e*g*k^2 + 142080*a^8*b^9*c^2*h*j*k^2 + 16017408*a^10*b^3*c^6*g*j^2*k - 4930560*a^9*b^5*c^5*g*j^2*k - 3649536*a^9*b^2*c^8*h^2*i*k - 1843200*a^8*b^4*c^7*h^2*i*k + 85524480*a^8*b^5*c^6*e*g*k^2 + 474240*a^8*b^7*c^4*g*j^2*k + 288000*a^7*b^6*c^6*h^2*i*k + 63360*a^6*b^8*c^5*h^2*i*k - 8064*a^5*b^10*c^4*h^2*i*k - 1152*a^4*b^12*c^3*h^2*i*k - 15437824*a^11*b^2*c^6*f*j*k^2 - 32034816*a^10*b^2*c^7*e*j^2*k - 14369280*a^8*b^8*c^3*f*j*k^2 - 13271040*a^8*b^3*c^8*g^2*i*k + 80267904*a^7*b^7*c^5*d*h*k^2 + 79626240*a^7*b^2*c^10*e^2*g*k + 11059200*a^9*b^5*c^5*g*i*k^2 + 8847360*a^9*b^2*c^8*g*i^2*k - 42113280*a^7*b^9*c^3*d*j*k^2 + 6389760*a^8*b^7*c^4*g*i*k^2 + 5898240*a^8*b^4*c^7*g*i^2*k - 37601280*a^9*b^4*c^6*f*h*k^2 - 2949120*a^7*b^9*c^3*g*i*k^2 + 2242560*a^7*b^10*c^2*f*j*k^2 - 2211840*a^7*b^5*c^7*g^2*i*k + 1769472*a^6*b^7*c^6*g^2*i*k + 749568*a^8*b^3*c^8*h*i^2*j - 442368*a^7*b^6*c^6*g*i^2*k + 442368*a^6*b^11*c^2*g*i*k^2 - 442368*a^6*b^8*c^5*g*i^2*k + 317952*a^7*b^5*c^7*h*i^2*j - 221184*a^5*b^9*c^5*g^2*i*k + 73728*a^5*b^10*c^4*g*i^2*k + 38400*a^6*b^7*c^6*h*i^2*j - 1920*a^5*b^9*c^5*h*i^2*j + 9861120*a^9*b^4*c^6*e*j^2*k - 110280960*a^4*b^6*c^9*d^2*e*k - 93330432*a^6*b^8*c^5*d*f*k^2 + 24645888*a^8*b^6*c^5*f*h*k^2 + 6359040*a^8*b^3*c^8*g*h^2*k - 22118400*a^9*b^4*c^6*e*i*k^2 - 3862528*a^8*b^2*c^9*f^2*i*k - 2248704*a^7*b^4*c^8*f^2*i*k - 1290240*a^9*b^2*c^8*g*i*j^2 - 948480*a^8*b^6*c^5*e*j^2*k - 860160*a^8*b^4*c^7*g*i*j^2 - 414720*a^7*b^5*c^7*g*h^2*k + 303360*a^6*b^6*c^7*f^2*i*k + 266880*a^5*b^8*c^6*f^2*i*k - 224640*a^6*b^7*c^6*g*h^2*k - 80640*a^7*b^6*c^6*g*i*j^2 - 72960*a^4*b^10*c^5*f^2*i*k + 17280*a^5*b^9*c^5*g*h^2*k + 12672*a^6*b^8*c^5*g*i*j^2 + 5504*a^3*b^12*c^4*f^2*i*k + 3456*a^4*b^11*c^4*g*h^2*k - 384*a^5*b^10*c^4*g*i*j^2 - 128*a^2*b^14*c^3*f^2*i*k + 30265344*a^6*b^4*c^9*d^2*i*k - 12779520*a^8*b^6*c^5*e*i*k^2 - 11796480*a^8*b^3*c^8*e*i^2*k - 8847360*a^7*b^3*c^9*e^2*i*k - 7925760*a^10*b^2*c^7*f*h*k^2 + 7077888*a^6*b^5*c^8*e^2*i*k - 39813120*a^7*b^3*c^9*e*g^2*k - 73175040*a^9*b^2*c^8*d*f*k^2 + 5898240*a^7*b^8*c^4*e*i*k^2 + 5542272*a^6*b^11*c^2*d*j*k^2 - 5420160*a^7*b^8*c^4*f*h*k^2 + 55140480*a^4*b^7*c^8*d^2*g*k + 1271808*a^7*b^3*c^9*g^2*h*j - 1040384*a^8*b^2*c^9*f*i^2*j + 884736*a^7*b^5*c^7*e*i^2*k - 884736*a^6*b^10*c^3*e*i*k^2 + 884736*a^6*b^7*c^6*e*i^2*k - 884736*a^5*b^7*c^7*e^2*i*k - 697344*a^7*b^4*c^8*f*i^2*j + 414720*a^6*b^5*c^8*g^2*h*j + 226560*a^6*b^10*c^3*f*h*k^2 - 147456*a^5*b^9*c^5*e*i^2*k - 121856*a^6*b^6*c^7*f*i^2*j + 82560*a^5*b^12*c^2*f*h*k^2 + 49152*a^5*b^12*c^2*e*i*k^2 - 17280*a^5*b^7*c^7*g^2*h*j + 8960*a^5*b^8*c^6*f*i^2*j + 14194944*a^5*b^6*c^8*d^2*i*k - 12718080*a^8*b^2*c^9*e*h^2*k - 10615680*a^4*b^8*c^7*d^2*i*k - 26542080*a^6*b^4*c^9*e^2*g*k - 23592960*a^7*b^7*c^5*e*g*k^2 - 5142528*a^8*b^3*c^8*f*h*j^2 + 5068800*a^7*b^2*c^10*f^2*h*j - 3755520*a^7*b^3*c^9*f*h^2*j + 3336192*a^7*b^3*c^9*f^2*g*k + 3000960*a^6*b^4*c^9*f^2*h*j + 2893824*a^3*b^10*c^6*d^2*i*k + 1720

320*a^8*b^3*c^8*e*i*j^2 + 1704960*a^6*b^5*c^8*f^2*g*k - 1307520*a^5*b^7*c^7
*f^2*g*k - 1085760*a^6*b^5*c^8*f*h^2*j - 959040*a^7*b^5*c^7*f*h*j^2 + 82944
0*a^7*b^4*c^8*e*h^2*k - 552960*a^7*b^2*c^10*g*h^2*i - 552960*a^6*b^4*c^9*g*
h^2*i + 449280*a^6*b^6*c^7*e*h^2*k - 422784*a^2*b^12*c^5*d^2*i*k + 253440*a
^4*b^9*c^6*f^2*g*k + 161280*a^7*b^5*c^7*e*i*j^2 - 145152*a^5*b^6*c^8*g*h^2*
i + 103200*a^6*b^7*c^6*f*h*j^2 + 41280*a^5*b^6*c^8*f^2*h*j - 37188*a^4*b^8*
c^7*f^2*h*j - 34560*a^5*b^8*c^6*e*h^2*k - 25344*a^6*b^7*c^6*e*i*j^2 - 17280
*a^3*b^11*c^5*f^2*g*k + 13536*a^5*b^7*c^7*f*h^2*j - 6912*a^4*b^10*c^5*e*h^2
*k + 5490*a^4*b^9*c^6*f*h^2*j - 3456*a^4*b^8*c^7*g*h^2*i + 1980*a^3*b^10*c^
6*f^2*h*j + 810*a^5*b^9*c^5*f*h*j^2 + 768*a^5*b^9*c^5*e*i*j^2 + 384*a^2*b^1
3*c^4*f^2*g*k - 270*a^4*b^11*c^4*f*h*j^2 - 180*a^3*b^11*c^5*f*h^2*j - 30*a^
2*b^12*c^5*f^2*h*j + 6*a^3*b^13*c^3*f*h*j^2 + 30067200*a^6*b^2*c^11*d^2*h*j
 + 13271040*a^6*b^5*c^8*e*g^2*k - 10857600*a^6*b^9*c^4*d*h*k^2 + 2949120*a^
6*b^9*c^4*e*g*k^2 + 2654208*a^5*b^6*c^8*e^2*g*k + 2125824*a^7*b^3*c^9*d*i^2
*j + 1658880*a^6*b^3*c^10*e^2*h*j - 1419264*a^6*b^4*c^9*f*g^2*j - 1327104*a
^5*b^7*c^7*e*g^2*k - 921600*a^7*b^2*c^10*f*g^2*j - 737280*a^7*b^2*c^10*f*h*
i^2 - 568320*a^6*b^4*c^9*f*h*i^2 + 207360*a^4*b^13*c^2*d*h*k^2 - 147456*a^5
*b^11*c^3*e*g*k^2 - 136704*a^5*b^6*c^8*f*h*i^2 + 133632*a^6*b^5*c^8*d*i^2*j
 - 96768*a^5*b^7*c^7*d*i^2*j + 80640*a^5*b^6*c^8*f*g^2*j - 69120*a^5*b^5*c^
9*e^2*h*j + 13440*a^4*b^9*c^6*d*i^2*j - 5760*a^5*b^11*c^3*d*h*k^2 - 2304*a^
4*b^8*c^7*f*h*i^2 + 384*a^3*b^10*c^6*f*h*i^2 + 11930112*a^8*b^2*c^9*d*h*j^2
 - 11646720*a^3*b^9*c^7*d^2*g*k + 8432640*a^7*b^2*c^10*d*h^2*j + 24140160*a
^5*b^10*c^4*d*f*k^2 - 6672384*a^7*b^2*c^10*e*f^2*k + 4450176*a^7*b^4*c^8*d*
h*j^2 + 4337280*a^6*b^4*c^9*d*h^2*j - 3870720*a^8*b^2*c^9*e*g*j^2 - 3409920
*a^6*b^4*c^9*e*f^2*k - 2885760*a^5*b^4*c^10*d^2*h*j - 2844288*a^4*b^6*c^9*d
^2*h*j + 2615040*a^5*b^6*c^8*e*f^2*k - 1687680*a^6*b^6*c^7*d*h*j^2 + 148262
4*a^2*b^11*c^6*d^2*g*k - 1290240*a^6*b^2*c^11*f^2*g*i + 1105920*a^6*b^3*c^1
0*e*h^2*i + 1019412*a^3*b^8*c^8*d^2*h*j - 1007424*a^5*b^6*c^8*d*h^2*j - 860
160*a^5*b^4*c^10*f^2*g*i - 645120*a^7*b^4*c^8*e*g*j^2 - 506880*a^4*b^8*c^7*
e*f^2*k + 290304*a^5*b^5*c^9*e*h^2*i + 197460*a^5*b^8*c^6*d*h*j^2 - 143802*
a^2*b^10*c^7*d^2*h*j + 80640*a^6*b^6*c^7*e*g*j^2 - 80640*a^4*b^6*c^9*f^2*g*
i + 51948*a^4*b^8*c^7*d*h^2*j + 34560*a^3*b^10*c^6*e*f^2*k + 12672*a^3*b^8*
c^8*f^2*g*i + 10800*a^3*b^10*c^6*d*h^2*j + 6912*a^4*b^7*c^8*e*h^2*i - 2304*
a^5*b^8*c^6*e*g*j^2 - 768*a^2*b^12*c^5*e*f^2*k - 684*a^3*b^12*c^4*d*h*j^2 -
 540*a^2*b^12*c^5*d*h^2*j - 384*a^2*b^10*c^7*f^2*g*i - 90*a^4*b^10*c^5*d*h*
j^2 + 18*a^2*b^14*c^3*d*h*j^2 + 23385600*a^6*b^2*c^11*d*f^2*j + 23293440*a^
3*b^8*c^8*d^2*e*k + 6137856*a^6*b^3*c^10*d*g^2*j - 5677056*a^6*b^2*c^11*e^2
*f*j + 5308416*a^6*b^2*c^11*e*g^2*i - 5308416*a^5*b^3*c^11*e^2*g*i - 378624
0*a^4*b^12*c^3*d*f*k^2 - 3538944*a^6*b^3*c^10*e*g*i^2 + 2654208*a^5*b^4*c^1
0*e*g^2*i + 1658880*a^6*b^3*c^10*d*h*i^2 - 1354752*a^5*b^5*c^9*d*g^2*j - 11
05920*a^5*b^4*c^10*f*g^2*h - 884736*a^5*b^5*c^9*e*g*i^2 - 552960*a^6*b^2*c^
11*f*g^2*h + 357120*a^3*b^14*c^2*d*f*k^2 + 322560*a^5*b^4*c^10*e^2*f*j + 26
2656*a^5*b^5*c^9*d*h*i^2 + 120960*a^4*b^7*c^8*d*g^2*j - 55296*a^4*b^7*c^8*d
*h*i^2 - 34560*a^4*b^6*c^9*f*g^2*h + 3456*a^3*b^8*c^8*f*g^2*h + 1152*a^3*b^
9*c^7*d*h*i^2 + 1152*a^2*b^11*c^6*d*h*i^2 - 13149696*a^7*b^3*c^9*d*f*j^2 -

11612160*a^5*b^2*c^12*d^2*g*i + 10906560*a^4*b^5*c^10*d^2*f*j - 7418880*a^5
*b^3*c^11*d^2*f*j + 3148992*a^6*b^5*c^8*d*f*j^2 - 2985696*a^3*b^7*c^9*d^2*f
*j - 2965248*a^2*b^10*c^7*d^2*e*k + 1720320*a^5*b^3*c^11*e*f^2*i - 1658880*
a^6*b^2*c^11*e*g*h^2 + 1596672*a^3*b^6*c^10*d^2*g*i - 1505280*a^4*b^6*c^9*d
*f^2*j - 829440*a^5*b^4*c^10*e*g*h^2 - 508032*a^2*b^8*c^9*d^2*g*i + 378954*
a^2*b^9*c^8*d^2*f*j + 362880*a^5*b^4*c^10*d*f^2*j + 296964*a^3*b^8*c^8*d*f^
2*j + 161280*a^4*b^5*c^10*e*f^2*i - 77070*a^4*b^9*c^6*d*f*j^2 - 30240*a^5*b
^7*c^7*d*f*j^2 - 25344*a^3*b^7*c^9*e*f^2*i - 20736*a^4*b^6*c^9*e*g*h^2 - 19
278*a^2*b^10*c^7*d*f^2*j + 8820*a^3*b^11*c^5*d*f*j^2 + 768*a^2*b^9*c^8*e*f^
2*i - 378*a^2*b^13*c^4*d*f*j^2 - 5419008*a^5*b^3*c^11*d*e^2*j - 4423680*a^5
*b^2*c^12*e^2*f*h + 4147200*a^5*b^3*c^11*d*g^2*h - 2580480*a^6*b^2*c^11*d*f
*i^2 - 967680*a^5*b^4*c^10*d*f*i^2 + 483840*a^4*b^5*c^10*d*e^2*j - 414720*a
^4*b^5*c^10*d*g^2*h - 138240*a^4*b^4*c^11*e^2*f*h + 64512*a^4*b^6*c^9*d*f*i
^2 + 39168*a^3*b^8*c^8*d*f*i^2 - 31104*a^3*b^7*c^9*d*g^2*h + 13824*a^3*b^6*
c^10*e^2*f*h + 10368*a^2*b^9*c^8*d*g^2*h - 9216*a^2*b^10*c^7*d*f*i^2 + 1563
0336*a^5*b^2*c^12*d*f^2*h - 14459904*a^4*b^3*c^12*d^2*f*h + 9630144*a^3*b^5
*c^11*d^2*f*h - 8764416*a^5*b^3*c^11*d*f*h^2 - 3870720*a^5*b^2*c^12*e*f^2*g
 - 3193344*a^3*b^5*c^11*d^2*e*i + 2867328*a^4*b^4*c^11*d*f^2*h - 2095200*a^
2*b^7*c^10*d^2*f*h - 1414080*a^3*b^6*c^10*d*f^2*h - 34836480*a^4*b^2*c^13*d
^2*e*g + 1016064*a^2*b^7*c^10*d^2*e*i - 645120*a^4*b^4*c^11*e*f^2*g + 30672
0*a^3*b^7*c^9*d*f*h^2 + 197820*a^2*b^8*c^9*d*f^2*h + 146880*a^4*b^5*c^10*d*
f*h^2 + 80640*a^3*b^6*c^10*e*f^2*g - 55350*a^2*b^9*c^8*d*f*h^2 - 2304*a^2*b
^8*c^9*e*f^2*g - 3870720*a^5*b^2*c^12*d*f*g^2 - 1935360*a^4*b^4*c^11*d*f*g^
2 - 1658880*a^4*b^3*c^12*d*e^2*h + 725760*a^3*b^6*c^10*d*f*g^2 + 17418240*a
^3*b^4*c^12*d^2*e*g - 124416*a^3*b^5*c^11*d*e^2*h - 96768*a^2*b^8*c^9*d*f*g
^2 + 41472*a^2*b^7*c^10*d*e^2*h - 3919104*a^2*b^6*c^11*d^2*e*g - 7741440*a^
4*b^2*c^13*d*e^2*f + 2903040*a^3*b^4*c^12*d*e^2*f - 387072*a^2*b^6*c^11*d*e
^2*f - 681246720*a^9*b*c^9*d^2*k^2 + 265912320*a^11*b^3*c^5*e*k^3 + 1887436
80*a^12*b^2*c^5*g*k^3 - 132956160*a^11*b^4*c^4*g*k^3 - 52101120*a^13*b*c^5*
j^2*k^2 + 25722880*a^12*b^3*c^4*i*k^3 + 19644416*a^11*b^5*c^3*i*k^3 - 15836
80*a^9*b^9*c*j^2*k^2 - 9142272*a^10*b^7*c^2*i*k^3 - 74022912*a^10*b^5*c^4*e
*k^3 - 20643840*a^11*b*c^7*h^2*k^2 + 37011456*a^10*b^6*c^3*g*k^3 - 2293760*
a^9*b^3*c^7*i^3*k - 557056*a^8*b^5*c^6*i^3*k + 147456*a^7*b^7*c^5*i^3*k - 6
5536*a^6*b^12*c*i^2*k^2 + 32768*a^6*b^9*c^4*i^3*k - 8192*a^5*b^11*c^3*i^3*k
 + 430080*a^10*b*c^8*i^2*j^2 - 2880*a^5*b^13*c*h^2*k^2 + 6635520*a^7*b^4*c^
8*g^3*k - 4792320*a^9*b^8*c^2*g*k^3 - 2211840*a^6*b^6*c^7*g^3*k + 1359360*a
^10*b^2*c^7*h*j^3 + 1173120*a^9*b^4*c^6*h*j^3 + 743040*a^7*b^4*c^8*h^3*j +
622080*a^8*b^2*c^9*h^3*j + 221184*a^5*b^8*c^6*g^3*k + 107136*a^6*b^6*c^7*h^
3*j - 32640*a^8*b^6*c^5*h*j^3 - 5796*a^7*b^8*c^4*h*j^3 + 540*a^5*b^8*c^6*h^
3*j - 270*a^4*b^10*c^5*h^3*j + 210*a^6*b^10*c^3*h*j^3 - 2949120*a^10*b*c^8*
f^2*k^2 + 17694720*a^6*b^3*c^10*e^3*k + 184320*a^8*b*c^10*h^2*i^2 - 3520*a^
3*b^15*c*f^2*k^2 + 9584640*a^9*b^7*c^3*e*k^3 - 2293760*a^9*b^3*c^7*f*j^3 -
2293760*a^6*b^3*c^10*f^3*j - 1769472*a^5*b^5*c^9*e^3*k - 884736*a^6*b^3*c^1
0*g^3*i - 589824*a^7*b^3*c^9*g*i^3 - 491520*a^8*b^9*c^2*e*k^3 - 442368*a^5*
b^5*c^9*g^3*i - 294912*a^6*b^5*c^8*g*i^3 - 199360*a^8*b^5*c^6*f*j^3 - 19936

0*a^5*b^5*c^9*f^3*j + 61920*a^7*b^7*c^5*f*j^3 + 61920*a^4*b^7*c^8*f^3*j - 4
9152*a^5*b^7*c^7*g*i^3 - 3682*a^6*b^9*c^4*f*j^3 - 3682*a^3*b^9*c^7*f^3*j +
70*a^5*b^11*c^3*f*j^3 + 70*a^2*b^11*c^6*f^3*j + 3870720*a^8*b*c^10*e^2*j^2
+ 430080*a^7*b*c^11*f^2*i^2 - 14152320*a^4*b^4*c^11*d^3*j + 10644480*a^5*b^
2*c^12*d^3*j + 5483520*a^9*b^2*c^8*d*j^3 + 4269888*a^3*b^6*c^10*d^3*j + 353
8944*a^5*b^2*c^12*e^3*i - 1648128*a^5*b^3*c^11*f^3*h + 1330560*a^8*b^4*c^7*
d*j^3 + 1179648*a^7*b^2*c^10*e*i^3 - 898560*a^6*b^3*c^10*f*h^3 - 826560*a^7
*b^6*c^6*d*j^3 - 607068*a^2*b^8*c^9*d^3*j + 589824*a^6*b^4*c^9*e*i^3 - 3542
40*a^5*b^5*c^9*f*h^3 - 354240*a^4*b^5*c^10*f^3*h + 145188*a^6*b^8*c^5*d*j^3
 + 98304*a^5*b^6*c^8*e*i^3 + 43680*a^3*b^7*c^9*f^3*h - 21600*a^4*b^7*c^8*f*
h^3 - 9576*a^5*b^10*c^4*d*j^3 + 1350*a^3*b^9*c^7*f*h^3 - 1050*a^2*b^9*c^8*f
^3*h - 504*a*b^14*c^4*d^2*j^2 + 210*a^4*b^12*c^3*d*j^3 + 3870720*a^6*b*c^12
*d^2*i^2 + 1658880*a^6*b*c^12*e^2*h^2 - 9792*a*b^11*c^7*d^2*i^2 + 16547328*
a^4*b^2*c^13*d^3*h - 12306816*a^3*b^4*c^12*d^3*h + 37310976*a^3*b^3*c^13*d^
3*f + 3037824*a^2*b^6*c^11*d^3*h - 2654208*a^5*b^3*c^11*e*g^3 + 1949184*a^6
*b^2*c^11*d*h^3 + 1296000*a^5*b^4*c^10*d*h^3 - 155520*a^4*b^6*c^9*d*h^3 - 4
0500*a*b^10*c^8*d^2*h^2 - 8100*a^3*b^8*c^8*d*h^3 + 4050*a^2*b^10*c^7*d*h^3
+ 3870720*a^5*b*c^13*e^2*f^2 + 34836480*a^4*b*c^14*d^2*e^2 - 108864*a*b^9*c
^9*d^2*g^2 - 8068032*a^2*b^5*c^12*d^3*f - 5623296*a^4*b^3*c^12*d*f^3 + 1737
792*a^3*b^5*c^11*d*f^3 - 260190*a*b^8*c^10*d^2*f^2 - 211680*a^2*b^7*c^10*d*
f^3 - 435456*a*b^7*c^11*d^2*e^2 - 377487360*a^12*b*c^6*e*k^3 + 1434977280*a
^8*b^3*c^8*d^2*k^2 + 173408256*a^7*c^12*d^2*e*k + 3276800*a^12*c^7*i*j^2*k
- 125829120*a^13*b*c^5*i*k^3 + 26214400*a^12*c^7*f*j*k^2 + 1179648*a^10*c^9
*h^2*i*k + 13440*a^6*b^13*h*j*k^2 + 50331648*a^11*c^8*e*i*k^2 + 110100480*a
^10*c^9*d*f*k^2 + 57802752*a^8*c^11*d^2*i*k + 9830400*a^11*c^8*e*j^2*k - 32
76800*a^9*c^10*f^2*i*k + 4480*a^5*b^14*f*j*k^2 + 15728640*a^11*c^8*f*h*k^2
- 409600*a^9*c^10*f*i^2*j - 1152*b^16*c^3*d^2*i*k - 1220516352*a^7*b^5*c^7*
d^2*k^2 + 3538944*a^9*c^10*e*h^2*k + 384000*a^8*c^11*f^2*h*j + 13440*a^4*b^
15*d*j*k^2 + 384*a^3*b^16*f*h*k^2 + 20321280*a^7*c^12*d^2*h*j - 245760*a^8*
c^11*f*h*i^2 + 3456*b^15*c^4*d^2*g*k - 270*b^14*c^5*d^2*h*j - 9830400*a^8*c
^11*e*f^2*k + 4838400*a^9*c^10*d*h*j^2 + 2903040*a^8*c^11*d*h^2*j - 1966080
*a^10*b*c^8*i^3*k + 1433600*a^9*b^9*c*i*k^3 + 1152*a^2*b^17*d*h*k^2 - 36864
00*a^7*c^12*e^2*f*j - 53084160*a^7*b*c^11*e^3*k - 6912*b^14*c^5*d^2*e*k - 3
456*b^12*c^7*d^2*g*i + 630*b^13*c^6*d^2*f*j + 2688000*a^7*c^12*d*f^2*j + 24
5760*a^8*b^10*c*g*k^3 - 2211840*a^6*c^13*e^2*f*h - 1720320*a^7*c^12*d*f*i^2
 - 9450*b^11*c^8*d^2*f*h + 6912*b^11*c^8*d^2*e*i + 1612800*a^6*c^13*d*f^2*h
 - 1344000*a^10*b*c^8*f*j^3 - 1344000*a^7*b*c^11*f^3*j - 393216*a^8*b*c^10*
g*i^3 - 23616*a*b^17*c*d^2*k^2 - 20736*b^10*c^9*d^2*e*g - 75188736*a^4*b*c^
14*d^3*f - 883200*a^6*b*c^12*f^3*h - 317952*a^7*b*c^11*f*h^3 + 43416*a*b^10
*c^8*d^3*j - 15482880*a^5*c^14*d*e^2*f - 10616832*a^5*b*c^13*e^3*g - 345060
*a*b^8*c^10*d^3*h - 4262400*a^5*b*c^13*d*f^3 + 852768*a*b^7*c^11*d^3*f + 73
50*a*b^9*c^9*d*f^3 + 584578368*a^6*b^7*c^6*d^2*k^2 + 93905920*a^12*b^3*c^4*
j^2*k^2 - 177997248*a^5*b^9*c^5*d^2*k^2 - 50967040*a^11*b^5*c^3*j^2*k^2 + 1
04693760*a^9*b^2*c^8*e^2*k^2 + 12849984*a^10*b^7*c^2*j^2*k^2 + 20021248*a^1
1*b^2*c^6*i^2*k^2 - 85524480*a^8*b^4*c^7*e^2*k^2 + 33223680*a^10*b^3*c^6*h^

2*k^2 + 4227072*a^10*b^4*c^5*i^2*k^2 - 3973120*a^9*b^6*c^4*i^2*k^2 + 344064
*a^7*b^10*c^2*i^2*k^2 - 81920*a^8*b^8*c^3*i^2*k^2 - 11386368*a^9*b^5*c^5*h^
2*k^2 + 26173440*a^9*b^4*c^6*g^2*k^2 - 21381120*a^8*b^6*c^5*g^2*k^2 + 18874
368*a^10*b^2*c^7*g^2*k^2 + 501760*a^9*b^3*c^7*i^2*j^2 + 452160*a^8*b^7*c^4*
h^2*k^2 + 385920*a^7*b^9*c^3*h^2*k^2 + 170240*a^8*b^5*c^6*i^2*j^2 - 48960*a
^6*b^11*c^2*h^2*k^2 + 9216*a^7*b^7*c^5*i^2*j^2 - 1984*a^6*b^9*c^4*i^2*j^2 +
 64*a^5*b^11*c^3*i^2*j^2 + 5898240*a^7*b^8*c^4*g^2*k^2 + 1419840*a^8*b^4*c^
7*h^2*j^2 + 1387008*a^9*b^2*c^8*h^2*j^2 - 737280*a^6*b^10*c^3*g^2*k^2 + 849
60*a^7*b^6*c^6*h^2*j^2 + 36864*a^5*b^12*c^2*g^2*k^2 - 8010*a^6*b^8*c^5*h^2*
j^2 - 180*a^5*b^10*c^4*h^2*j^2 + 9*a^4*b^12*c^3*h^2*j^2 + 14115840*a^9*b^3*
c^7*f^2*k^2 - 9231552*a^7*b^7*c^5*f^2*k^2 + 23592960*a^7*b^6*c^6*e^2*k^2 +
4984320*a^8*b^5*c^6*f^2*k^2 + 3759040*a^6*b^9*c^4*f^2*k^2 + 36190080*a^4*b^
11*c^4*d^2*k^2 + 967680*a^8*b^3*c^8*g^2*j^2 - 727360*a^5*b^11*c^3*f^2*k^2 +
 276480*a^7*b^3*c^9*h^2*i^2 + 161280*a^7*b^5*c^7*g^2*j^2 + 140544*a^6*b^5*c
^8*h^2*i^2 + 72960*a^4*b^13*c^2*f^2*k^2 + 25344*a^5*b^7*c^7*h^2*i^2 - 20160
*a^6*b^7*c^6*g^2*j^2 + 576*a^5*b^9*c^5*g^2*j^2 + 576*a^4*b^9*c^6*h^2*i^2 +
3808000*a^8*b^2*c^9*f^2*j^2 - 2949120*a^6*b^8*c^5*e^2*k^2 + 1643712*a^7*b^4
*c^8*f^2*j^2 + 884736*a^7*b^2*c^10*g^2*i^2 + 884736*a^6*b^4*c^9*g^2*i^2 + 2
21184*a^5*b^6*c^8*g^2*i^2 + 147456*a^5*b^10*c^4*e^2*k^2 - 125440*a^6*b^6*c^
7*f^2*j^2 - 13790*a^5*b^8*c^6*f^2*j^2 + 1785*a^4*b^10*c^5*f^2*j^2 - 70*a^3*
b^12*c^4*f^2*j^2 - 4953600*a^3*b^13*c^3*d^2*k^2 + 18427392*a^7*b^2*c^10*d^2
*j^2 + 645120*a^7*b^3*c^9*e^2*j^2 + 501760*a^6*b^3*c^10*f^2*i^2 + 442944*a^
2*b^15*c^2*d^2*k^2 + 414720*a^6*b^3*c^10*g^2*h^2 + 207360*a^5*b^5*c^9*g^2*h
^2 + 170240*a^5*b^5*c^9*f^2*i^2 - 80640*a^6*b^5*c^8*e^2*j^2 + 9216*a^4*b^7*
c^8*f^2*i^2 + 5184*a^4*b^7*c^8*g^2*h^2 + 2304*a^5*b^7*c^7*e^2*j^2 - 1984*a^
3*b^9*c^7*f^2*i^2 + 64*a^2*b^11*c^6*f^2*i^2 - 4148928*a^6*b^4*c^9*d^2*j^2 +
 3538944*a^6*b^2*c^11*e^2*i^2 + 1684224*a^6*b^2*c^11*f^2*h^2 + 1264320*a^5*
b^4*c^10*f^2*h^2 - 1183392*a^5*b^6*c^8*d^2*j^2 + 884736*a^5*b^4*c^10*e^2*i^
2 + 645750*a^4*b^8*c^7*d^2*j^2 + 126720*a^4*b^6*c^9*f^2*h^2 - 115920*a^3*b^
10*c^6*d^2*j^2 - 13950*a^3*b^8*c^8*f^2*h^2 + 10836*a^2*b^12*c^5*d^2*j^2 + 2
25*a^2*b^10*c^7*f^2*h^2 + 1935360*a^5*b^3*c^11*d^2*i^2 + 967680*a^5*b^3*c^1
1*f^2*g^2 + 829440*a^5*b^3*c^11*e^2*h^2 - 532224*a^4*b^5*c^10*d^2*i^2 + 161
280*a^4*b^5*c^10*f^2*g^2 - 96768*a^3*b^7*c^9*d^2*i^2 + 62784*a^2*b^9*c^8*d^
2*i^2 + 20736*a^4*b^5*c^10*e^2*h^2 - 20160*a^3*b^7*c^9*f^2*g^2 + 576*a^2*b^
9*c^8*f^2*g^2 + 11487744*a^5*b^2*c^12*d^2*h^2 + 7962624*a^5*b^2*c^12*e^2*g^
2 + 35525376*a^4*b^2*c^13*d^2*f^2 - 1412640*a^3*b^6*c^10*d^2*h^2 + 461376*a
^4*b^4*c^11*d^2*h^2 + 375030*a^2*b^8*c^9*d^2*h^2 + 8709120*a^4*b^3*c^12*d^2
*g^2 - 4354560*a^3*b^5*c^11*d^2*g^2 + 979776*a^2*b^7*c^10*d^2*g^2 + 645120*
a^4*b^3*c^12*e^2*f^2 - 80640*a^3*b^5*c^11*e^2*f^2 + 2304*a^2*b^7*c^10*e^2*f
^2 - 15269184*a^3*b^4*c^12*d^2*f^2 + 2870784*a^2*b^6*c^11*d^2*f^2 - 1741824
0*a^3*b^3*c^13*d^2*e^2 + 3919104*a^2*b^5*c^12*d^2*e^2 + 384*a*b^18*d*f*k^2
- 199229440*a^14*b^2*c^3*k^4 + 8388608*a^12*c^7*i^2*k^2 + 75497472*a^10*c^9
*e^2*k^2 + 78400*a^8*b^11*j^2*k^2 + 4096*a^5*b^14*i^2*k^2 + 345600*a^10*c^9
*h^2*j^2 + 576*a^4*b^15*h^2*k^2 + 57937920*a^13*b^4*c^2*k^4 + 320000*a^9*c^
10*f^2*j^2 + 64*a^2*b^17*f^2*k^2 + 16934400*a^8*c^11*d^2*j^2 + 9*b^16*c^3*d

^2*j^2 + 3538944*a^7*c^12*e^2*i^2 + 115200*a^7*c^12*f^2*h^2 + 576*b^13*c^6*
d^2*i^2 + 2025*b^12*c^7*d^2*h^2 + 6096384*a^6*c^13*d^2*h^2 + 492800*a^11*b^
2*c^6*j^4 + 351456*a^10*b^4*c^5*j^4 - 43120*a^9*b^6*c^4*j^4 + 5184*b^11*c^8
*d^2*g^2 + 1225*a^8*b^8*c^3*j^4 + 131072*a^8*b^2*c^9*i^4 + 98304*a^7*b^4*c^
8*i^4 + 32768*a^6*b^6*c^7*i^4 + 11025*b^10*c^9*d^2*f^2 + 4096*a^5*b^8*c^6*i
^4 + 5644800*a^5*c^14*d^2*f^2 + 142560*a^6*b^4*c^9*h^4 + 103680*a^7*b^2*c^1
0*h^4 + 32400*a^5*b^6*c^8*h^4 + 20736*b^9*c^10*d^2*e^2 + 2025*a^4*b^8*c^7*h
^4 + 331776*a^5*b^4*c^10*g^4 + 492800*a^5*b^2*c^12*f^4 + 351456*a^4*b^4*c^1
1*f^4 - 43120*a^3*b^6*c^10*f^4 + 1225*a^2*b^8*c^9*f^4 - 27433728*a^3*b^2*c^
14*d^4 + 6446304*a^2*b^4*c^13*d^4 + a^2*b^14*c^3*f^2*j^2 - 81920*a^8*b^11*i
*k^3 + 384000*a^11*c^8*h*j^3 + 138240*a^9*c^10*h^3*j + 47416320*a^6*c^13*d^
3*j - 1134*b^12*c^7*d^3*j + 7077888*a^6*c^13*e^3*i + 2688000*a^10*c^9*d*j^3
 + 786432*a^8*c^11*e*i^3 + 28449792*a^5*c^14*d^3*h - 7782400*a^12*b^6*c*k^4
 + 17010*b^10*c^9*d^3*h + 580608*a^7*c^12*d*h^3 - 39690*b^9*c^10*d^3*f - 73
4832*a*b^6*c^12*d^4 + 268435456*a^15*c^4*k^4 + 576*b^19*d^2*k^2 + 409600*a^
11*b^8*k^4 + 160000*a^12*c^7*j^4 + 65536*a^9*c^10*i^4 + 20736*a^8*c^11*h^4
+ 49787136*a^4*c^15*d^4 + 160000*a^6*c^13*f^4 + 5308416*a^5*c^14*e^4 + 3572
1*b^8*c^11*d^4, z, n)*x*(8388608*a^11*b*c^13 - 512*a^4*b^15*c^6 + 14336*a^5
*b^13*c^7 - 172032*a^6*b^11*c^8 + 1146880*a^7*b^9*c^9 - 4587520*a^8*b^7*c^1
0 + 11010048*a^9*b^5*c^11 - 14680064*a^10*b^3*c^12))/(64*(4096*a^10*c^10 +
a^4*b^12*c^4 - 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7*b^6*c^7 + 3840*
a^8*b^4*c^8 - 6144*a^9*b^2*c^9))) - (x*(451584*a^6*c^13*d^2 + 18*b^12*c^7*d
^2 - 25600*a^7*c^12*f^2 + 9216*a^8*c^11*h^2 + 128*a^4*b^15*k^2 + 25600*a^10
*c^9*j^2 - 504*a*b^10*c^8*d^2 - 73728*a^6*b*c^12*e^2 - 8192*a^8*b*c^10*i^2
- 3712*a^5*b^13*c*k^2 - 3538944*a^11*b*c^7*k^2 + 6228*a^2*b^8*c^9*d^2 - 426
24*a^3*b^6*c^10*d^2 + 176256*a^4*b^4*c^11*d^2 - 423936*a^5*b^2*c^12*d^2 - 4
608*a^4*b^5*c^10*e^2 + 36864*a^5*b^3*c^11*e^2 + 2*a^2*b^10*c^7*f^2 - 84*a^3
*b^8*c^8*f^2 + 3520*a^4*b^6*c^9*f^2 - 26240*a^5*b^4*c^10*f^2 + 59904*a^6*b^
2*c^11*f^2 - 1152*a^4*b^7*c^8*g^2 + 9216*a^5*b^5*c^9*g^2 - 18432*a^6*b^3*c^
10*g^2 + 468*a^4*b^8*c^7*h^2 - 3456*a^5*b^6*c^8*h^2 + 5760*a^6*b^4*c^9*h^2
- 128*a^4*b^9*c^6*i^2 + 512*a^5*b^7*c^7*i^2 + 1536*a^6*b^5*c^8*i^2 - 4096*a
^7*b^3*c^9*i^2 + 2*a^4*b^12*c^3*j^2 - 88*a^5*b^10*c^4*j^2 + 1236*a^6*b^8*c^
5*j^2 - 5760*a^7*b^6*c^6*j^2 + 8320*a^8*b^4*c^7*j^2 - 6144*a^9*b^2*c^8*j^2
+ 46464*a^6*b^11*c^2*k^2 - 326400*a^7*b^9*c^3*k^2 + 1394560*a^8*b^7*c^4*k^2
 - 3640320*a^9*b^5*c^5*k^2 + 5404672*a^10*b^3*c^6*k^2 + 129024*a^7*c^12*d*h
 + 215040*a^8*c^11*d*j + 786432*a^9*c^10*e*k + 30720*a^9*c^10*h*j + 262144*
a^10*c^9*i*k + 12*a*b^11*c^7*d*f - 218112*a^6*b*c^12*d*f - 49152*a^7*b*c^11
*e*i - 9216*a^7*b*c^11*f*h - 25600*a^8*b*c^10*f*j - 393216*a^9*b*c^9*g*k -
420*a^2*b^9*c^8*d*f + 4992*a^3*b^7*c^9*d*f - 36480*a^4*b^5*c^10*d*f + 14438
4*a^5*b^3*c^11*d*f + 36*a^2*b^10*c^7*d*h - 360*a^3*b^8*c^8*d*h + 3456*a^4*b
^6*c^9*d*h + 4608*a^4*b^6*c^9*e*g - 11520*a^5*b^4*c^10*d*h - 36864*a^5*b^4*
c^10*e*g - 27648*a^6*b^2*c^11*d*h + 73728*a^6*b^2*c^11*e*g + 12*a^3*b^9*c^7
*f*h - 1536*a^4*b^7*c^8*e*i - 2304*a^4*b^7*c^8*f*h + 168*a^4*b^8*c^7*d*j +
9216*a^5*b^5*c^9*e*i + 17280*a^5*b^5*c^9*f*h - 768*a^5*b^6*c^8*d*j - 30720*
a^6*b^3*c^10*f*h + 11520*a^6*b^4*c^9*d*j - 98304*a^7*b^2*c^10*d*j + 768*a^4

*b^8*c^7*g*i + 140*a^4*b^9*c^6*f*j - 4608*a^5*b^6*c^8*g*i - 3584*a^5*b^7*c^
7*f*j + 1536*a^5*b^8*c^6*e*k + 20352*a^6*b^5*c^8*f*j - 26112*a^6*b^6*c^7*e*
k + 24576*a^7*b^2*c^10*g*i - 26624*a^7*b^3*c^9*f*j + 184320*a^7*b^4*c^8*e*k
 - 614400*a^8*b^2*c^9*e*k - 60*a^4*b^10*c^5*h*j + 1560*a^5*b^8*c^6*h*j - 76
8*a^5*b^9*c^5*g*k - 8832*a^6*b^6*c^7*h*j + 13056*a^6*b^7*c^6*g*k + 13056*a^
7*b^4*c^8*h*j - 92160*a^7*b^5*c^7*g*k - 3072*a^8*b^2*c^9*h*j + 307200*a^8*b
^3*c^8*g*k + 256*a^5*b^10*c^4*i*k - 3840*a^6*b^8*c^5*i*k + 22016*a^7*b^6*c^
6*i*k - 40960*a^8*b^4*c^7*i*k - 73728*a^9*b^2*c^8*i*k))/(64*(4096*a^10*c^10
 + a^4*b^12*c^4 - 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7*b^6*c^7 + 38
40*a^8*b^4*c^8 - 6144*a^9*b^2*c^9))) + (x*(13824*a^4*c^12*e^3 + 512*a^7*c^9
*i^3 - 640*a^7*b^9*k^3 - 54*b^7*c^9*d^2*e + 27*b^8*c^8*d^2*g + 11840*a^8*b^
7*c*k^3 - 376832*a^11*b*c^4*k^3 + 13824*a^5*c^11*e^2*i + 4608*a^6*c^10*e*i^
2 - 9*b^9*c^7*d^2*i + 112896*a^6*c^10*d^2*k + 98304*a^9*c^7*e*k^2 + 9*b^12*
c^4*d^2*k - 6400*a^7*c^9*f^2*k + 64*a^4*b^12*i*k^2 + 2304*a^8*c^8*h^2*k + 3
2768*a^10*c^6*i*k^2 + 6400*a^10*c^6*j^2*k - 1728*a^4*b^3*c^9*g^3 + 64*a^4*b
^6*c^6*i^3 + 384*a^5*b^4*c^7*i^3 + 768*a^6*b^2*c^8*i^3 - 85824*a^9*b^5*c^2*
k^3 + 287296*a^10*b^3*c^3*k^3 - 20160*a^4*c^12*d*e*f - 6720*a^5*c^11*d*f*i
 - 2880*a^5*c^11*e*f*h - 4800*a^6*c^10*e*f*j - 960*a^6*c^10*f*h*i + 32256*a^
7*c^9*d*h*k - 1600*a^7*c^9*f*i*j + 53760*a^8*c^8*d*j*k + 7680*a^9*c^7*h*j*k
 + 972*a*b^5*c^10*d^2*e + 24192*a^3*b*c^12*d^2*e - 486*a*b^6*c^9*d^2*g + 62
40*a^4*b*c^11*e*f^2 - 20736*a^4*b*c^11*e^2*g + 144*a*b^7*c^8*d^2*i + 8064*a
^4*b*c^11*d^2*i + 1728*a^5*b*c^10*e*h^2 - 252*a*b^10*c^5*d^2*k + 2080*a^5*b
*c^10*f^2*i + 3840*a^7*b*c^8*e*j^2 - 2304*a^6*b*c^9*g*i^2 - 122112*a^6*b*c^
9*e^2*k + 576*a^6*b*c^9*h^2*i - 192*a^4*b^11*c*g*k^2 - 49152*a^9*b*c^6*g*k^
2 + 1280*a^8*b*c^7*i*j^2 - 1088*a^5*b^10*c*i*k^2 - 13568*a^8*b*c^7*i^2*k -
7344*a^2*b^3*c^11*d^2*e + 3672*a^2*b^4*c^10*d^2*g - 6*a^2*b^5*c^9*e*f^2 - 1
2096*a^3*b^2*c^11*d^2*g + 192*a^3*b^3*c^10*e*f^2 + 10368*a^4*b^2*c^10*e*g^2
 - 900*a^2*b^5*c^9*d^2*i + 3*a^2*b^6*c^8*f^2*g + 1584*a^3*b^3*c^10*d^2*i -
96*a^3*b^4*c^9*f^2*g - 3120*a^4*b^2*c^10*f^2*g + 1296*a^4*b^3*c^9*e*h^2 + 6
912*a^4*b^2*c^10*e^2*i + 1152*a^4*b^4*c^8*e*i^2 + 4608*a^5*b^2*c^9*e*i^2 -
a^2*b^7*c^7*f^2*i + 3114*a^2*b^8*c^6*d^2*k + 30*a^3*b^5*c^8*f^2*i - 21222*a
^3*b^6*c^7*d^2*k + 1104*a^4*b^3*c^9*f^2*i - 648*a^4*b^4*c^8*g*h^2 + 82584*a
^4*b^4*c^8*d^2*k + 6*a^4*b^7*c^5*e*j^2 - 864*a^5*b^2*c^9*g*h^2 - 166464*a^5
*b^2*c^9*d^2*k - 204*a^5*b^5*c^6*e*j^2 + 1488*a^6*b^3*c^7*e*j^2 + 1728*a^4*
b^4*c^8*g^2*i - 576*a^4*b^5*c^7*g*i^2 - 4608*a^4*b^5*c^7*e^2*k + 384*a^4*b^
10*c^2*e*k^2 + 3456*a^5*b^2*c^9*g^2*i - 2304*a^5*b^3*c^8*g*i^2 + 43776*a^5*
b^3*c^8*e^2*k - 7296*a^5*b^8*c^3*e*k^2 + 54912*a^6*b^6*c^4*e*k^2 - 188160*a
^7*b^4*c^5*e*k^2 + 228480*a^8*b^2*c^6*e*k^2 + a^2*b^10*c^4*f^2*k - 42*a^3*b
^8*c^5*f^2*k + 216*a^4*b^5*c^7*h^2*i + 535*a^4*b^6*c^6*f^2*k - 3*a^4*b^8*c^
4*g*j^2 + 720*a^5*b^3*c^8*h^2*i - 1840*a^5*b^4*c^7*f^2*k + 102*a^5*b^6*c^5*
g*j^2 - 624*a^6*b^2*c^8*f^2*k - 744*a^6*b^4*c^6*g*j^2 - 1920*a^7*b^2*c^7*g*
j^2 - 1152*a^4*b^7*c^5*g^2*k + 10944*a^5*b^5*c^6*g^2*k + 3648*a^5*b^9*c^2*g
*k^2 - 30528*a^6*b^3*c^7*g^2*k - 27456*a^6*b^7*c^3*g*k^2 + 94080*a^7*b^5*c^
4*g*k^2 - 114240*a^8*b^3*c^5*g*k^2 + 9*a^4*b^8*c^4*h^2*k + a^4*b^9*c^3*i*j^
2 + 72*a^5*b^6*c^5*h^2*k - 32*a^5*b^7*c^4*i*j^2 - 360*a^6*b^4*c^6*h^2*k + 1

80*a^6*b^5*c^5*i*j^2 - 4320*a^7*b^2*c^7*h^2*k + 1136*a^7*b^3*c^6*i*j^2 - 12
8*a^4*b^9*c^3*i^2*k + 704*a^5*b^7*c^4*i^2*k + 960*a^6*b^5*c^5*i^2*k + 6720*
a^6*b^8*c^2*i*k^2 - 8704*a^7*b^3*c^6*i^2*k - 13056*a^7*b^6*c^3*i*k^2 - 2464
0*a^8*b^4*c^4*i*k^2 + 92544*a^9*b^2*c^5*i*k^2 - 10*a^7*b^6*c^3*j^2*k + 1560
*a^8*b^4*c^4*j^2*k - 11136*a^9*b^2*c^5*j^2*k - 36*a*b^6*c^9*d*e*f + 18*a*b^
7*c^8*d*f*g + 15552*a^4*b*c^11*d*e*h + 10080*a^4*b*c^11*d*f*g - 6*a*b^8*c^7
*d*f*i + 21888*a^5*b*c^10*d*e*j + 6*a*b^11*c^4*d*f*k + 5184*a^5*b*c^10*d*h*
i - 13824*a^5*b*c^10*e*g*i + 1440*a^5*b*c^10*f*g*h - 4128*a^6*b*c^9*d*f*k +
 7296*a^6*b*c^9*d*i*j + 5184*a^6*b*c^9*e*h*j + 2400*a^6*b*c^9*f*g*j - 81408
*a^7*b*c^8*e*i*k + 4896*a^7*b*c^8*f*h*k + 1728*a^7*b*c^8*h*i*j + 5600*a^8*b
*c^7*f*j*k + 900*a^2*b^4*c^10*d*e*f - 4896*a^3*b^2*c^11*d*e*f - 108*a^2*b^5
*c^9*d*e*h - 450*a^2*b^5*c^9*d*f*g + 2448*a^3*b^3*c^10*d*f*g + 138*a^2*b^6*
c^8*d*f*i + 54*a^2*b^6*c^8*d*g*h - 516*a^3*b^4*c^9*d*f*i - 36*a^3*b^4*c^9*e
*f*h - 4992*a^4*b^2*c^10*d*f*i - 7776*a^4*b^2*c^10*d*g*h - 6048*a^4*b^2*c^1
0*e*f*h - 2016*a^4*b^3*c^9*d*e*j - 18*a^2*b^7*c^7*d*h*i - 210*a^2*b^9*c^5*d
*f*k - 36*a^3*b^5*c^8*d*h*i + 18*a^3*b^5*c^8*f*g*h + 2496*a^3*b^7*c^6*d*f*k
 + 2592*a^4*b^3*c^9*d*h*i - 6912*a^4*b^3*c^9*e*g*i + 3024*a^4*b^3*c^9*f*g*h
 + 1008*a^4*b^4*c^8*d*g*j + 420*a^4*b^4*c^8*e*f*j - 13770*a^4*b^5*c^7*d*f*k
 - 10944*a^5*b^2*c^9*d*g*j - 7392*a^5*b^2*c^9*e*f*j + 31536*a^5*b^3*c^8*d*f
*k + 18*a^2*b^10*c^4*d*h*k - 6*a^3*b^6*c^7*f*h*i - 180*a^3*b^8*c^5*d*h*k -
1020*a^4*b^4*c^8*f*h*i - 336*a^4*b^5*c^7*d*i*j - 180*a^4*b^5*c^7*e*h*j - 21
0*a^4*b^5*c^7*f*g*j - 162*a^4*b^6*c^6*d*h*k + 4608*a^4*b^6*c^6*e*g*k - 2496
*a^5*b^2*c^9*f*h*i + 2976*a^5*b^3*c^8*d*i*j + 2880*a^5*b^3*c^8*e*h*j + 3696
*a^5*b^3*c^8*f*g*j + 10080*a^5*b^4*c^7*d*h*k - 43776*a^5*b^4*c^7*e*g*k - 45
792*a^6*b^2*c^8*d*h*k + 122112*a^6*b^2*c^8*e*g*k + 6*a^3*b^9*c^4*f*h*k + 70
*a^4*b^6*c^6*f*i*j + 90*a^4*b^6*c^6*g*h*j - 1536*a^4*b^7*c^5*e*i*k - 102*a^
4*b^7*c^5*f*h*k + 210*a^4*b^8*c^4*d*j*k - 1092*a^5*b^4*c^7*f*i*j - 1440*a^5
*b^4*c^7*g*h*j + 11520*a^5*b^5*c^6*e*i*k - 390*a^5*b^5*c^6*f*h*k - 3696*a^5
*b^6*c^5*d*j*k - 3264*a^6*b^2*c^8*f*i*j - 2592*a^6*b^2*c^8*g*h*j - 11520*a^
6*b^3*c^7*e*i*k + 5040*a^6*b^3*c^7*f*h*k + 26160*a^6*b^4*c^6*d*j*k - 79296*
a^7*b^2*c^7*d*j*k - 30*a^4*b^7*c^5*h*i*j + 768*a^4*b^8*c^4*g*i*k + 420*a^5*
b^5*c^6*h*i*j - 5760*a^5*b^6*c^5*g*i*k + 70*a^5*b^7*c^4*f*j*k + 1824*a^6*b^
3*c^7*h*i*j + 5760*a^6*b^4*c^6*g*i*k - 1722*a^6*b^5*c^5*f*j*k + 40704*a^7*b
^2*c^7*g*i*k + 7824*a^7*b^3*c^6*f*j*k + 210*a^6*b^6*c^4*h*j*k + 384*a^7*b^4
*c^5*h*j*k - 13728*a^8*b^2*c^6*h*j*k))/(64*(4096*a^10*c^10 + a^4*b^12*c^4 -
 24*a^5*b^10*c^5 + 240*a^6*b^8*c^6 - 1280*a^7*b^6*c^7 + 3840*a^8*b^4*c^8 -
6144*a^9*b^2*c^9)))*root(56371445760*a^11*b^8*c^12*z^4 - 503316480*a^8*b^14
*c^9*z^4 + 47185920*a^7*b^16*c^8*z^4 - 2621440*a^6*b^18*c^7*z^4 + 65536*a^5
*b^20*c^6*z^4 - 171798691840*a^14*b^2*c^15*z^4 + 193273528320*a^13*b^4*c^14
*z^4 - 128849018880*a^12*b^6*c^13*z^4 - 16911433728*a^10*b^10*c^11*z^4 + 35
23215360*a^9*b^12*c^10*z^4 + 68719476736*a^15*c^16*z^4 - 47185920*a^7*b^16*
c^5*k*z^3 + 2621440*a^6*b^18*c^4*k*z^3 - 65536*a^5*b^20*c^3*k*z^3 + 1717986
91840*a^14*b^2*c^12*k*z^3 - 193273528320*a^13*b^4*c^11*k*z^3 + 128849018880
*a^12*b^6*c^10*k*z^3 + 16911433728*a^10*b^10*c^8*k*z^3 - 3523215360*a^9*b^1
2*c^7*k*z^3 - 56371445760*a^11*b^8*c^9*k*z^3 + 503316480*a^8*b^14*c^6*k*z^3

 - 68719476736*a^15*c^13*k*z^3 + 1536*a*b^18*c^6*d*f*z^2 - 2571632640*a^9*b
^5*c^11*d*j*z^2 + 2548039680*a^9*b^3*c^13*d*h*z^2 + 2453667840*a^9*b^7*c^9*
e*k*z^2 + 2181038080*a^12*b^3*c^10*i*k*z^2 - 6492782592*a^10*b^5*c^10*e*k*z
^2 + 1509949440*a^9*b^3*c^13*e*g*z^2 - 1401421824*a^8*b^5*c^12*d*h*z^2 - 12
26833920*a^9*b^8*c^8*g*k*z^2 - 1321205760*a^9*b^2*c^14*d*f*z^2 - 2793406464
*a^11*b*c^13*d*j*z^2 + 9563013120*a^11*b^3*c^11*e*k*z^2 + 890634240*a^8*b^7
*c^10*d*j*z^2 - 754974720*a^8*b^5*c^12*e*g*z^2 - 570425344*a^11*b^5*c^9*i*k
*z^2 + 732168192*a^7*b^6*c^12*d*f*z^2 - 581959680*a^10*b^4*c^11*f*j*z^2 - 6
03979776*a^10*b^2*c^13*e*i*z^2 + 534773760*a^11*b^3*c^11*h*j*z^2 - 55836672
0*a^8*b^9*c^8*e*k*z^2 - 4781506560*a^11*b^4*c^10*g*k*z^2 - 2013265920*a^13*
b*c^11*i*k*z^2 - 456130560*a^9*b^4*c^12*f*h*z^2 + 384040960*a^9*b^6*c^10*f*
j*z^2 - 264241152*a^10*b^7*c^8*i*k*z^2 + 390463488*a^7*b^7*c^11*d*h*z^2 + 2
79183360*a^8*b^10*c^7*g*k*z^2 + 301989888*a^10*b^3*c^12*g*i*z^2 + 222822400
*a^9*b^9*c^7*i*k*z^2 - 366280704*a^6*b^8*c^11*d*f*z^2 - 330301440*a^8*b^4*c
^13*d*f*z^2 + 254017536*a^8*b^6*c^11*f*h*z^2 - 1887436800*a^10*b*c^14*d*h*z
^2 + 188743680*a^10*b^2*c^13*f*h*z^2 - 185303040*a^7*b^9*c^9*d*j*z^2 - 1179
64800*a^10*b^5*c^10*h*j*z^2 - 6039797760*a^12*b*c^12*e*k*z^2 - 67502080*a^8
*b^11*c^6*i*k*z^2 + 121634816*a^11*b^2*c^12*f*j*z^2 + 188743680*a^7*b^7*c^1
1*e*g*z^2 - 115671040*a^8*b^8*c^9*f*j*z^2 + 125829120*a^8*b^6*c^11*e*i*z^2
 + 10813440*a^7*b^13*c^5*i*k*z^2 + 76677120*a^7*b^11*c^7*e*k*z^2 - 38338560*
a^7*b^12*c^6*g*k*z^2 - 37355520*a^9*b^7*c^9*h*j*z^2 - 917504*a^6*b^15*c^4*i
*k*z^2 + 32768*a^5*b^17*c^3*i*k*z^2 - 62914560*a^8*b^7*c^10*g*i*z^2 + 23101
440*a^8*b^9*c^8*h*j*z^2 - 4349952*a^7*b^11*c^7*h*j*z^2 + 2949120*a^6*b^14*c
^5*g*k*z^2 + 337920*a^6*b^13*c^6*h*j*z^2 - 98304*a^5*b^16*c^4*g*k*z^2 - 768
0*a^5*b^15*c^5*h*j*z^2 - 61931520*a^7*b^8*c^10*f*h*z^2 + 23592960*a^7*b^9*c
^9*g*i*z^2 + 17940480*a^7*b^10*c^8*f*j*z^2 - 47185920*a^7*b^8*c^10*e*i*z^2
 - 5898240*a^6*b^13*c^6*e*k*z^2 - 3538944*a^6*b^11*c^8*g*i*z^2 - 1347584*a^6
*b^12*c^7*f*j*z^2 + 196608*a^5*b^15*c^5*e*k*z^2 + 196608*a^5*b^13*c^7*g*i*z
^2 + 35840*a^5*b^14*c^6*f*j*z^2 + 96583680*a^5*b^10*c^10*d*f*z^2 + 23371776
*a^6*b^11*c^8*d*j*z^2 - 51609600*a^6*b^9*c^10*d*h*z^2 + 7077888*a^6*b^10*c^
9*e*i*z^2 + 6144000*a^6*b^10*c^9*f*h*z^2 - 1677312*a^5*b^13*c^7*d*j*z^2 - 3
93216*a^5*b^12*c^8*e*i*z^2 + 61440*a^5*b^12*c^8*f*h*z^2 + 53760*a^4*b^15*c^
6*d*j*z^2 - 46080*a^4*b^14*c^7*f*h*z^2 + 1536*a^3*b^16*c^6*f*h*z^2 - 235929
60*a^6*b^9*c^10*e*g*z^2 + 1179648*a^5*b^11*c^9*e*g*z^2 + 829440*a^4*b^13*c^
8*d*h*z^2 + 368640*a^5*b^11*c^9*d*h*z^2 - 105984*a^3*b^15*c^7*d*h*z^2 + 460
8*a^2*b^17*c^6*d*h*z^2 - 15175680*a^4*b^12*c^9*d*f*z^2 + 1428480*a^3*b^14*c
^8*d*f*z^2 - 73728*a^2*b^16*c^7*d*f*z^2 + 4108320768*a^10*b^3*c^12*d*j*z^2
 - 1207959552*a^10*b*c^14*e*g*z^2 - 578813952*a^12*b*c^12*h*j*z^2 + 32463912
96*a^10*b^6*c^9*g*k*z^2 - 402653184*a^11*b*c^13*g*i*z^2 + 3019898880*a^12*b
^2*c^11*g*k*z^2 - 440401920*a^10*b*c^14*f^2*z^2 - 188743680*a^11*b*c^13*h^2
*z^2 + 1761607680*a^10*c^15*d*f*z^2 - 655360*a^6*b^18*c*k^2*z^2 - 94464*a*b
^17*c^7*d^2*z^2 + 6936330240*a^8*b^3*c^14*d^2*z^2 + 2464874496*a^6*b^7*c^12
*d^2*z^2 - 3963617280*a^9*b*c^15*d^2*z^2 + 58007224320*a^13*b^4*c^8*k^2*z^2
 + 14968422400*a^11*b^8*c^6*k^2*z^2 + 805306368*a^11*c^14*e*i*z^2 - 3596615
6800*a^12*b^6*c^7*k^2*z^2 + 419430400*a^12*c^13*f*j*z^2 - 1509949440*a^9*b^

2*c^14*e^2*z^2 + 251658240*a^11*c^14*f*h*z^2 - 56874762240*a^14*b^2*c^9*k^2
*z^2 - 5400428544*a^7*b^5*c^13*d^2*z^2 + 890470400*a^9*b^12*c^4*k^2*z^2 + 7
54974720*a^8*b^4*c^13*e^2*z^2 - 730054656*a^5*b^9*c^11*d^2*z^2 + 477102080*
a^12*b^3*c^10*j^2*z^2 + 477102080*a^9*b^3*c^13*f^2*z^2 - 377487360*a^9*b^4*
c^12*g^2*z^2 + 301989888*a^10*b^2*c^13*g^2*z^2 - 174325760*a^11*b^5*c^9*j^2
*z^2 - 126156800*a^8*b^14*c^3*k^2*z^2 + 188743680*a^8*b^6*c^11*g^2*z^2 + 14
1557760*a^10*b^3*c^12*h^2*z^2 - 174325760*a^8*b^5*c^12*f^2*z^2 - 188743680*
a^7*b^6*c^12*e^2*z^2 - 4350935040*a^10*b^10*c^5*k^2*z^2 + 146165760*a^4*b^1
1*c^10*d^2*z^2 - 50331648*a^10*b^4*c^11*i^2*z^2 + 11796480*a^7*b^16*c^2*k^2
*z^2 - 33554432*a^11*b^2*c^12*i^2*z^2 + 11206656*a^10*b^7*c^8*j^2*z^2 + 892
9280*a^9*b^9*c^7*j^2*z^2 + 20971520*a^9*b^6*c^10*i^2*z^2 - 2600960*a^8*b^11
*c^6*j^2*z^2 + 291840*a^7*b^13*c^5*j^2*z^2 - 14080*a^6*b^15*c^4*j^2*z^2 + 2
56*a^5*b^17*c^3*j^2*z^2 - 47185920*a^7*b^8*c^10*g^2*z^2 - 26542080*a^8*b^7*
c^10*h^2*z^2 - 2752512*a^7*b^10*c^8*i^2*z^2 + 2621440*a^8*b^8*c^9*i^2*z^2 +
524288*a^6*b^12*c^7*i^2*z^2 - 32768*a^5*b^14*c^6*i^2*z^2 + 9584640*a^7*b^9
*c^9*h^2*z^2 - 2359296*a^9*b^5*c^11*h^2*z^2 - 1290240*a^6*b^11*c^8*h^2*z^2
+ 46080*a^5*b^13*c^7*h^2*z^2 + 2304*a^4*b^15*c^6*h^2*z^2 + 5898240*a^6*b^10
*c^9*g^2*z^2 - 294912*a^5*b^12*c^8*g^2*z^2 + 11206656*a^7*b^7*c^11*f^2*z^2
+ 8929280*a^6*b^9*c^10*f^2*z^2 + 23592960*a^6*b^8*c^11*e^2*z^2 - 2600960*a^
5*b^11*c^9*f^2*z^2 + 291840*a^4*b^13*c^8*f^2*z^2 - 14080*a^3*b^15*c^7*f^2*z
^2 + 256*a^2*b^17*c^6*f^2*z^2 - 19860480*a^3*b^13*c^9*d^2*z^2 - 1179648*a^5
*b^10*c^10*e^2*z^2 + 1771776*a^2*b^15*c^8*d^2*z^2 - 440401920*a^13*b*c^11*j
^2*z^2 + 1207959552*a^10*c^15*e^2*z^2 + 134217728*a^12*c^13*i^2*z^2 + 25769
803776*a^15*c^10*k^2*z^2 + 16384*a^5*b^20*k^2*z^2 + 2304*b^19*c^6*d^2*z^2 +
165150720*a^9*b*c^12*d*g*j*z + 23592960*a^10*b*c^11*g*h*j*z + 169869312*a^
7*b*c^14*d*e*f*z + 99090432*a^8*b*c^13*d*g*h*z - 3145728*a^9*b*c^12*f*h*i*z
+ 56623104*a^8*b*c^13*d*f*i*z - 1536*a*b^18*c^3*d*f*k*z - 9437184*a^8*b*c^
13*e*f*h*z + 1536*a*b^15*c^6*d*f*i*z - 4608*a*b^14*c^7*d*f*g*z + 9216*a*b^1
3*c^8*d*e*f*z + 2173501440*a^9*b^5*c^8*d*j*k*z - 1987706880*a^9*b^3*c^10*d*
h*k*z + 1121255424*a^8*b^5*c^9*d*h*k*z + 861143040*a^8*b^4*c^10*d*f*k*z - 8
59963392*a^7*b^6*c^9*d*f*k*z - 780779520*a^8*b^7*c^7*d*j*k*z - 754974720*a^
9*b^3*c^10*e*g*k*z + 2222456832*a^11*b*c^10*d*j*k*z - 454164480*a^11*b^3*c^
8*h*j*k*z + 377487360*a^8*b^5*c^9*e*g*k*z + 290979840*a^10*b^4*c^8*f*j*k*z
+ 381026304*a^6*b^8*c^8*d*f*k*z + 412876800*a^8*b^2*c^12*d*e*j*z + 30198988
8*a^10*b^2*c^10*e*i*k*z - 320421888*a^7*b^7*c^8*d*h*k*z + 185794560*a^10*b^
5*c^7*h*j*k*z - 192020480*a^9*b^6*c^7*f*j*k*z + 190709760*a^9*b^4*c^9*f*h*k
*z - 150994944*a^10*b^3*c^9*g*i*k*z + 168990720*a^7*b^9*c^6*d*j*k*z + 23592
9600*a^9*b^2*c^11*d*f*k*z - 206438400*a^8*b^3*c^11*d*g*j*z - 206438400*a^7*
b^4*c^11*d*e*j*z - 101646336*a^8*b^6*c^8*f*h*k*z - 29245440*a^9*b^7*c^6*h*j
*k*z - 60817408*a^11*b^2*c^9*f*j*k*z + 57835520*a^8*b^8*c^6*f*j*k*z + 21941
4528*a^7*b^2*c^13*d*e*h*z - 70778880*a^10*b^2*c^10*f*h*k*z + 677376*a^7*b^1
1*c^4*h*j*k*z - 645120*a^8*b^9*c^5*h*j*k*z - 53760*a^6*b^13*c^3*h*j*k*z + 3
1457280*a^8*b^7*c^7*g*i*k*z - 62914560*a^8*b^6*c^8*e*i*k*z - 94371840*a^7*b
^7*c^8*e*g*k*z - 221773824*a^6*b^3*c^13*d*e*f*z + 82575360*a^9*b^2*c^11*d*i
*j*z + 11796480*a^10*b^2*c^10*h*i*j*z - 11796480*a^7*b^9*c^6*g*i*k*z - 8970

240*a^7*b^10*c^5*f*j*k*z + 103219200*a^7*b^5*c^10*d*g*j*z - 2457600*a^8*b^6
*c^8*h*i*j*z + 1769472*a^6*b^11*c^5*g*i*k*z + 921600*a^7*b^8*c^7*h*i*j*z +
673792*a^6*b^12*c^4*f*j*k*z - 138240*a^6*b^10*c^6*h*i*j*z - 98304*a^5*b^13*
c^4*g*i*k*z - 17920*a^5*b^14*c^3*f*j*k*z + 7680*a^5*b^12*c^5*h*i*j*z - 9713
6640*a^5*b^10*c^7*d*f*k*z - 29491200*a^9*b^3*c^10*g*h*j*z + 58982400*a^9*b^
2*c^11*e*h*j*z + 23592960*a^7*b^8*c^7*e*i*k*z - 22169088*a^6*b^11*c^5*d*j*k
*z + 21381120*a^7*b^8*c^7*f*h*k*z + 14745600*a^8*b^5*c^9*g*h*j*z + 42854400
*a^6*b^9*c^7*d*h*k*z - 109707264*a^7*b^3*c^12*d*g*h*z - 3686400*a^7*b^7*c^8
*g*h*j*z - 3538944*a^6*b^10*c^6*e*i*k*z + 1645056*a^5*b^13*c^4*d*j*k*z - 89
0880*a^6*b^10*c^6*f*h*k*z + 460800*a^6*b^9*c^7*g*h*j*z - 330240*a^5*b^12*c^
5*f*h*k*z + 196608*a^5*b^12*c^5*e*i*k*z - 53760*a^4*b^15*c^3*d*j*k*z + 4608
0*a^4*b^14*c^4*f*h*k*z - 23040*a^5*b^11*c^6*g*h*j*z - 1536*a^3*b^16*c^3*f*h
*k*z - 29491200*a^8*b^4*c^10*e*h*j*z - 17203200*a^7*b^6*c^9*d*i*j*z + 11796
480*a^6*b^9*c^7*e*g*k*z + 110886912*a^6*b^4*c^12*d*f*g*z + 7372800*a^7*b^6*
c^9*e*h*j*z + 40108032*a^8*b^2*c^12*d*h*i*z + 6451200*a^6*b^8*c^8*d*i*j*z +
 2359296*a^8*b^3*c^11*f*h*i*z - 967680*a^5*b^10*c^7*d*i*j*z - 921600*a^6*b^
8*c^8*e*h*j*z - 829440*a^4*b^13*c^5*d*h*k*z - 589824*a^5*b^11*c^6*e*g*k*z -
 491520*a^6*b^7*c^9*f*h*i*z + 184320*a^5*b^9*c^8*f*h*i*z + 105984*a^3*b^15*
c^4*d*h*k*z + 69120*a^5*b^11*c^6*d*h*k*z + 53760*a^4*b^12*c^6*d*i*j*z + 460
80*a^5*b^10*c^7*e*h*j*z - 27648*a^4*b^11*c^7*f*h*i*z - 4608*a^2*b^17*c^3*d*
h*k*z + 1536*a^3*b^13*c^6*f*h*i*z - 25804800*a^6*b^7*c^9*d*g*j*z - 88473600
*a^6*b^4*c^12*d*e*h*z + 51609600*a^6*b^6*c^10*d*e*j*z - 84934656*a^7*b^2*c^
13*d*f*g*z + 117964800*a^5*b^5*c^12*d*e*f*z + 15160320*a^4*b^12*c^6*d*f*k*z
 - 45613056*a^7*b^3*c^12*d*f*i*z + 44236800*a^6*b^5*c^11*d*g*h*z - 10321920
*a^6*b^6*c^10*d*h*i*z + 7077888*a^7*b^4*c^11*d*h*i*z - 5898240*a^7*b^4*c^11
*f*g*h*z + 4718592*a^8*b^2*c^12*f*g*h*z + 3225600*a^5*b^9*c^8*d*g*j*z + 294
9120*a^6*b^6*c^10*f*g*h*z + 2396160*a^5*b^8*c^9*d*h*i*z - 1428480*a^3*b^14*
c^5*d*f*k*z - 737280*a^5*b^8*c^9*f*g*h*z - 161280*a^4*b^11*c^7*d*g*j*z + 92
160*a^4*b^10*c^8*f*g*h*z + 73728*a^2*b^16*c^4*d*f*k*z - 50688*a^3*b^12*c^7*
d*h*i*z - 27648*a^4*b^10*c^8*d*h*i*z - 4608*a^3*b^12*c^7*f*g*h*z + 4608*a^2
*b^14*c^6*d*h*i*z - 58982400*a^5*b^6*c^11*d*f*g*z + 11796480*a^7*b^3*c^12*e
*f*h*z + 8847360*a^5*b^7*c^10*d*f*i*z - 6635520*a^5*b^7*c^10*d*g*h*z - 6451
200*a^5*b^8*c^9*d*e*j*z - 5898240*a^6*b^5*c^11*e*f*h*z - 3809280*a^4*b^9*c^
9*d*f*i*z + 2359296*a^6*b^5*c^11*d*f*i*z + 1474560*a^5*b^7*c^10*e*f*h*z + 6
81984*a^3*b^11*c^8*d*f*i*z + 322560*a^4*b^10*c^8*d*e*j*z - 276480*a^4*b^9*c
^9*d*g*h*z - 184320*a^4*b^9*c^9*e*f*h*z + 179712*a^3*b^11*c^8*d*g*h*z - 552
96*a^2*b^13*c^7*d*f*i*z - 13824*a^2*b^13*c^7*d*g*h*z + 9216*a^3*b^11*c^8*e*
f*h*z + 16220160*a^4*b^8*c^10*d*f*g*z + 13271040*a^5*b^6*c^11*d*e*h*z - 239
6160*a^3*b^10*c^9*d*f*g*z + 552960*a^4*b^8*c^10*d*e*h*z - 359424*a^3*b^10*c
^9*d*e*h*z + 175104*a^2*b^12*c^8*d*f*g*z + 27648*a^2*b^12*c^8*d*e*h*z - 324
40320*a^4*b^7*c^11*d*e*f*z + 4792320*a^3*b^9*c^10*d*e*f*z - 350208*a^2*b^11
*c^9*d*e*f*z + 1439170560*a^10*b*c^11*d*h*k*z - 3361603584*a^10*b^3*c^9*d*j
*k*z + 603979776*a^10*b*c^11*e*g*k*z + 407371776*a^12*b*c^9*h*j*k*z + 20132
6592*a^11*b*c^10*g*i*k*z + 346816512*a^7*b*c^14*d^2*g*z + 129761280*a^11*b*
c^10*h^2*k*z + 121896960*a^10*b*c^11*f^2*k*z + 458752*a^6*b^15*c*i*k^2*z +

19660800*a^11*b*c^10*g*j^2*z + 49152*a^5*b^16*c*g*k^2*z + 7077888*a^9*b*c^1
2*g*h^2*z + 94464*a*b^17*c^4*d^2*k*z - 19660800*a^8*b*c^13*f^2*g*z - 66816*
a*b^14*c^7*d^2*i*z + 214272*a*b^13*c^8*d^2*g*z - 428544*a*b^12*c^9*d^2*e*z
+ 2390753280*a^11*b^4*c^7*g*k^2*z - 2411421696*a^6*b^7*c^9*d^2*k*z - 660307
9680*a^8*b^3*c^11*d^2*k*z + 3715891200*a^9*b*c^12*d^2*k*z - 880803840*a^10*
c^12*d*f*k*z - 1623195648*a^10*b^6*c^6*g*k^2*z - 402653184*a^11*c^11*e*i*k*
z - 1509949440*a^12*b^2*c^8*g*k^2*z - 209715200*a^12*c^10*f*j*k*z - 3303014
40*a^9*c^13*d*e*j*z + 3019898880*a^12*b*c^9*e*k^2*z - 125829120*a^11*c^11*f
*h*k*z - 110100480*a^10*c^12*d*i*j*z - 198180864*a^8*c^14*d*e*h*z - 1572864
0*a^11*c^11*h*i*j*z - 1226833920*a^9*b^7*c^6*e*k^2*z - 47185920*a^10*c^12*e
*h*j*z - 66060288*a^9*c^13*d*h*i*z - 1090519040*a^12*b^3*c^7*i*k^2*z + 1022
754816*a^6*b^2*c^14*d^2*e*z + 5216108544*a^7*b^5*c^10*d^2*k*z + 754974720*a
^9*b^2*c^11*e^2*k*z + 721529856*a^5*b^9*c^8*d^2*k*z + 613416960*a^9*b^8*c^5
*g*k^2*z - 642318336*a^5*b^4*c^13*d^2*e*z - 4781506560*a^11*b^3*c^8*e*k^2*z
 - 398131200*a^12*b^3*c^7*j^2*k*z - 511377408*a^6*b^3*c^13*d^2*g*z - 377487
360*a^8*b^4*c^10*e^2*k*z + 285212672*a^11*b^5*c^6*i*k^2*z + 199065600*a^11*
b^5*c^6*j^2*k*z + 279183360*a^8*b^9*c^5*e*k^2*z + 321159168*a^5*b^5*c^12*d^
2*g*z + 188743680*a^9*b^4*c^9*g^2*k*z + 132120576*a^10*b^7*c^5*i*k^2*z - 15
0994944*a^10*b^2*c^10*g^2*k*z - 111411200*a^9*b^9*c^4*i*k^2*z - 126812160*a
^10*b^3*c^9*h^2*k*z + 225312768*a^7*b^2*c^13*d^2*i*z - 139591680*a^8*b^10*c
^4*g*k^2*z - 49766400*a^10*b^7*c^5*j^2*k*z - 145463040*a^4*b^11*c^7*d^2*k*z
 - 94371840*a^8*b^6*c^8*g^2*k*z + 223395840*a^4*b^6*c^12*d^2*e*z + 33751040
*a^8*b^11*c^3*i*k^2*z - 78970880*a^9*b^3*c^10*f^2*k*z + 94371840*a^7*b^6*c^
9*e^2*k*z + 25165824*a^10*b^4*c^8*i^2*k*z + 6220800*a^9*b^9*c^4*j^2*k*z + 3
9223296*a^9*b^5*c^8*h^2*k*z - 311040*a^8*b^11*c^3*j^2*k*z + 16777216*a^11*b
^2*c^9*i^2*k*z - 10485760*a^9*b^6*c^7*i^2*k*z - 5406720*a^7*b^13*c^2*i*k^2*
z + 1376256*a^7*b^10*c^5*i^2*k*z - 1310720*a^8*b^8*c^6*i^2*k*z - 262144*a^6
*b^12*c^4*i^2*k*z + 16384*a^5*b^14*c^3*i^2*k*z + 10354688*a^11*b^2*c^9*i*j^
2*z + 23592960*a^7*b^8*c^7*g^2*k*z + 38559744*a^7*b^7*c^8*f^2*k*z + 1916928
0*a^7*b^12*c^3*g*k^2*z - 2048000*a^9*b^6*c^7*i*j^2*z - 1520640*a^7*b^9*c^6*
h^2*k*z - 1105920*a^8*b^7*c^7*h^2*k*z + 849920*a^8*b^8*c^6*i*j^2*z - 393216
*a^10*b^4*c^8*i*j^2*z + 195840*a^6*b^11*c^5*h^2*k*z - 145920*a^7*b^10*c^5*i
*j^2*z + 11520*a^5*b^13*c^4*h^2*k*z + 11008*a^6*b^12*c^4*i*j^2*z - 2304*a^4
*b^15*c^3*h^2*k*z - 256*a^5*b^14*c^3*i*j^2*z - 25362432*a^10*b^3*c^9*g*j^2*
z - 24739840*a^8*b^5*c^9*f^2*k*z - 38338560*a^7*b^11*c^4*e*k^2*z - 2949120*
a^6*b^10*c^6*g^2*k*z - 1474560*a^6*b^14*c^2*g*k^2*z + 50724864*a^10*b^2*c^1
0*e*j^2*z + 147456*a^5*b^12*c^5*g^2*k*z - 15150080*a^6*b^9*c^7*f^2*k*z + 13
271040*a^9*b^5*c^8*g*j^2*z - 111697920*a^4*b^7*c^11*d^2*g*z - 3563520*a^8*b
^7*c^7*g*j^2*z + 3538944*a^9*b^2*c^11*h^2*i*z + 2912000*a^5*b^11*c^6*f^2*k*
z - 737280*a^7*b^6*c^9*h^2*i*z + 506880*a^7*b^9*c^6*g*j^2*z - 291840*a^4*b^
13*c^5*f^2*k*z + 276480*a^6*b^8*c^8*h^2*i*z - 41472*a^5*b^10*c^7*h^2*i*z -
34560*a^6*b^11*c^5*g*j^2*z + 14080*a^3*b^15*c^4*f^2*k*z + 2304*a^4*b^12*c^6
*h^2*i*z + 768*a^5*b^13*c^4*g*j^2*z - 256*a^2*b^17*c^3*f^2*k*z - 11796480*a
^6*b^8*c^8*e^2*k*z - 26542080*a^9*b^4*c^9*e*j^2*z + 19837440*a^3*b^13*c^6*d
^2*k*z + 2949120*a^6*b^13*c^3*e*k^2*z + 589824*a^5*b^10*c^7*e^2*k*z - 98304

*a^5*b^15*c^2*e*k^2*z - 10354688*a^8*b^2*c^12*f^2*i*z - 43646976*a^6*b^4*c^
12*d^2*i*z - 8847360*a^8*b^3*c^11*g*h^2*z + 7127040*a^8*b^6*c^8*e*j^2*z + 4
423680*a^7*b^5*c^10*g*h^2*z + 2048000*a^6*b^6*c^10*f^2*i*z - 1771776*a^2*b^
15*c^5*d^2*k*z - 1105920*a^6*b^7*c^9*g*h^2*z - 1013760*a^7*b^8*c^7*e*j^2*z
- 849920*a^5*b^8*c^9*f^2*i*z + 393216*a^7*b^4*c^11*f^2*i*z + 145920*a^4*b^1
0*c^8*f^2*i*z + 138240*a^5*b^9*c^8*g*h^2*z + 69120*a^6*b^10*c^6*e*j^2*z - 1
1008*a^3*b^12*c^7*f^2*i*z - 6912*a^4*b^11*c^7*g*h^2*z - 1536*a^5*b^12*c^5*e
*j^2*z + 256*a^2*b^14*c^6*f^2*i*z - 32587776*a^5*b^6*c^11*d^2*i*z + 2536243
2*a^7*b^3*c^12*f^2*g*z + 21657600*a^4*b^8*c^10*d^2*i*z + 17694720*a^8*b^2*c
^12*e*h^2*z - 50724864*a^7*b^2*c^13*e*f^2*z - 13271040*a^6*b^5*c^11*f^2*g*z
- 8847360*a^7*b^4*c^11*e*h^2*z - 5810688*a^3*b^10*c^9*d^2*i*z + 3563520*a^
5*b^7*c^10*f^2*g*z + 2211840*a^6*b^6*c^10*e*h^2*z + 845568*a^2*b^12*c^8*d^2
*i*z - 506880*a^4*b^9*c^9*f^2*g*z - 276480*a^5*b^8*c^9*e*h^2*z + 34560*a^3*
b^11*c^8*f^2*g*z + 13824*a^4*b^10*c^8*e*h^2*z - 768*a^2*b^13*c^7*f^2*g*z +
26542080*a^6*b^4*c^12*e*f^2*z + 23362560*a^3*b^9*c^10*d^2*g*z - 46725120*a^
3*b^8*c^11*d^2*e*z - 7127040*a^5*b^6*c^11*e*f^2*z - 2965248*a^2*b^11*c^9*d^
2*g*z + 1013760*a^4*b^8*c^10*e*f^2*z - 69120*a^3*b^10*c^9*e*f^2*z + 1536*a^
2*b^12*c^8*e*f^2*z + 5930496*a^2*b^10*c^10*d^2*e*z + 1006632960*a^13*b*c^8*
i*k^2*z + 3246391296*a^10*b^5*c^7*e*k^2*z + 318504960*a^13*b*c^8*j^2*k*z +
61538304*a^10*b^10*c^2*k^3*z - 603979776*a^10*c^12*e^2*k*z - 693633024*a^7*
c^15*d^2*e*z - 231211008*a^8*c^14*d^2*i*z - 67108864*a^12*c^10*i^2*k*z - 13
107200*a^12*c^10*i*j^2*z - 16384*a^5*b^17*i*k^2*z - 39321600*a^11*c^11*e*j^
2*z - 4718592*a^10*c^12*h^2*i*z - 2304*b^19*c^3*d^2*k*z + 13107200*a^9*c^13
*f^2*i*z + 2304*b^16*c^6*d^2*i*z - 14155776*a^9*c^13*e*h^2*z + 39321600*a^8
*c^14*e*f^2*z - 4833280*a^9*b^12*c*k^3*z - 6912*b^15*c^7*d^2*g*z + 69625446
40*a^14*b^2*c^6*k^3*z + 13824*b^14*c^8*d^2*e*z + 1876951040*a^12*b^6*c^4*k^
3*z - 4844421120*a^13*b^4*c^5*k^3*z - 437780480*a^11*b^8*c^3*k^3*z - 429496
7296*a^15*c^7*k^3*z + 163840*a^8*b^14*k^3*z + 6144000*a^10*b*c^8*f*i*j*k -
5898240*a^10*b*c^8*g*h*j*k - 41287680*a^9*b*c^9*d*g*j*k + 4472832*a^9*b*c^9
*f*h*i*k + 18432000*a^9*b*c^9*e*f*j*k + 3391488*a^8*b*c^10*e*h*i*j + 122880
0*a^8*b*c^10*f*g*i*j - 24772608*a^8*b*c^10*d*g*h*k + 13418496*a^8*b*c^10*e*
f*h*k + 11649024*a^8*b*c^10*d*f*i*k + 737280*a^7*b*c^11*f*g*h*i - 768*a*b^1
5*c^3*d*f*i*k - 19307520*a^7*b*c^11*d*f*h*j + 16367616*a^7*b*c^11*d*e*i*j +
3686400*a^7*b*c^11*e*f*g*j + 34947072*a^7*b*c^11*d*e*f*k + 2304*a*b^14*c^4
*d*f*g*k - 180*a*b^13*c^5*d*f*h*j + 11059200*a^6*b*c^12*d*e*h*i + 5160960*a
^6*b*c^12*d*f*g*i + 2211840*a^6*b*c^12*e*f*g*h - 4608*a*b^13*c^5*d*e*f*k -
2304*a*b^11*c^7*d*f*g*i + 4608*a*b^10*c^8*d*e*f*i + 15482880*a^5*b*c^13*d*e
*f*g - 13824*a*b^9*c^9*d*e*f*g - 225976320*a^8*b^2*c^9*d*e*j*k + 112988160*
a^8*b^3*c^8*d*g*j*k - 11427840*a^10*b^2*c^7*h*i*j*k - 4177920*a^9*b^4*c^6*h
*i*j*k + 1399296*a^8*b^6*c^5*h*i*j*k - 26880*a^6*b^10*c^3*h*i*j*k + 16128*a
^7*b^8*c^4*h*i*j*k - 61562880*a^9*b^2*c^8*d*i*j*k + 20090880*a^9*b^3*c^7*g*
h*j*k + 119623680*a^7*b^4*c^8*d*e*j*k + 10485760*a^9*b^3*c^7*f*i*j*k - 4018
1760*a^9*b^2*c^8*e*h*j*k - 3778560*a^8*b^5*c^6*g*h*j*k - 137797632*a^7*b^2*
c^10*d*e*h*k - 1248768*a^7*b^7*c^5*f*i*j*k + 229376*a^6*b^9*c^4*f*i*j*k + 2
20160*a^8*b^5*c^6*f*i*j*k - 209664*a^7*b^7*c^5*g*h*j*k + 80640*a^6*b^9*c^4*

g*h*j*k - 8960*a^5*b^11*c^3*f*i*j*k - 59811840*a^7*b^5*c^7*d*g*j*k + 530841
60*a^8*b^2*c^9*e*g*i*k - 11120640*a^8*b^4*c^7*f*g*j*k + 10455552*a^7*b^6*c^
6*d*i*j*k - 9216000*a^9*b^2*c^8*f*g*j*k + 7557120*a^8*b^4*c^7*e*h*j*k + 739
7376*a^8*b^3*c^8*f*h*i*k + 5230080*a^7*b^6*c^6*f*g*j*k - 37675008*a^8*b^2*c
^9*d*h*i*k - 3633408*a^6*b^8*c^5*d*i*j*k + 2211840*a^8*b^4*c^7*d*i*j*k + 68
898816*a^7*b^3*c^9*d*g*h*k - 1695744*a^8*b^2*c^9*g*h*i*j - 1400832*a^7*b^4*
c^8*g*h*i*j + 967680*a^7*b^5*c^7*f*h*i*k - 783360*a^6*b^7*c^6*f*h*i*k - 741
888*a^6*b^8*c^5*f*g*j*k + 499968*a^5*b^10*c^4*d*i*j*k + 419328*a^7*b^6*c^6*
e*h*j*k - 253440*a^6*b^6*c^7*g*h*i*j - 161280*a^6*b^8*c^5*e*h*j*k + 42240*a
^5*b^9*c^5*f*h*i*k + 26880*a^5*b^10*c^4*f*g*j*k - 26880*a^4*b^12*c^3*d*i*j*
k + 13824*a^4*b^11*c^4*f*h*i*k + 11520*a^5*b^8*c^6*g*h*i*j - 768*a^3*b^13*c
^3*f*h*i*k + 22241280*a^8*b^3*c^8*e*f*j*k + 14222592*a^6*b^7*c^6*d*g*j*k -
10460160*a^7*b^5*c^7*e*f*j*k + 8847360*a^7*b^4*c^8*e*g*i*k - 7741440*a^7*b^
4*c^8*f*g*h*k - 7077888*a^6*b^6*c^7*e*g*i*k + 6935040*a^6*b^6*c^7*d*h*i*k -
6709248*a^8*b^2*c^9*f*g*h*k - 3612672*a^7*b^4*c^8*d*h*i*k + 2801664*a^7*b^
3*c^9*e*h*i*j + 2506752*a^7*b^3*c^9*f*g*i*j + 2419200*a^6*b^6*c^7*f*g*h*k -
1661184*a^5*b^9*c^5*d*g*j*k + 1483776*a^6*b^7*c^6*e*f*j*k - 1463040*a^5*b^
8*c^6*d*h*i*k + 884736*a^5*b^8*c^6*e*g*i*k + 838656*a^6*b^5*c^8*f*g*i*j + 5
06880*a^6*b^5*c^8*e*h*i*j + 80640*a^4*b^11*c^4*d*g*j*k - 53760*a^5*b^9*c^5*
e*f*j*k - 53760*a^5*b^7*c^7*f*g*i*j - 46080*a^4*b^10*c^5*f*g*h*k - 34560*a^
5*b^8*c^6*f*g*h*k + 25344*a^3*b^12*c^4*d*h*i*k - 23040*a^5*b^7*c^7*e*h*i*j
+ 13824*a^4*b^10*c^5*d*h*i*k + 2304*a^3*b^12*c^4*f*g*h*k - 2304*a^2*b^14*c^
3*d*h*i*k - 29030400*a^6*b^5*c^8*d*g*h*k + 28606464*a^7*b^3*c^9*d*f*i*k - 2
8445184*a^6*b^6*c^7*d*e*j*k + 58060800*a^6*b^4*c^9*d*e*h*k + 15482880*a^7*b
^3*c^9*e*f*h*k - 8183808*a^7*b^2*c^10*d*g*i*j - 6718464*a^6*b^5*c^8*d*f*i*k
- 5087232*a^7*b^2*c^10*e*g*h*j - 5013504*a^7*b^2*c^10*e*f*i*j - 4838400*a^
6*b^5*c^8*e*f*h*k + 4112640*a^5*b^7*c^7*d*g*h*k - 3663360*a^5*b^7*c^7*d*f*i
*k + 3322368*a^5*b^8*c^6*d*e*j*k - 2285568*a^6*b^4*c^9*d*g*i*j + 1896960*a^
4*b^9*c^6*d*f*i*k + 1843200*a^6*b^3*c^10*f*g*h*i - 1677312*a^6*b^4*c^9*e*f*
i*j - 1658880*a^6*b^4*c^9*e*g*h*j + 68345856*a^6*b^3*c^10*d*e*f*k + 783360*
a^5*b^5*c^9*f*g*h*i + 741888*a^5*b^6*c^8*d*g*i*j - 34172928*a^6*b^4*c^9*d*f
*g*k - 340992*a^3*b^11*c^5*d*f*i*k - 161280*a^4*b^10*c^5*d*e*j*k + 138240*a
^4*b^9*c^6*d*g*h*k + 107520*a^5*b^6*c^8*e*f*i*j + 92160*a^4*b^9*c^6*e*f*h*k
- 89856*a^3*b^11*c^5*d*g*h*k - 80640*a^4*b^8*c^7*d*g*i*j + 69120*a^5*b^7*c
^7*e*f*h*k + 69120*a^5*b^6*c^8*e*g*h*j + 27648*a^2*b^13*c^4*d*f*i*k + 18432
*a^4*b^7*c^8*f*g*h*i + 6912*a^2*b^13*c^4*d*g*h*k - 4608*a^3*b^11*c^5*e*f*h*
k - 2304*a^3*b^9*c^7*f*g*h*i + 27164160*a^5*b^6*c^8*d*f*g*k - 22164480*a^6*
b^3*c^10*d*f*h*j - 54328320*a^5*b^5*c^9*d*e*f*k - 17473536*a^7*b^2*c^10*d*f
*g*k - 8225280*a^5*b^6*c^8*d*e*h*k - 8087040*a^4*b^8*c^7*d*f*g*k + 5677056*
a^6*b^3*c^10*e*f*g*j - 5529600*a^6*b^2*c^11*d*g*h*i + 4571136*a^6*b^3*c^10*
d*e*i*j - 3686400*a^6*b^2*c^11*e*f*h*i + 2805120*a^5*b^5*c^9*d*f*h*j - 2211
840*a^5*b^4*c^10*d*g*h*i - 1566720*a^5*b^4*c^10*e*f*h*i - 1483776*a^5*b^5*c
^9*d*e*i*j + 1198080*a^3*b^10*c^6*d*f*g*k + 437184*a^4*b^7*c^8*d*f*h*j - 32
2560*a^5*b^5*c^9*e*f*g*j + 317952*a^4*b^6*c^9*d*g*h*i - 276480*a^4*b^8*c^7*
d*e*h*k + 179712*a^3*b^10*c^6*d*e*h*k + 161280*a^4*b^7*c^8*d*e*i*j - 146268

*a^3*b^9*c^7*d*f*h*j - 87552*a^2*b^12*c^5*d*f*g*k - 36864*a^4*b^6*c^9*e*f*h
*i - 13824*a^2*b^12*c^5*d*e*h*k + 9360*a^2*b^11*c^6*d*f*h*j + 6912*a^3*b^8*
c^8*d*g*h*i - 6912*a^2*b^10*c^7*d*g*h*i + 4608*a^3*b^8*c^8*e*f*h*i - 245514
24*a^6*b^2*c^11*d*e*g*j + 16174080*a^4*b^7*c^8*d*e*f*k + 5419008*a^5*b^4*c^
10*d*e*g*j + 5160960*a^5*b^3*c^11*d*f*g*i + 4423680*a^5*b^3*c^11*e*f*g*h +
4423680*a^5*b^3*c^11*d*e*h*i - 2396160*a^3*b^9*c^7*d*e*f*k - 635904*a^4*b^5
*c^10*d*e*h*i - 483840*a^4*b^6*c^9*d*e*g*j - 354816*a^3*b^7*c^9*d*f*g*i + 3
22560*a^4*b^5*c^10*d*f*g*i + 175104*a^2*b^11*c^6*d*e*f*k + 138240*a^4*b^5*c
^10*e*f*g*h + 59904*a^2*b^9*c^8*d*f*g*i - 13824*a^3*b^7*c^9*e*f*g*h - 13824
*a^3*b^7*c^9*d*e*h*i + 13824*a^2*b^9*c^8*d*e*h*i - 16588800*a^5*b^2*c^12*d*
e*g*h - 10321920*a^5*b^2*c^12*d*e*f*i + 1658880*a^4*b^4*c^11*d*e*g*h + 7096
32*a^3*b^6*c^10*d*e*f*i - 645120*a^4*b^4*c^11*d*e*f*i + 124416*a^3*b^6*c^10
*d*e*g*h - 119808*a^2*b^8*c^9*d*e*f*i - 41472*a^2*b^8*c^9*d*e*g*h + 7741440
*a^4*b^3*c^12*d*e*f*g - 2903040*a^3*b^5*c^11*d*e*f*g + 387072*a^2*b^7*c^10*
d*e*f*g - 381026304*a^11*b*c^7*d*j*k^2 - 241827840*a^10*b*c^8*d*h*k^2 - 656
67072*a^12*b*c^6*h*j*k^2 - 169344*a^7*b^11*c*h*j*k^2 - 25165824*a^11*b*c^7*
g*i*k^2 - 4915200*a^11*b*c^7*g*j^2*k - 53084160*a^8*b*c^10*e^2*i*k - 754974
72*a^10*b*c^8*e*g*k^2 - 86704128*a^7*b*c^11*d^2*g*k + 565248*a^9*b*c^9*h*i^
2*j - 168448*a^6*b^12*c*f*j*k^2 - 24576*a^5*b^13*c*g*i*k^2 - 1769472*a^9*b*
c^9*g*h^2*k - 17694720*a^9*b*c^9*e*i^2*k - 411264*a^5*b^13*c*d*j*k^2 - 1152
0*a^4*b^14*c*f*h*k^2 + 4915200*a^8*b*c^10*f^2*g*k + 2580480*a^9*b*c^9*e*i*j
^2 - 2496000*a^9*b*c^9*f*h*j^2 - 1543680*a^8*b*c^10*f*h^2*j + 33408*a*b^14*
c^4*d^2*i*k - 59512320*a^6*b*c^12*d^2*f*j + 5087232*a^7*b*c^11*e^2*h*j + 27
27936*a^8*b*c^10*d*i^2*j - 26496*a^3*b^15*c*d*h*k^2 + 1105920*a^7*b*c^11*e*
h^2*i - 107136*a*b^13*c^5*d^2*g*k + 10260*a*b^12*c^6*d^2*h*j - 10616832*a^6
*b*c^12*e^2*g*i - 3538944*a^7*b*c^11*e*g*i^2 + 1843200*a^7*b*c^11*d*h*i^2 -
 18432*a^2*b^16*c*d*f*k^2 - 15552000*a^8*b*c^10*d*f*j^2 + 24551424*a^6*b*c^
12*d*e^2*j - 37062144*a^5*b*c^13*d^2*f*h + 2580480*a^6*b*c^12*e*f^2*i + 214
272*a*b^12*c^6*d^2*e*k + 65664*a*b^10*c^8*d^2*g*i - 25074*a*b^11*c^7*d^2*f*
j + 420*a*b^12*c^6*d*f^2*j + 6*a*b^15*c^3*d*f*j^2 + 23224320*a^5*b*c^13*d^2
*e*i + 384*a*b^12*c^6*d*f*i^2 - 5985792*a^6*b*c^12*d*f*h^2 + 206010*a*b^9*c
^9*d^2*f*h - 131328*a*b^9*c^9*d^2*e*i - 6300*a*b^10*c^8*d*f^2*h + 1350*a*b^
11*c^7*d*f*h^2 + 16588800*a^5*b*c^13*d*e^2*h + 3456*a*b^10*c^8*d*f*g^2 + 43
5456*a*b^8*c^10*d^2*e*g + 13824*a*b^8*c^10*d*e^2*f + 3932160*a^11*c^8*h*i*j
*k + 27525120*a^10*c^9*d*i*j*k + 82575360*a^9*c^10*d*e*j*k + 11796480*a^10*
c^9*e*h*j*k + 16515072*a^9*c^10*d*h*i*k + 49545216*a^8*c^11*d*e*h*k - 24576
00*a^8*c^11*e*f*i*j - 1474560*a^7*c^12*e*f*h*i - 10321920*a^6*c^13*d*e*f*i
+ 737077248*a^10*b^3*c^6*d*j*k^2 - 518814720*a^9*b^5*c^5*d*j*k^2 + 44135424
0*a^9*b^3*c^7*d*h*k^2 - 429871104*a^6*b^2*c^11*d^2*e*k - 272212992*a^8*b^5*
c^6*d*h*k^2 + 305731584*a^5*b^4*c^10*d^2*e*k + 192412800*a^8*b^7*c^4*d*j*k^
2 + 111912960*a^11*b^3*c^5*h*j*k^2 + 214935552*a^6*b^3*c^10*d^2*g*k + 20242
7136*a^7*b^6*c^6*d*f*k^2 - 49904640*a^10*b^5*c^4*h*j*k^2 - 178513920*a^8*b^
4*c^7*d*f*k^2 - 152865792*a^5*b^5*c^9*d^2*g*k - 114388992*a^7*b^2*c^10*d^2*
i*k + 94961664*a^10*b^2*c^7*e*i*k^2 - 9039872*a^11*b^2*c^6*i*j^2*k - 564940
80*a^10*b^4*c^5*f*j*k^2 - 2052096*a^10*b^4*c^5*i*j^2*k + 1327360*a^9*b^6*c^

4*i*j^2*k - 158080*a^8*b^8*c^3*i*j^2*k - 47480832*a^10*b^3*c^6*g*i*k^2 + 45
576960*a^9*b^6*c^4*f*j*k^2 + 7954560*a^9*b^7*c^3*h*j*k^2 - 104693760*a^9*b^
3*c^7*e*g*k^2 + 142080*a^8*b^9*c^2*h*j*k^2 + 16017408*a^10*b^3*c^6*g*j^2*k
- 4930560*a^9*b^5*c^5*g*j^2*k - 3649536*a^9*b^2*c^8*h^2*i*k - 1843200*a^8*b
^4*c^7*h^2*i*k + 85524480*a^8*b^5*c^6*e*g*k^2 + 474240*a^8*b^7*c^4*g*j^2*k
+ 288000*a^7*b^6*c^6*h^2*i*k + 63360*a^6*b^8*c^5*h^2*i*k - 8064*a^5*b^10*c^
4*h^2*i*k - 1152*a^4*b^12*c^3*h^2*i*k - 15437824*a^11*b^2*c^6*f*j*k^2 - 320
34816*a^10*b^2*c^7*e*j^2*k - 14369280*a^8*b^8*c^3*f*j*k^2 - 13271040*a^8*b^
3*c^8*g^2*i*k + 80267904*a^7*b^7*c^5*d*h*k^2 + 79626240*a^7*b^2*c^10*e^2*g*
k + 11059200*a^9*b^5*c^5*g*i*k^2 + 8847360*a^9*b^2*c^8*g*i^2*k - 42113280*a
^7*b^9*c^3*d*j*k^2 + 6389760*a^8*b^7*c^4*g*i*k^2 + 5898240*a^8*b^4*c^7*g*i^
2*k - 37601280*a^9*b^4*c^6*f*h*k^2 - 2949120*a^7*b^9*c^3*g*i*k^2 + 2242560*
a^7*b^10*c^2*f*j*k^2 - 2211840*a^7*b^5*c^7*g^2*i*k + 1769472*a^6*b^7*c^6*g^
2*i*k + 749568*a^8*b^3*c^8*h*i^2*j - 442368*a^7*b^6*c^6*g*i^2*k + 442368*a^
6*b^11*c^2*g*i*k^2 - 442368*a^6*b^8*c^5*g*i^2*k + 317952*a^7*b^5*c^7*h*i^2*
j - 221184*a^5*b^9*c^5*g^2*i*k + 73728*a^5*b^10*c^4*g*i^2*k + 38400*a^6*b^7
*c^6*h*i^2*j - 1920*a^5*b^9*c^5*h*i^2*j + 9861120*a^9*b^4*c^6*e*j^2*k - 110
280960*a^4*b^6*c^9*d^2*e*k - 93330432*a^6*b^8*c^5*d*f*k^2 + 24645888*a^8*b^
6*c^5*f*h*k^2 + 6359040*a^8*b^3*c^8*g*h^2*k - 22118400*a^9*b^4*c^6*e*i*k^2
- 3862528*a^8*b^2*c^9*f^2*i*k - 2248704*a^7*b^4*c^8*f^2*i*k - 1290240*a^9*b
^2*c^8*g*i*j^2 - 948480*a^8*b^6*c^5*e*j^2*k - 860160*a^8*b^4*c^7*g*i*j^2 -
414720*a^7*b^5*c^7*g*h^2*k + 303360*a^6*b^6*c^7*f^2*i*k + 266880*a^5*b^8*c^
6*f^2*i*k - 224640*a^6*b^7*c^6*g*h^2*k - 80640*a^7*b^6*c^6*g*i*j^2 - 72960*
a^4*b^10*c^5*f^2*i*k + 17280*a^5*b^9*c^5*g*h^2*k + 12672*a^6*b^8*c^5*g*i*j^
2 + 5504*a^3*b^12*c^4*f^2*i*k + 3456*a^4*b^11*c^4*g*h^2*k - 384*a^5*b^10*c^
4*g*i*j^2 - 128*a^2*b^14*c^3*f^2*i*k + 30265344*a^6*b^4*c^9*d^2*i*k - 12779
520*a^8*b^6*c^5*e*i*k^2 - 11796480*a^8*b^3*c^8*e*i^2*k - 8847360*a^7*b^3*c^
9*e^2*i*k - 7925760*a^10*b^2*c^7*f*h*k^2 + 7077888*a^6*b^5*c^8*e^2*i*k - 39
813120*a^7*b^3*c^9*e*g^2*k - 73175040*a^9*b^2*c^8*d*f*k^2 + 5898240*a^7*b^8
*c^4*e*i*k^2 + 5542272*a^6*b^11*c^2*d*j*k^2 - 5420160*a^7*b^8*c^4*f*h*k^2 +
55140480*a^4*b^7*c^8*d^2*g*k + 1271808*a^7*b^3*c^9*g^2*h*j - 1040384*a^8*b
^2*c^9*f*i^2*j + 884736*a^7*b^5*c^7*e*i^2*k - 884736*a^6*b^10*c^3*e*i*k^2 +
884736*a^6*b^7*c^6*e*i^2*k - 884736*a^5*b^7*c^7*e^2*i*k - 697344*a^7*b^4*c
^8*f*i^2*j + 414720*a^6*b^5*c^8*g^2*h*j + 226560*a^6*b^10*c^3*f*h*k^2 - 147
456*a^5*b^9*c^5*e*i^2*k - 121856*a^6*b^6*c^7*f*i^2*j + 82560*a^5*b^12*c^2*f
*h*k^2 + 49152*a^5*b^12*c^2*e*i*k^2 - 17280*a^5*b^7*c^7*g^2*h*j + 8960*a^5*
b^8*c^6*f*i^2*j + 14194944*a^5*b^6*c^8*d^2*i*k - 12718080*a^8*b^2*c^9*e*h^2
*k - 10615680*a^4*b^8*c^7*d^2*i*k - 26542080*a^6*b^4*c^9*e^2*g*k - 23592960
*a^7*b^7*c^5*e*g*k^2 - 5142528*a^8*b^3*c^8*f*h*j^2 + 5068800*a^7*b^2*c^10*f
^2*h*j - 3755520*a^7*b^3*c^9*f*h^2*j + 3336192*a^7*b^3*c^9*f^2*g*k + 300096
0*a^6*b^4*c^9*f^2*h*j + 2893824*a^3*b^10*c^6*d^2*i*k + 1720320*a^8*b^3*c^8*
e*i*j^2 + 1704960*a^6*b^5*c^8*f^2*g*k - 1307520*a^5*b^7*c^7*f^2*g*k - 10857
60*a^6*b^5*c^8*f*h^2*j - 959040*a^7*b^5*c^7*f*h*j^2 + 829440*a^7*b^4*c^8*e*
h^2*k - 552960*a^7*b^2*c^10*g*h^2*i - 552960*a^6*b^4*c^9*g*h^2*i + 449280*a
^6*b^6*c^7*e*h^2*k - 422784*a^2*b^12*c^5*d^2*i*k + 253440*a^4*b^9*c^6*f^2*g

*k + 161280*a^7*b^5*c^7*e*i*j^2 - 145152*a^5*b^6*c^8*g*h^2*i + 103200*a^6*b^7*c^6*f*h*j^2 + 41280*a^5*b^6*c^8*f^2*h*j - 37188*a^4*b^8*c^7*f^2*h*j - 34560*a^5*b^8*c^6*e*h^2*k - 25344*a^6*b^7*c^6*e*i*j^2 - 17280*a^3*b^11*c^5*f^2*g*k + 13536*a^5*b^7*c^7*f*h^2*j - 6912*a^4*b^10*c^5*e*h^2*k + 5490*a^4*b^9*c^6*f*h^2*j - 3456*a^4*b^8*c^7*g*h^2*i + 1980*a^3*b^10*c^6*f^2*h*j + 810*a^5*b^9*c^5*f*h*j^2 + 768*a^5*b^9*c^5*e*i*j^2 + 384*a^2*b^13*c^4*f^2*g*k - 270*a^4*b^11*c^4*f*h*j^2 - 180*a^3*b^11*c^5*f*h^2*j - 30*a^2*b^12*c^5*f^2*h*j + 6*a^3*b^13*c^3*f*h*j^2 + 30067200*a^6*b^2*c^11*d^2*h*j + 13271040*a^6*b^5*c^8*e*g^2*k - 10857600*a^6*b^9*c^4*d*h*k^2 + 2949120*a^6*b^9*c^4*e*g*k^2 + 2654208*a^5*b^6*c^8*e^2*g*k + 2125824*a^7*b^3*c^9*d*i^2*j + 1658880*a^6*b^3*c^10*e^2*h*j - 1419264*a^6*b^4*c^9*f*g^2*j - 1327104*a^5*b^7*c^7*e*g^2*k - 921600*a^7*b^2*c^10*f*g^2*j - 737280*a^7*b^2*c^10*f*h*i^2 - 568320*a^6*b^4*c^9*f*h*i^2 + 207360*a^4*b^13*c^2*d*h*k^2 - 147456*a^5*b^11*c^3*e*g*k^2 - 136704*a^5*b^6*c^8*f*h*i^2 + 133632*a^6*b^5*c^8*d*i^2*j - 96768*a^5*b^7*c^7*d*i^2*j + 80640*a^5*b^6*c^8*f*g^2*j - 69120*a^5*b^5*c^9*e^2*h*j + 13440*a^4*b^9*c^6*d*i^2*j - 5760*a^5*b^11*c^3*d*h*k^2 - 2304*a^4*b^8*c^7*f*h*i^2 + 384*a^3*b^10*c^6*f*h*i^2 + 11930112*a^8*b^2*c^9*d*h*j^2 - 11646720*a^3*b^9*c^7*d^2*g*k + 8432640*a^7*b^2*c^10*d*h^2*j + 24140160*a^5*b^10*c^4*d*f*k^2 - 6672384*a^7*b^2*c^10*e*f^2*k + 4450176*a^7*b^4*c^8*d*h*j^2 + 4337280*a^6*b^4*c^9*d*h^2*j - 3870720*a^8*b^2*c^9*e*g*j^2 - 3409920*a^6*b^4*c^9*e*f^2*k - 2885760*a^5*b^4*c^10*d^2*h*j - 2844288*a^4*b^6*c^9*d^2*h*j + 2615040*a^5*b^6*c^8*e*f^2*k - 1687680*a^6*b^6*c^7*d*h*j^2 + 1482624*a^2*b^11*c^6*d^2*g*k - 1290240*a^6*b^2*c^11*f^2*g*i + 1105920*a^6*b^3*c^10*e*h^2*i + 1019412*a^3*b^8*c^8*d^2*h*j - 1007424*a^5*b^6*c^8*d*h^2*j - 860160*a^5*b^4*c^10*f^2*g*i - 645120*a^7*b^4*c^8*e*g*j^2 - 506880*a^4*b^8*c^7*e*f^2*k + 290304*a^5*b^5*c^9*e*h^2*i + 197460*a^5*b^8*c^6*d*h*j^2 - 143802*a^2*b^10*c^7*d^2*h*j + 80640*a^6*b^6*c^7*e*g*j^2 - 80640*a^4*b^6*c^9*f^2*g*i + 51948*a^4*b^8*c^7*d*h^2*j + 34560*a^3*b^10*c^6*e*f^2*k + 12672*a^3*b^8*c^8*f^2*g*i + 10800*a^3*b^10*c^6*d*h^2*j + 6912*a^4*b^7*c^8*e*h^2*i - 2304*a^5*b^8*c^6*e*g*j^2 - 768*a^2*b^12*c^5*e*f^2*k - 684*a^3*b^12*c^4*d*h*j^2 - 540*a^2*b^12*c^5*d*h^2*j - 384*a^2*b^10*c^7*f^2*g*i - 90*a^4*b^10*c^5*d*h*j^2 + 18*a^2*b^14*c^3*d*h*j^2 + 23385600*a^6*b^2*c^11*d*f^2*j + 23293440*a^3*b^8*c^8*d^2*e*k + 6137856*a^6*b^3*c^10*d*g^2*j - 5677056*a^6*b^2*c^11*e^2*f*j + 5308416*a^6*b^2*c^11*e*g^2*i - 5308416*a^5*b^3*c^11*e^2*g*i - 3786240*a^4*b^12*c^3*d*f*k^2 - 3538944*a^6*b^3*c^10*e*g*i^2 + 2654208*a^5*b^4*c^10*e*g^2*i + 1658880*a^6*b^3*c^10*d*h*i^2 - 1354752*a^5*b^5*c^9*d*g^2*j - 1105920*a^5*b^4*c^10*f*g^2*h - 884736*a^5*b^5*c^9*e*g*i^2 - 552960*a^6*b^2*c^11*f*g^2*h + 357120*a^3*b^14*c^2*d*f*k^2 + 322560*a^5*b^4*c^10*e^2*f*j + 262656*a^5*b^5*c^9*d*h*i^2 + 120960*a^4*b^7*c^8*d*g^2*j - 55296*a^4*b^7*c^8*d*h*i^2 - 34560*a^4*b^6*c^9*f*g^2*h + 3456*a^3*b^8*c^8*f*g^2*h + 1152*a^3*b^9*c^7*d*h*i^2 + 1152*a^2*b^11*c^6*d*h*i^2 - 13149696*a^7*b^3*c^9*d*f*j^2 - 11612160*a^5*b^2*c^12*d^2*g*i + 10906560*a^4*b^5*c^10*d^2*f*j - 7418880*a^5*b^3*c^11*d^2*f*j + 3148992*a^6*b^5*c^8*d*f*j^2 - 2985696*a^3*b^7*c^9*d^2*f*j - 2965248*a^2*b^10*c^7*d^2*e*k + 1720320*a^5*b^3*c^11*e*f^2*i - 1658880*a^6*b^2*c^11*e*g*h^2 + 1596672*a^3*b^6*c^10*d^2*g*i - 1505280*a^4*b^6*c^9*d*f^2*j - 829440* a^5*b^4*c^10*e*g*h^2 - 508032*a^2*b^8*c^9*d^2*g*i + 378954*a^2*b^9*c^8*d^2*f*j + 362880*a^5*b^4*c^10*d*f^2*j + 296964*a^3*b^8*c^8*d*f^2*j + 161280*a^4*b^5*c^10*e*f^2*i - 77070*a^4*b^9*c^6*d*f*j^2 - 30240*a^5*b^7*c^7*d*f*j^2 - 25344*a^3*b^7*c^9*e*f^2*i - 20736*a^4*b^6*c^9*e*g*h^2 - 19278*a^2*b^10*c^7*d*f^2*j + 8820*a^3*b^11*c^5*d*f*j^2 + 768*a^2*b^9*c^8*e*f^2*i - 378*a^2*b^13*c^4*d*f*j^2 - 5419008*a^5*b^3*c^11*d*e^2*j - 4423680*a^5*b^2*c^12*e^2*f*h + 4147200*a^5*b^3*c^11*d*g^2*h - 2580480*a^6*b^2*c^11*d*f*i^2 - 967680*a^5*b^4*c^10*d*f*i^2 + 483840*a^4*b^5*c^10*d*e^2*j - 414720*a^4*b^5*c^10*d*g^2*h - 138240*a^4*b^4*c^11*e^2*f*h + 64512*a^4*b^6*c^9*d*f*i^2 + 39168*a^3*b^8*c^8*d*f*i^2 - 31104*a^3*b^7*c^9*d*g^2*h + 13824*a^3*b^6*c^10*e^2*f*h + 10368*a^2*b^9*c^8*d*g^2*h - 9216*a^2*b^10*c^7*d*f*i^2 + 15630336*a^5*b^2*c^12*d*f^2*h - 14459904*a^4*b^3*c^12*d^2*f*h + 9630144*a^3*b^5*c^11*d^2*f*h - 8764416*a^5*b^3*c^11*d*f*h^2 - 3870720*a^5*b^2*c^12*e*f^2*g - 3193344*a^3*b^5*c^11*d^2*e*i + 2867328*a^4*b^4*c^11*d*f^2*h - 2095200*a^2*b^7*c^10*d^2*f*h - 1414080*a^3*b^6*c^10*d*f^2*h - 34836480*a^4*b^2*c^13*d^2*e*g + 1016064*a^2*b^7*c^10*d^2*e*i - 645120*a^4*b^4*c^11*e*f^2*g + 306720*a^3*b^7*c^9*d*f*h^2 + 197820*a^2*b^8*c^9*d*f^2*h + 146880*a^4*b^5*c^10*d*f*h^2 + 80640*a^3*b^6*c^10*e*f^2*g - 55350*a^2*b^9*c^8*d*f*h^2 - 2304*a^2*b^8*c^9*e*f^2*g - 3870720*a^5*b^2*c^12*d*f*g^2 - 1935360*a^4*b^4*c^11*d*f*g^2 - 1658880*a^4*b^3*c^12*d*e^2*h + 725760*a^3*b^6*c^10*d*f*g^2 + 17418240*a^3*b^4*c^12*d^2*e*g - 124416*a^3*b^5*c^11*d*e^2*h - 96768*a^2*b^8*c^9*d*f*g^2 + 41472*a^2*b^7*c^10*d*e^2*h - 3919104*a^2*b^6*c^11*d^2*e*g - 7741440*a^4*b^2*c^13*d*e^2*f + 2903040*a^3*b^4*c^12*d*e^2*f - 387072*a^2*b^6*c^11*d*e^2*f - 681246720*a^9*b*c^9*d^2*k^2 + 265912320*a^11*b^3*c^5*e*k^3 + 188743680*a^12*b^2*c^5*g*k^3 - 132956160*a^11*b^4*c^4*g*k^3 - 52101120*a^13*b*c^5*j^2*k^2 + 25722880*a^12*b^3*c^4*i*k^3 + 19644416*a^11*b^5*c^3*i*k^3 - 1583680*a^9*b^9*c*j^2*k^2 - 9142272*a^10*b^7*c^2*i*k^3 - 74022912*a^10*b^5*c^4*e*k^3 - 20643840*a^11*b*c^7*h^2*k^2 + 37011456*a^10*b^6*c^3*g*k^3 - 2293760*a^9*b^3*c^7*i^3*k - 557056*a^8*b^5*c^6*i^3*k + 147456*a^7*b^7*c^5*i^3*k - 65536*a^6*b^12*c*i^2*k^2 + 32768*a^6*b^9*c^4*i^3*k - 8192*a^5*b^11*c^3*i^3*k + 430080*a^10*b*c^8*i^2*j^2 - 2880*a^5*b^13*c*h^2*k^2 + 6635520*a^7*b^4*c^8*g^3*k - 4792320*a^9*b^8*c^2*g*k^3 - 2211840*a^6*b^6*c^7*g^3*k + 1359360*a^10*b^2*c^7*h*j^3 + 1173120*a^9*b^4*c^6*h*j^3 + 743040*a^7*b^4*c^8*h^3*j + 622080*a^8*b^2*c^9*h^3*j + 221184*a^5*b^8*c^6*g^3*k + 107136*a^6*b^6*c^7*h^3*j - 32640*a^8*b^6*c^5*h*j^3 - 5796*a^7*b^8*c^4*h*j^3 + 540*a^5*b^8*c^6*h^3*j - 270*a^4*b^10*c^5*h^3*j + 210*a^6*b^10*c^3*h*j^3 - 2949120*a^10*b*c^8*f^2*k^2 + 17694720*a^6*b^3*c^10*e^3*k + 184320*a^8*b*c^10*h^2*i^2 - 3520*a^3*b^15*c*f^2*k^2 + 9584640*a^9*b^7*c^3*e*k^3 - 2293760*a^9*b^3*c^7*f*j^3 - 2293760*a^6*b^3*c^10*f^3*j - 1769472*a^5*b^5*c^9*e^3*k - 884736*a^6*b^3*c^10*g^3*i - 589824*a^7*b^3*c^9*g*i^3 - 491520*a^8*b^9*c^2*e*k^3 - 442368*a^5*b^5*c^9*g^3*i - 294912*a^6*b^5*c^8*g*i^3 - 199360*a^8*b^5*c^6*f*j^3 - 199360*a^5*b^5*c^9*f^3*j + 61920*a^7*b^7*c^5*f*j^3 + 61920*a^4*b^7*c^8*f^3*j - 49152*a^5*b^7*c^7*g*i^3 - 3682*a^6*b^9*c^4*f*j^3 - 3682*a^3*b^9*c^7*f^3*j + 70*a^5*b^11*c^3*f*j^3 + 70*a^2*b^11*c^6*f^3*j + 3870720*a^8*b*c^10*e^2*j^2 + 430080*a^7*b*c^11*f^2*i^2 - 14152320*a^4*b^4*c^11*d^3*j + 10644480*a^5*b^2*c^12*d^3*j + 5

483520*a^9*b^2*c^8*d*j^3 + 4269888*a^3*b^6*c^10*d^3*j + 3538944*a^5*b^2*c^1
2*e^3*i - 1648128*a^5*b^3*c^11*f^3*h + 1330560*a^8*b^4*c^7*d*j^3 + 1179648*
a^7*b^2*c^10*e*i^3 - 898560*a^6*b^3*c^10*f*h^3 - 826560*a^7*b^6*c^6*d*j^3 -
607068*a^2*b^8*c^9*d^3*j + 589824*a^6*b^4*c^9*e*i^3 - 354240*a^5*b^5*c^9*f
*h^3 - 354240*a^4*b^5*c^10*f^3*h + 145188*a^6*b^8*c^5*d*j^3 + 98304*a^5*b^6
*c^8*e*i^3 + 43680*a^3*b^7*c^9*f^3*h - 21600*a^4*b^7*c^8*f*h^3 - 9576*a^5*b
^10*c^4*d*j^3 + 1350*a^3*b^9*c^7*f*h^3 - 1050*a^2*b^9*c^8*f^3*h - 504*a*b^1
4*c^4*d^2*j^2 + 210*a^4*b^12*c^3*d*j^3 + 3870720*a^6*b*c^12*d^2*i^2 + 16588
80*a^6*b*c^12*e^2*h^2 - 9792*a*b^11*c^7*d^2*i^2 + 16547328*a^4*b^2*c^13*d^3
*h - 12306816*a^3*b^4*c^12*d^3*h + 37310976*a^3*b^3*c^13*d^3*f + 3037824*a^
2*b^6*c^11*d^3*h - 2654208*a^5*b^3*c^11*e*g^3 + 1949184*a^6*b^2*c^11*d*h^3
+ 1296000*a^5*b^4*c^10*d*h^3 - 155520*a^4*b^6*c^9*d*h^3 - 40500*a*b^10*c^8*
d^2*h^2 - 8100*a^3*b^8*c^8*d*h^3 + 4050*a^2*b^10*c^7*d*h^3 + 3870720*a^5*b*
c^13*e^2*f^2 + 34836480*a^4*b*c^14*d^2*e^2 - 108864*a*b^9*c^9*d^2*g^2 - 806
8032*a^2*b^5*c^12*d^3*f - 5623296*a^4*b^3*c^12*d*f^3 + 1737792*a^3*b^5*c^11
*d*f^3 - 260190*a*b^8*c^10*d^2*f^2 - 211680*a^2*b^7*c^10*d*f^3 - 435456*a*b
^7*c^11*d^2*e^2 - 377487360*a^12*b*c^6*e*k^3 + 1434977280*a^8*b^3*c^8*d^2*k
^2 + 173408256*a^7*c^12*d^2*e*k + 3276800*a^12*c^7*i*j^2*k - 125829120*a^13
*b*c^5*i*k^3 + 26214400*a^12*c^7*f*j*k^2 + 1179648*a^10*c^9*h^2*i*k + 13440
*a^6*b^13*h*j*k^2 + 50331648*a^11*c^8*e*i*k^2 + 110100480*a^10*c^9*d*f*k^2
+ 57802752*a^8*c^11*d^2*i*k + 9830400*a^11*c^8*e*j^2*k - 3276800*a^9*c^10*f
^2*i*k + 4480*a^5*b^14*f*j*k^2 + 15728640*a^11*c^8*f*h*k^2 - 409600*a^9*c^1
0*f*i^2*j - 1152*b^16*c^3*d^2*i*k - 1220516352*a^7*b^5*c^7*d^2*k^2 + 353894
4*a^9*c^10*e*h^2*k + 384000*a^8*c^11*f^2*h*j + 13440*a^4*b^15*d*j*k^2 + 384
*a^3*b^16*f*h*k^2 + 20321280*a^7*c^12*d^2*h*j - 245760*a^8*c^11*f*h*i^2 + 3
456*b^15*c^4*d^2*g*k - 270*b^14*c^5*d^2*h*j - 9830400*a^8*c^11*e*f^2*k + 48
38400*a^9*c^10*d*h*j^2 + 2903040*a^8*c^11*d*h^2*j - 1966080*a^10*b*c^8*i^3*
k + 1433600*a^9*b^9*c*i*k^3 + 1152*a^2*b^17*d*h*k^2 - 3686400*a^7*c^12*e^2*
f*j - 53084160*a^7*b*c^11*e^3*k - 6912*b^14*c^5*d^2*e*k - 3456*b^12*c^7*d^2
*g*i + 630*b^13*c^6*d^2*f*j + 2688000*a^7*c^12*d*f^2*j + 245760*a^8*b^10*c*
g*k^3 - 2211840*a^6*c^13*e^2*f*h - 1720320*a^7*c^12*d*f*i^2 - 9450*b^11*c^8
*d^2*f*h + 6912*b^11*c^8*d^2*e*i + 1612800*a^6*c^13*d*f^2*h - 1344000*a^10*
b*c^8*f*j^3 - 1344000*a^7*b*c^11*f^3*j - 393216*a^8*b*c^10*g*i^3 - 23616*a*
b^17*c*d^2*k^2 - 20736*b^10*c^9*d^2*e*g - 75188736*a^4*b*c^14*d^3*f - 88320
0*a^6*b*c^12*f^3*h - 317952*a^7*b*c^11*f*h^3 + 43416*a*b^10*c^8*d^3*j - 154
82880*a^5*c^14*d*e^2*f - 10616832*a^5*b*c^13*e^3*g - 345060*a*b^8*c^10*d^3*
h - 4262400*a^5*b*c^13*d*f^3 + 852768*a*b^7*c^11*d^3*f + 7350*a*b^9*c^9*d*f
^3 + 584578368*a^6*b^7*c^6*d^2*k^2 + 93905920*a^12*b^3*c^4*j^2*k^2 - 177997
248*a^5*b^9*c^5*d^2*k^2 - 50967040*a^11*b^5*c^3*j^2*k^2 + 104693760*a^9*b^2
*c^8*e^2*k^2 + 12849984*a^10*b^7*c^2*j^2*k^2 + 20021248*a^11*b^2*c^6*i^2*k^
2 - 85524480*a^8*b^4*c^7*e^2*k^2 + 33223680*a^10*b^3*c^6*h^2*k^2 + 4227072*
a^10*b^4*c^5*i^2*k^2 - 3973120*a^9*b^6*c^4*i^2*k^2 + 344064*a^7*b^10*c^2*i^
2*k^2 - 81920*a^8*b^8*c^3*i^2*k^2 - 11386368*a^9*b^5*c^5*h^2*k^2 + 26173440
*a^9*b^4*c^6*g^2*k^2 - 21381120*a^8*b^6*c^5*g^2*k^2 + 18874368*a^10*b^2*c^7
*g^2*k^2 + 501760*a^9*b^3*c^7*i^2*j^2 + 452160*a^8*b^7*c^4*h^2*k^2 + 385920

*a^7*b^9*c^3*h^2*k^2 + 170240*a^8*b^5*c^6*i^2*j^2 - 48960*a^6*b^11*c^2*h^2*
k^2 + 9216*a^7*b^7*c^5*i^2*j^2 - 1984*a^6*b^9*c^4*i^2*j^2 + 64*a^5*b^11*c^3
*i^2*j^2 + 5898240*a^7*b^8*c^4*g^2*k^2 + 1419840*a^8*b^4*c^7*h^2*j^2 + 1387
008*a^9*b^2*c^8*h^2*j^2 - 737280*a^6*b^10*c^3*g^2*k^2 + 84960*a^7*b^6*c^6*h
^2*j^2 + 36864*a^5*b^12*c^2*g^2*k^2 - 8010*a^6*b^8*c^5*h^2*j^2 - 180*a^5*b^
10*c^4*h^2*j^2 + 9*a^4*b^12*c^3*h^2*j^2 + 14115840*a^9*b^3*c^7*f^2*k^2 - 92
31552*a^7*b^7*c^5*f^2*k^2 + 23592960*a^7*b^6*c^6*e^2*k^2 + 4984320*a^8*b^5*
c^6*f^2*k^2 + 3759040*a^6*b^9*c^4*f^2*k^2 + 36190080*a^4*b^11*c^4*d^2*k^2 +
 967680*a^8*b^3*c^8*g^2*j^2 - 727360*a^5*b^11*c^3*f^2*k^2 + 276480*a^7*b^3*
c^9*h^2*i^2 + 161280*a^7*b^5*c^7*g^2*j^2 + 140544*a^6*b^5*c^8*h^2*i^2 + 729
60*a^4*b^13*c^2*f^2*k^2 + 25344*a^5*b^7*c^7*h^2*i^2 - 20160*a^6*b^7*c^6*g^2
*j^2 + 576*a^5*b^9*c^5*g^2*j^2 + 576*a^4*b^9*c^6*h^2*i^2 + 3808000*a^8*b^2*
c^9*f^2*j^2 - 2949120*a^6*b^8*c^5*e^2*k^2 + 1643712*a^7*b^4*c^8*f^2*j^2 + 8
84736*a^7*b^2*c^10*g^2*i^2 + 884736*a^6*b^4*c^9*g^2*i^2 + 221184*a^5*b^6*c^
8*g^2*i^2 + 147456*a^5*b^10*c^4*e^2*k^2 - 125440*a^6*b^6*c^7*f^2*j^2 - 1379
0*a^5*b^8*c^6*f^2*j^2 + 1785*a^4*b^10*c^5*f^2*j^2 - 70*a^3*b^12*c^4*f^2*j^2
 - 4953600*a^3*b^13*c^3*d^2*k^2 + 18427392*a^7*b^2*c^10*d^2*j^2 + 645120*a^
7*b^3*c^9*e^2*j^2 + 501760*a^6*b^3*c^10*f^2*i^2 + 442944*a^2*b^15*c^2*d^2*k
^2 + 414720*a^6*b^3*c^10*g^2*h^2 + 207360*a^5*b^5*c^9*g^2*h^2 + 170240*a^5*
b^5*c^9*f^2*i^2 - 80640*a^6*b^5*c^8*e^2*j^2 + 9216*a^4*b^7*c^8*f^2*i^2 + 51
84*a^4*b^7*c^8*g^2*h^2 + 2304*a^5*b^7*c^7*e^2*j^2 - 1984*a^3*b^9*c^7*f^2*i^
2 + 64*a^2*b^11*c^6*f^2*i^2 - 4148928*a^6*b^4*c^9*d^2*j^2 + 3538944*a^6*b^2
*c^11*e^2*i^2 + 1684224*a^6*b^2*c^11*f^2*h^2 + 1264320*a^5*b^4*c^10*f^2*h^2
 - 1183392*a^5*b^6*c^8*d^2*j^2 + 884736*a^5*b^4*c^10*e^2*i^2 + 645750*a^4*b
^8*c^7*d^2*j^2 + 126720*a^4*b^6*c^9*f^2*h^2 - 115920*a^3*b^10*c^6*d^2*j^2 -
 13950*a^3*b^8*c^8*f^2*h^2 + 10836*a^2*b^12*c^5*d^2*j^2 + 225*a^2*b^10*c^7*
f^2*h^2 + 1935360*a^5*b^3*c^11*d^2*i^2 + 967680*a^5*b^3*c^11*f^2*g^2 + 8294
40*a^5*b^3*c^11*e^2*h^2 - 532224*a^4*b^5*c^10*d^2*i^2 + 161280*a^4*b^5*c^10
*f^2*g^2 - 96768*a^3*b^7*c^9*d^2*i^2 + 62784*a^2*b^9*c^8*d^2*i^2 + 20736*a^
4*b^5*c^10*e^2*h^2 - 20160*a^3*b^7*c^9*f^2*g^2 + 576*a^2*b^9*c^8*f^2*g^2 +
11487744*a^5*b^2*c^12*d^2*h^2 + 7962624*a^5*b^2*c^12*e^2*g^2 + 35525376*a^4
*b^2*c^13*d^2*f^2 - 1412640*a^3*b^6*c^10*d^2*h^2 + 461376*a^4*b^4*c^11*d^2*
h^2 + 375030*a^2*b^8*c^9*d^2*h^2 + 8709120*a^4*b^3*c^12*d^2*g^2 - 4354560*a
^3*b^5*c^11*d^2*g^2 + 979776*a^2*b^7*c^10*d^2*g^2 + 645120*a^4*b^3*c^12*e^2
*f^2 - 80640*a^3*b^5*c^11*e^2*f^2 + 2304*a^2*b^7*c^10*e^2*f^2 - 15269184*a^
3*b^4*c^12*d^2*f^2 + 2870784*a^2*b^6*c^11*d^2*f^2 - 17418240*a^3*b^3*c^13*d
^2*e^2 + 3919104*a^2*b^5*c^12*d^2*e^2 + 384*a*b^18*d*f*k^2 - 199229440*a^14
*b^2*c^3*k^4 + 8388608*a^12*c^7*i^2*k^2 + 75497472*a^10*c^9*e^2*k^2 + 78400
*a^8*b^11*j^2*k^2 + 4096*a^5*b^14*i^2*k^2 + 345600*a^10*c^9*h^2*j^2 + 576*a
^4*b^15*h^2*k^2 + 57937920*a^13*b^4*c^2*k^4 + 320000*a^9*c^10*f^2*j^2 + 64*
a^2*b^17*f^2*k^2 + 16934400*a^8*c^11*d^2*j^2 + 9*b^16*c^3*d^2*j^2 + 3538944
*a^7*c^12*e^2*i^2 + 115200*a^7*c^12*f^2*h^2 + 576*b^13*c^6*d^2*i^2 + 2025*b
^12*c^7*d^2*h^2 + 6096384*a^6*c^13*d^2*h^2 + 492800*a^11*b^2*c^6*j^4 + 3514
56*a^10*b^4*c^5*j^4 - 43120*a^9*b^6*c^4*j^4 + 5184*b^11*c^8*d^2*g^2 + 1225*
a^8*b^8*c^3*j^4 + 131072*a^8*b^2*c^9*i^4 + 98304*a^7*b^4*c^8*i^4 + 32768*a^

```
6*b^6*c^7*i^4 + 11025*b^10*c^9*d^2*f^2 + 4096*a^5*b^8*c^6*i^4 + 5644800*a^5
*c^14*d^2*f^2 + 142560*a^6*b^4*c^9*h^4 + 103680*a^7*b^2*c^10*h^4 + 32400*a^
5*b^6*c^8*h^4 + 20736*b^9*c^10*d^2*e^2 + 2025*a^4*b^8*c^7*h^4 + 331776*a^5*
b^4*c^10*g^4 + 492800*a^5*b^2*c^12*f^4 + 351456*a^4*b^4*c^11*f^4 - 43120*a^
3*b^6*c^10*f^4 + 1225*a^2*b^8*c^9*f^4 - 27433728*a^3*b^2*c^14*d^4 + 6446304
*a^2*b^4*c^13*d^4 + a^2*b^14*c^3*f^2*j^2 - 81920*a^8*b^11*i*k^3 + 384000*a^
11*c^8*h*j^3 + 138240*a^9*c^10*h^3*j + 47416320*a^6*c^13*d^3*j - 1134*b^12*
c^7*d^3*j + 7077888*a^6*c^13*e^3*i + 2688000*a^10*c^9*d*j^3 + 786432*a^8*c^
11*e*i^3 + 28449792*a^5*c^14*d^3*h - 7782400*a^12*b^6*c*k^4 + 17010*b^10*c^
9*d^3*h + 580608*a^7*c^12*d*h^3 - 39690*b^9*c^10*d^3*f - 734832*a*b^6*c^12*
d^4 + 268435456*a^15*c^4*k^4 + 576*b^19*d^2*k^2 + 409600*a^11*b^8*k^4 + 160
000*a^12*c^7*j^4 + 65536*a^9*c^10*i^4 + 20736*a^8*c^11*h^4 + 49787136*a^4*c
^15*d^4 + 160000*a^6*c^13*f^4 + 5308416*a^5*c^14*e^4 + 35721*b^8*c^11*d^4,
z, n), n, 1, 4)
```

3.60 $\int (a+bx^2+cx^4)^3 (ad+aex+(bd+af)x^2+bex^3+$

Optimal result

Integrand size = 63, antiderivative size = 416

$$\begin{aligned}
&\int (a+bx^2+cx^4)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx \\
&= a^4dx+\frac{1}{2}a^4ex^2+\frac{1}{3}a^3(4bd+af)x^3+a^3bex^4+\frac{2}{5}a^2\left(3b^2d+2acd+2abf\right)x^5 \\
&\quad +\frac{1}{3}a^2\left(3b^2+2ac\right)ex^6+\frac{2}{7}a\left(2b^3d+6abcd+3ab^2f+2a^2cf\right)x^7+\frac{1}{2}ab\left(b^2+3ac\right)ex^8 \\
&\quad +\frac{1}{9}\left(b^4d+12ab^2cd+6a^2c^2d+4ab^3f+12a^2bcf\right)x^9+\frac{1}{10}\left(b^4+12ab^2c+6a^2c^2\right)ex^{10} \\
&\quad +\frac{1}{11}\left(4b^3cd+12abc^2d+b^4f+12ab^2cf+6a^2c^2f\right)x^{11}+\frac{1}{3}bc\left(b^2+3ac\right)ex^{12} \\
&\quad +\frac{2}{13}c\left(3b^2cd+2ac^2d+2b^3f+6abcf\right)x^{13}+\frac{1}{7}c^2\left(3b^2+2ac\right)ex^{14} \\
&\quad +\frac{2}{15}c^2\left(2bcd+3b^2f+2acf\right)x^{15}+\frac{1}{4}bc^3ex^{16}+\frac{1}{17}c^3(cd+4bf)x^{17}+\frac{1}{18}c^4ex^{18}+\frac{1}{19}c^4fx^{19}
\end{aligned}$$

```
[Out] a^4*d*x+1/2*a^4*e*x^2+1/3*a^3*(a*f+4*b*d)*x^3+a^3*b*e*x^4+2/5*a^2*(2*a*b*f+
2*a*c*d+3*b^2*d)*x^5+1/3*a^2*(2*a*c+3*b^2)*e*x^6+2/7*a*(2*a^2*c*f+3*a*b^2*f
+6*a*b*c*d+2*b^3*d)*x^7+1/2*a*b*(3*a*c+b^2)*e*x^8+1/9*(12*a^2*b*c*f+6*a^2*c
^2*d+4*a*b^3*f+12*a*b^2*c*d+b^4*d)*x^9+1/10*(6*a^2*c^2+12*a*b^2*c+b^4)*e*x^
10+1/11*(6*a^2*c^2*f+12*a*b^2*c*f+12*a*b*c^2*d+b^4*f+4*b^3*c*d)*x^11+1/3*b*
c*(3*a*c+b^2)*e*x^12+2/13*c*(6*a*b*c*f+2*a*c^2*d+2*b^3*f+3*b^2*c*d)*x^13+1/
7*c^2*(2*a*c+3*b^2)*e*x^14+2/15*c^2*(2*a*c*f+3*b^2*f+2*b*c*d)*x^15+1/4*b*c^
3*e*x^16+1/17*c^3*(4*b*f+c*d)*x^17+1/18*c^4*e*x^18+1/19*c^4*f*x^19
```

Rubi [A] (verified)

Time = 0.41 (sec) , antiderivative size = 416, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.016, Rules used = {1685}

$$\int (a+bx^2+cx^4)^3\left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$= a^4dx+\frac{1}{2}a^4ex^2+\frac{1}{3}a^3x^3(af+4bd)+a^3bex^4+\frac{2}{5}a^2x^5\left(2abf+2acd+3b^2d\right)$$
$$+\frac{1}{3}a^2ex^6\left(2ac+3b^2\right)+\frac{1}{10}ex^{10}\left(6a^2c^2+12ab^2c+b^4\right)+\frac{2}{7}ax^7\left(2a^2cf+3ab^2f+6abcd+2b^3d\right)$$
$$+\frac{1}{11}x^{11}\left(6a^2c^2f+12ab^2cf+12abc^2d+b^4f+4b^3cd\right)$$
$$+\frac{1}{9}x^9\left(12a^2bcf+6a^2c^2d+4ab^3f+12ab^2cd+b^4d\right)$$
$$+\frac{2}{15}c^2x^{15}\left(2acf+3b^2f+2bcd\right)+\frac{1}{7}c^2ex^{14}\left(2ac+3b^2\right)+\frac{1}{3}bcex^{12}\left(3ac+b^2\right)$$
$$+\frac{1}{2}abex^8\left(3ac+b^2\right)+\frac{2}{13}cx^{13}\left(6abcf+2ac^2d+2b^3f+3b^2cd\right)$$
$$+\frac{1}{17}c^3x^{17}(4bf+cd)+\frac{1}{4}bc^3ex^{16}+\frac{1}{18}c^4ex^{18}+\frac{1}{19}c^4fx^{19}$$

```
[In] Int[(a + b*x^2 + c*x^4)^3*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d +
 b*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^4*d*x + (a^4*e*x^2)/2 + (a^3*(4*b*d + a*f)*x^3)/3 + a^3*b*e*x^4 + (2*a^2*
(3*b^2*d + 2*a*c*d + 2*a*b*f)*x^5)/5 + (a^2*(3*b^2 + 2*a*c)*e*x^6)/3 + (2*a
*(2*b^3*d + 6*a*b*c*d + 3*a*b^2*f + 2*a^2*c*f)*x^7)/7 + (a*b*(b^2 + 3*a*c)*
e*x^8)/2 + ((b^4*d + 12*a*b^2*c*d + 6*a^2*c^2*d + 4*a*b^3*f + 12*a^2*b*c*f)
*x^9)/9 + ((b^4 + 12*a*b^2*c + 6*a^2*c^2)*e*x^10)/10 + ((4*b^3*c*d + 12*a*b
*c^2*d + b^4*f + 12*a*b^2*c*f + 6*a^2*c^2*f)*x^11)/11 + (b*c*(b^2 + 3*a*c)*
e*x^12)/3 + (2*c*(3*b^2*c*d + 2*a*c^2*d + 2*b^3*f + 6*a*b*c*f)*x^13)/13 + (
c^2*(3*b^2 + 2*a*c)*e*x^14)/7 + (2*c^2*(2*b*c*d + 3*b^2*f + 2*a*c*f)*x^15)/
15 + (b*c^3*e*x^16)/4 + (c^3*(c*d + 4*b*f)*x^17)/17 + (c^4*e*x^18)/18 + (c^
4*f*x^19)/19
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} = \int \Big(& a^4d + a^4ex + a^3(4bd + af)x^2 + 4a^3bex^3 + 2a^2\left(3b^2d + 2acd + 2abf\right)x^4 \\ &+ 2a^2\left(3b^2 + 2ac\right)ex^5 + 2a\left(2b^3d + 6abcd + 3ab^2f + 2a^2cf\right)x^6 + 4ab\left(b^2 + 3ac\right)ex^7 \\ &+ \left(b^4d + 12ab^2cd + 6a^2c^2d + 4ab^3f + 12a^2bcf\right)x^8 + \left(b^4 + 12ab^2c + 6a^2c^2\right)ex^9 \\ &+ \left(4b^3cd + 12abc^2d + b^4f + 12ab^2cf + 6a^2c^2f\right)x^{10} + 4bc\left(b^2 + 3ac\right)ex^{11} \\ &+ 2c\left(3b^2cd + 2ac^2d + 2b^3f + 6abcf\right)x^{12} + 2c^2\left(3b^2 + 2ac\right)ex^{13} \\ &+ 2c^2\left(2bcd + 3b^2f + 2acf\right)x^{14} + 4bc^3ex^{15} + c^3(cd + 4bf)x^{16} + c^4ex^{17} + c^4fx^{18}\Big)\,dx \\ =\ & a^4dx + \frac{1}{2}a^4ex^2 + \frac{1}{3}a^3(4bd + af)x^3 + a^3bex^4 + \frac{2}{5}a^2\left(3b^2d + 2acd + 2abf\right)x^5 \\ &+ \frac{1}{3}a^2\left(3b^2 + 2ac\right)ex^6 + \frac{2}{7}a\left(2b^3d + 6abcd + 3ab^2f + 2a^2cf\right)x^7 \\ &+ \frac{1}{2}ab\left(b^2 + 3ac\right)ex^8 + \frac{1}{9}\left(b^4d + 12ab^2cd + 6a^2c^2d + 4ab^3f + 12a^2bcf\right)x^9 \\ &+ \frac{1}{10}\left(b^4 + 12ab^2c + 6a^2c^2\right)ex^{10} + \frac{1}{11}\left(4b^3cd + 12abc^2d + b^4f + 12ab^2cf + 6a^2c^2f\right)x^{11} \\ &+ \frac{1}{3}bc\left(b^2 + 3ac\right)ex^{12} + \frac{2}{13}c\left(3b^2cd + 2ac^2d + 2b^3f + 6abcf\right)x^{13} \\ &+ \frac{1}{7}c^2\left(3b^2 + 2ac\right)ex^{14} + \frac{2}{15}c^2\left(2bcd + 3b^2f + 2acf\right)x^{15} \\ &+ \frac{1}{4}bc^3ex^{16} + \frac{1}{17}c^3(cd + 4bf)x^{17} + \frac{1}{18}c^4ex^{18} + \frac{1}{19}c^4fx^{19}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.08 (sec) , antiderivative size = 416, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \left(a + bx^2 + cx^4\right)^3\left(ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6\right)dx \\ &= a^4dx + \frac{1}{2}a^4ex^2 + \frac{1}{3}a^3(4bd + af)x^3 + a^3bex^4 + \frac{2}{5}a^2\left(3b^2d + 2acd + 2abf\right)x^5 \\ &+ \frac{1}{3}a^2\left(3b^2 + 2ac\right)ex^6 + \frac{2}{7}a\left(2b^3d + 6abcd + 3ab^2f + 2a^2cf\right)x^7 + \frac{1}{2}ab\left(b^2 + 3ac\right)ex^8 \\ &+ \frac{1}{9}\left(b^4d + 12ab^2cd + 6a^2c^2d + 4ab^3f + 12a^2bcf\right)x^9 + \frac{1}{10}\left(b^4 + 12ab^2c + 6a^2c^2\right)ex^{10} \\ &+ \frac{1}{11}\left(4b^3cd + 12abc^2d + b^4f + 12ab^2cf + 6a^2c^2f\right)x^{11} + \frac{1}{3}bc\left(b^2 + 3ac\right)ex^{12} \\ &+ \frac{2}{13}c\left(3b^2cd + 2ac^2d + 2b^3f + 6abcf\right)x^{13} + \frac{1}{7}c^2\left(3b^2 + 2ac\right)ex^{14} \\ &+ \frac{2}{15}c^2\left(2bcd + 3b^2f + 2acf\right)x^{15} + \frac{1}{4}bc^3ex^{16} + \frac{1}{17}c^3(cd + 4bf)x^{17} + \frac{1}{18}c^4ex^{18} + \frac{1}{19}c^4fx^{19}\end{aligned}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)^3*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 +
(c*d + b*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^4*d*x + (a^4*e*x^2)/2 + (a^3*(4*b*d + a*f)*x^3)/3 + a^3*b*e*x^4 + (2*a^2*
(3*b^2*d + 2*a*c*d + 2*a*b*f)*x^5)/5 + (a^2*(3*b^2 + 2*a*c)*e*x^6)/3 + (2*a
*(2*b^3*d + 6*a*b*c*d + 3*a*b^2*f + 2*a^2*c*f)*x^7)/7 + (a*b*(b^2 + 3*a*c)*
e*x^8)/2 + ((b^4*d + 12*a*b^2*c*d + 6*a^2*c^2*d + 4*a*b^3*f + 12*a^2*b*c*f)
*x^9)/9 + ((b^4 + 12*a*b^2*c + 6*a^2*c^2)*e*x^10)/10 + ((4*b^3*c*d + 12*a*b
*c^2*d + b^4*f + 12*a*b^2*c*f + 6*a^2*c^2*f)*x^11)/11 + (b*c*(b^2 + 3*a*c)*
e*x^12)/3 + (2*c*(3*b^2*c*d + 2*a*c^2*d + 2*b^3*f + 6*a*b*c*f)*x^13)/13 + (
c^2*(3*b^2 + 2*a*c)*e*x^14)/7 + (2*c^2*(2*b*c*d + 3*b^2*f + 2*a*c*f)*x^15)/
15 + (b*c^3*e*x^16)/4 + (c^3*(c*d + 4*b*f)*x^17)/17 + (c^4*e*x^18)/18 + (c^
4*f*x^19)/19
```

Maple [A] (verified)

Time = 28.81 (sec) , antiderivative size = 412, normalized size of antiderivative = 0.99

method	result
norman	$\left(\frac{1}{3} f\, a^{4}+\frac{4}{3} d\, a^{3} b\right) x^{3}+\left(\frac{2}{7} a\, c^{3} e+\frac{3}{7} b^{2} c^{2} e\right) x^{14}+\left(\frac{2}{3} a^{3} c e+a^{2} b^{2} e\right) x^{6}+\left(\frac{4}{17} b\, c^{3} f+\frac{1}{17} c^{4} d\right) x^{17}+($
risch	$\frac{1}{2} a^{4} e\, x^{2}+\frac{1}{18} c^{4} e\, x^{18}+\frac{1}{19} c^{4} f\, x^{19}+\frac{2}{3} x^{6} a^{3} c e+x^{6} a^{2} b^{2} e+\frac{4}{5} x^{5} f\, a^{3} b+\frac{4}{5} x^{5} a^{3} c d+\frac{6}{5} x^{5} a^{2} b^{2} d+\frac{4}{3} x^{}$
parallelrisch	$\frac{1}{2} a^{4} e\, x^{2}+\frac{1}{18} c^{4} e\, x^{18}+\frac{1}{19} c^{4} f\, x^{19}+\frac{2}{3} x^{6} a^{3} c e+x^{6} a^{2} b^{2} e+\frac{4}{5} x^{5} f\, a^{3} b+\frac{4}{5} x^{5} a^{3} c d+\frac{6}{5} x^{5} a^{2} b^{2} d+\frac{4}{3} x^{}$
gosper	$x(3063060 f\, c^{4} x^{18}+3233230 c^{4} e\, x^{17}+13693680 b\, c^{3} f\, x^{16}+3423420 c^{4} d\, x^{16}+145495535 b\, c^{3} e\, x^{15}+15519504 a\, c^{3} f\, x^{14}+23279256 b^{2} c$
default	$\frac{c^{4} f\, x^{19}}{19}+\frac{c^{4} e\, x^{18}}{18}+\frac{\left(3 b\, c^{3} f+c^{3}(b f+c d)\right) x^{17}}{17}+\frac{b\, c^{3} e\, x^{16}}{4}+\frac{\left(\left(a\, c^{2}+2 b^{2} c+c\left(2 a c+b^{2}\right)\right) c f+3 b\, c^{2}(b f+c d)+c^{3}(a f+b d)\right) x^{15}}{15}$

```
[In] int((c*x^4+b*x^2+a)^3*(d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^
5+c*f*x^6),x,method=_RETURNVERBOSE)
```

```
[Out] (1/3*f*a^4+4/3*d*a^3*b)*x^3+(2/7*a*c^3*e+3/7*b^2*c^2*e)*x^14+(2/3*a^3*c*e+a
^2*b^2*e)*x^6+(4/17*b*c^3*f+1/17*c^4*d)*x^17+(a*b*c^2*e+1/3*b^3*c*e)*x^12+(
3/2*a^2*b*c*e+1/2*a*b^3*e)*x^8+(4/15*a*c^3*f+2/5*b^2*c^2*f+4/15*b*c^3*d)*x^
15+(3/5*a^2*c^2*e+6/5*a*b^2*c*e+1/10*b^4*e)*x^10+(4/5*f*a^3*b+4/5*a^3*c*d+6
/5*a^2*b^2*d)*x^5+(4/7*a^3*c*f+6/7*a^2*b^2*f+12/7*a^2*b*c*d+4/7*a*b^3*d)*x^
7+(12/13*a*b*c^2*f+4/13*a*c^3*d+4/13*b^3*c*f+6/13*b^2*c^2*d)*x^13+(6/11*a^2
*c^2*f+12/11*a*b^2*c*f+12/11*a*b*c^2*d+1/11*b^4*f+4/11*b^3*c*d)*x^11+(4/3*a
^2*b*c*f+2/3*a^2*c^2*d+4/9*a*b^3*f+4/3*a*b^2*c*d+1/9*d*b^4)*x^9+a^4*d*x+a^3
*b*e*x^4+1/2*a^4*e*x^2+1/18*c^4*e*x^18+1/19*c^4*f*x^19+1/4*b*c^3*e*x^16
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 418, normalized size of antiderivative = 1.00

$$\int (a+bx^2+cx^4)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx = \frac{1}{19}c^4fx^{19}+\frac{1}{18}c^4ex^{18}+\frac{1}{4}bc^3ex^{16}+\frac{1}{17}(c^4d+4bc^3f)x^{17}+\frac{1}{7}(3b^2c^2+2ac^3)ex^{14} + \frac{2}{15}(2bc^3d+(3b^2c^2+2ac^3)f)x^{15}+\frac{1}{3}(b^3c+3abc^2)ex^{12} + \frac{2}{13}((3b^2c^2+2ac^3)d+2(b^3c+3abc^2)f)x^{13}+\frac{1}{10}(b^4+12ab^2c+6a^2c^2)ex^{10} + \frac{1}{11}(4(b^3c+3abc^2)d+(b^4+12ab^2c+6a^2c^2)f)x^{11}+\frac{1}{2}(ab^3+3a^2bc)ex^8 + \frac{1}{9}((b^4+12ab^2c+6a^2c^2)d+4(ab^3+3a^2bc)f)x^9+a^3bex^4 + \frac{1}{3}(3a^2b^2+2a^3c)ex^6+\frac{2}{7}(2(ab^3+3a^2bc)d+(3a^2b^2+2a^3c)f)x^7 + \frac{1}{2}a^4ex^2+a^4dx+\frac{2}{5}(2a^3bf+(3a^2b^2+2a^3c)d)x^5+\frac{1}{3}(4a^3bd+a^4f)x^3$$

```
[In] integrate((c*x^4+b*x^2+a)^3*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="fricas")
```

```
[Out] 1/19*c^4*f*x^19 + 1/18*c^4*e*x^18 + 1/4*b*c^3*e*x^16 + 1/17*(c^4*d + 4*b*c^
3*f)*x^17 + 1/7*(3*b^2*c^2 + 2*a*c^3)*e*x^14 + 2/15*(2*b*c^3*d + (3*b^2*c^2
 + 2*a*c^3)*f)*x^15 + 1/3*(b^3*c + 3*a*b*c^2)*e*x^12 + 2/13*((3*b^2*c^2 + 2
*a*c^3)*d + 2*(b^3*c + 3*a*b*c^2)*f)*x^13 + 1/10*(b^4 + 12*a*b^2*c + 6*a^2*
c^2)*e*x^10 + 1/11*(4*(b^3*c + 3*a*b*c^2)*d + (b^4 + 12*a*b^2*c + 6*a^2*c^2
)*f)*x^11 + 1/2*(a*b^3 + 3*a^2*b*c)*e*x^8 + 1/9*((b^4 + 12*a*b^2*c + 6*a^2*
c^2)*d + 4*(a*b^3 + 3*a^2*b*c)*f)*x^9 + a^3*b*e*x^4 + 1/3*(3*a^2*b^2 + 2*a^
3*c)*e*x^6 + 2/7*(2*(a*b^3 + 3*a^2*b*c)*d + (3*a^2*b^2 + 2*a^3*c)*f)*x^7 +
1/2*a^4*e*x^2 + a^4*d*x + 2/5*(2*a^3*b*f + (3*a^2*b^2 + 2*a^3*c)*d)*x^5 + 1
/3*(4*a^3*b*d + a^4*f)*x^3
```

Sympy [A] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 503, normalized size of antiderivative = 1.21

$$\int \left(a+bx^2+cx^4\right)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= a^4dx + \frac{a^4ex^2}{2} + a^3bex^4 + \frac{bc^3ex^{16}}{4} + \frac{c^4ex^{18}}{18} + \frac{c^4fx^{19}}{19} + x^{17}\cdot\left(\frac{4bc^3f}{17} + \frac{c^4d}{17}\right) + x^{15}\cdot\left(\frac{4ac^3f}{15} + \frac{2b^2c^2f}{5} + \frac{4bc^3d}{15}\right) + x^{14}\cdot\left(\frac{2ac^3e}{7} + \frac{3b^2c^2e}{7}\right) + x^{13}\cdot\left(\frac{12abc^2f}{13} + \frac{4ac^3d}{13} + \frac{4b^3cf}{13} + \frac{6b^2c^2d}{13}\right) + x^{12}\left(abc^2e + \frac{b^3ce}{3}\right) + x^{11}\cdot\left(\frac{6a^2c^2f}{11} + \frac{12ab^2cf}{11} + \frac{12abc^2d}{11} + \frac{b^4f}{11} + \frac{4b^3cd}{11}\right) + x^{10}\cdot\left(\frac{3a^2c^2e}{5} + \frac{6ab^2ce}{5} + \frac{b^4e}{10}\right) + x^9\cdot\left(\frac{4a^2bcf}{3} + \frac{2a^2c^2d}{3} + \frac{4ab^3f}{9} + \frac{4ab^2cd}{3} + \frac{b^4d}{9}\right) + x^8\cdot\left(\frac{3a^2bce}{2} + \frac{ab^3e}{2}\right) + x^7\cdot\left(\frac{4a^3cf}{7} + \frac{6a^2b^2f}{7} + \frac{12a^2bcd}{7} + \frac{4ab^3d}{7}\right) + x^6\cdot\left(\frac{2a^3ce}{3} + a^2b^2e\right) + x^5\cdot\left(\frac{4a^3bf}{5} + \frac{4a^3cd}{5} + \frac{6a^2b^2d}{5}\right) + x^3\left(\frac{a^4f}{3} + \frac{4a^3bd}{3}\right)$$

```
[In] integrate((c*x**4+b*x**2+a)**3*(a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)
*x**4+c*e*x**5+c*f*x**6),x)
```

```
[Out] a**4*d*x + a**4*e*x**2/2 + a**3*b*e*x**4 + b*c**3*e*x**16/4 + c**4*e*x**18/
18 + c**4*f*x**19/19 + x**17*(4*b*c**3*f/17 + c**4*d/17) + x**15*(4*a*c**3*
f/15 + 2*b**2*c**2*f/5 + 4*b*c**3*d/15) + x**14*(2*a*c**3*e/7 + 3*b**2*c**2
*e/7) + x**13*(12*a*b*c**2*f/13 + 4*a*c**3*d/13 + 4*b**3*c*f/13 + 6*b**2*c*
*2*d/13) + x**12*(a*b*c**2*e + b**3*c*e/3) + x**11*(6*a**2*c**2*f/11 + 12*a
*b**2*c*f/11 + 12*a*b*c**2*d/11 + b**4*f/11 + 4*b**3*c*d/11) + x**10*(3*a**
2*c**2*e/5 + 6*a*b**2*c*e/5 + b**4*e/10) + x**9*(4*a**2*b*c*f/3 + 2*a**2*c*
*2*d/3 + 4*a*b**3*f/9 + 4*a*b**2*c*d/3 + b**4*d/9) + x**8*(3*a**2*b*c*e/2 +
 a*b**3*e/2) + x**7*(4*a**3*c*f/7 + 6*a**2*b**2*f/7 + 12*a**2*b*c*d/7 + 4*a
*b**3*d/7) + x**6*(2*a**3*c*e/3 + a**2*b**2*e) + x**5*(4*a**3*b*f/5 + 4*a**
3*c*d/5 + 6*a**2*b**2*d/5) + x**3*(a**4*f/3 + 4*a**3*b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 418, normalized size of antiderivative = 1.00

$$\int (a+bx^2+cx^4)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= \frac{1}{19}c^4fx^{19}+\frac{1}{18}c^4ex^{18}+\frac{1}{4}bc^3ex^{16}+\frac{1}{17}(c^4d+4bc^3f)x^{17}+\frac{1}{7}(3b^2c^2+2ac^3)ex^{14}$$
$$+\frac{2}{15}(2bc^3d+(3b^2c^2+2ac^3)f)x^{15}+\frac{1}{3}(b^3c+3abc^2)ex^{12}$$
$$+\frac{2}{13}((3b^2c^2+2ac^3)d+2(b^3c+3abc^2)f)x^{13}+\frac{1}{10}(b^4+12ab^2c+6a^2c^2)ex^{10}$$
$$+\frac{1}{11}(4(b^3c+3abc^2)d+(b^4+12ab^2c+6a^2c^2)f)x^{11}+\frac{1}{2}(ab^3+3a^2bc)ex^8$$
$$+\frac{1}{9}((b^4+12ab^2c+6a^2c^2)d+4(ab^3+3a^2bc)f)x^9+a^3bex^4$$
$$+\frac{1}{3}(3a^2b^2+2a^3c)ex^6+\frac{2}{7}(2(ab^3+3a^2bc)d+(3a^2b^2+2a^3c)f)x^7$$
$$+\frac{1}{2}a^4ex^2+a^4dx+\frac{2}{5}(2a^3bf+(3a^2b^2+2a^3c)d)x^5+\frac{1}{3}(4a^3bd+a^4f)x^3$$

```
[In] integrate((c*x^4+b*x^2+a)^3*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="maxima")
```

```
[Out] 1/19*c^4*f*x^19 + 1/18*c^4*e*x^18 + 1/4*b*c^3*e*x^16 + 1/17*(c^4*d + 4*b*c^
3*f)*x^17 + 1/7*(3*b^2*c^2 + 2*a*c^3)*e*x^14 + 2/15*(2*b*c^3*d + (3*b^2*c^2
 + 2*a*c^3)*f)*x^15 + 1/3*(b^3*c + 3*a*b*c^2)*e*x^12 + 2/13*((3*b^2*c^2 + 2
*a*c^3)*d + 2*(b^3*c + 3*a*b*c^2)*f)*x^13 + 1/10*(b^4 + 12*a*b^2*c + 6*a^2*
c^2)*e*x^10 + 1/11*(4*(b^3*c + 3*a*b*c^2)*d + (b^4 + 12*a*b^2*c + 6*a^2*c^2
)*f)*x^11 + 1/2*(a*b^3 + 3*a^2*b*c)*e*x^8 + 1/9*((b^4 + 12*a*b^2*c + 6*a^2*
c^2)*d + 4*(a*b^3 + 3*a^2*b*c)*f)*x^9 + a^3*b*e*x^4 + 1/3*(3*a^2*b^2 + 2*a^
3*c)*e*x^6 + 2/7*(2*(a*b^3 + 3*a^2*b*c)*d + (3*a^2*b^2 + 2*a^3*c)*f)*x^7 +
1/2*a^4*e*x^2 + a^4*d*x + 2/5*(2*a^3*b*f + (3*a^2*b^2 + 2*a^3*c)*d)*x^5 + 1
/3*(4*a^3*b*d + a^4*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 463, normalized size of antiderivative = 1.11

$$\int \left(a+bx^2+cx^4\right)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$=\frac{1}{19}c^4fx^{19}+\frac{1}{18}c^4ex^{18}+\frac{1}{17}c^4dx^{17}+\frac{4}{17}bc^3fx^{17}+\frac{1}{4}bc^3ex^{16}+\frac{4}{15}bc^3dx^{15}+\frac{2}{5}b^2c^2fx^{15}+\frac{4}{15}ac^3fx^{15}+\frac{3}{7}b^2c^2ex^{14}+\frac{2}{7}ac^3ex^{14}+\frac{6}{13}b^2c^2dx^{13}+\frac{4}{13}ac^3dx^{13}+\frac{4}{13}b^3cfx^{13}+\frac{12}{13}abc^2fx^{13}+\frac{1}{3}b^3cex^{12}+abc^2ex^{12}+\frac{4}{11}b^3cdx^{11}+\frac{12}{11}abc^2dx^{11}+\frac{1}{11}b^4fx^{11}+\frac{12}{11}ab^2cfx^{11}+\frac{6}{11}a^2c^2fx^{11}+\frac{1}{10}b^4ex^{10}+\frac{6}{5}ab^2cex^{10}+\frac{3}{5}a^2c^2ex^{10}+\frac{1}{9}b^4dx^9+\frac{4}{3}ab^2cdx^9+\frac{2}{3}a^2c^2dx^9+\frac{4}{9}ab^3fx^9+\frac{4}{3}a^2bcfx^9+\frac{1}{2}ab^3ex^8+\frac{3}{2}a^2bcex^8+\frac{4}{7}ab^3dx^7+\frac{12}{7}a^2bcdx^7+\frac{6}{7}a^2b^2fx^7+\frac{4}{7}a^3cfx^7+a^2b^2ex^6+\frac{2}{3}a^3cex^6+\frac{6}{5}a^2b^2dx^5+\frac{4}{5}a^3cdx^5+\frac{4}{5}a^3bfx^5+a^3bex^4+\frac{4}{3}a^3bdx^3+\frac{1}{3}a^4fx^3+\frac{1}{2}a^4ex^2+a^4dx$$

```
[In] integrate((c*x^4+b*x^2+a)^3*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="giac")
```

```
[Out] 1/19*c^4*f*x^19 + 1/18*c^4*e*x^18 + 1/17*c^4*d*x^17 + 4/17*b*c^3*f*x^17 + 1
/4*b*c^3*e*x^16 + 4/15*b*c^3*d*x^15 + 2/5*b^2*c^2*f*x^15 + 4/15*a*c^3*f*x^1
5 + 3/7*b^2*c^2*e*x^14 + 2/7*a*c^3*e*x^14 + 6/13*b^2*c^2*d*x^13 + 4/13*a*c^
3*d*x^13 + 4/13*b^3*c*f*x^13 + 12/13*a*b*c^2*f*x^13 + 1/3*b^3*c*e*x^12 + a*
b*c^2*e*x^12 + 4/11*b^3*c*d*x^11 + 12/11*a*b*c^2*d*x^11 + 1/11*b^4*f*x^11 +
 12/11*a*b^2*c*f*x^11 + 6/11*a^2*c^2*f*x^11 + 1/10*b^4*e*x^10 + 6/5*a*b^2*c
*e*x^10 + 3/5*a^2*c^2*e*x^10 + 1/9*b^4*d*x^9 + 4/3*a*b^2*c*d*x^9 + 2/3*a^2*
c^2*d*x^9 + 4/9*a*b^3*f*x^9 + 4/3*a^2*b*c*f*x^9 + 1/2*a*b^3*e*x^8 + 3/2*a^2
*b*c*e*x^8 + 4/7*a*b^3*d*x^7 + 12/7*a^2*b*c*d*x^7 + 6/7*a^2*b^2*f*x^7 + 4/7
*a^3*c*f*x^7 + a^2*b^2*e*x^6 + 2/3*a^3*c*e*x^6 + 6/5*a^2*b^2*d*x^5 + 4/5*a^
3*c*d*x^5 + 4/5*a^3*b*f*x^5 + a^3*b*e*x^4 + 4/3*a^3*b*d*x^3 + 1/3*a^4*f*x^3
 + 1/2*a^4*e*x^2 + a^4*d*x
```

Mupad [B] (verification not implemented)

Time = 0.30 (sec) , antiderivative size = 398, normalized size of antiderivative = 0.96

$$\int \left(a+bx^2+cx^4\right)^3 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$= x^3\left(\frac{f\,a^4}{3}+\frac{4\,b\,d\,a^3}{3}\right)+x^{17}\left(\frac{d\,c^4}{17}+\frac{4\,b\,f\,c^3}{17}\right)$$
$$+x^5\left(\frac{4\,f\,a^3\,b}{5}+\frac{4\,c\,d\,a^3}{5}+\frac{6\,d\,a^2\,b^2}{5}\right)+x^{15}\left(\frac{2\,f\,b^2\,c^2}{5}+\frac{4\,d\,b\,c^3}{15}+\frac{4\,a\,f\,c^3}{15}\right)$$
$$+x^9\left(\frac{4\,f\,a^2\,b\,c}{3}+\frac{2\,d\,a^2\,c^2}{3}+\frac{4\,f\,a\,b^3}{9}+\frac{4\,d\,a\,b^2\,c}{3}+\frac{d\,b^4}{9}\right)$$
$$+x^{11}\left(\frac{6\,f\,a^2\,c^2}{11}+\frac{12\,f\,a\,b^2\,c}{11}+\frac{12\,d\,a\,b\,c^2}{11}+\frac{f\,b^4}{11}+\frac{4\,d\,b^3\,c}{11}\right)$$
$$+x^7\left(\frac{4\,c\,f\,a^3}{7}+\frac{6\,f\,a^2\,b^2}{7}+\frac{12\,c\,d\,a^2\,b}{7}+\frac{4\,d\,a\,b^3}{7}\right)$$
$$+x^{13}\left(\frac{4\,f\,b^3\,c}{13}+\frac{6\,d\,b^2\,c^2}{13}+\frac{12\,a\,f\,b\,c^2}{13}+\frac{4\,a\,d\,c^3}{13}\right)+\frac{a^4\,e\,x^2}{2}+\frac{c^4\,e\,x^{18}}{18}+\frac{c^4\,f\,x^{19}}{19}$$
$$+\frac{e\,x^{10}\left(6\,a^2\,c^2+12\,a\,b^2\,c+b^4\right)}{10}+a^4\,d\,x+\frac{a^2\,e\,x^6\left(3\,b^2+2\,a\,c\right)}{3}+\frac{c^2\,e\,x^{14}\left(3\,b^2+2\,a\,c\right)}{7}$$
$$+a^3\,b\,e\,x^4+\frac{b\,c^3\,e\,x^{16}}{4}+\frac{a\,b\,e\,x^8\left(b^2+3\,a\,c\right)}{2}+\frac{b\,c\,e\,x^{12}\left(b^2+3\,a\,c\right)}{3}$$

```
[In] int((a + b*x^2 + c*x^4)^3*(a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x
+ b*e*x^3 + c*e*x^5 + c*f*x^6),x)
```

```
[Out] x^3*((a^4*f)/3 + (4*a^3*b*d)/3) + x^17*((c^4*d)/17 + (4*b*c^3*f)/17) + x^5*
((6*a^2*b^2*d)/5 + (4*a^3*c*d)/5 + (4*a^3*b*f)/5) + x^15*((2*b^2*c^2*f)/5 +
 (4*b*c^3*d)/15 + (4*a*c^3*f)/15) + x^9*((b^4*d)/9 + (2*a^2*c^2*d)/3 + (4*a
*b^3*f)/9 + (4*a*b^2*c*d)/3 + (4*a^2*b*c*f)/3) + x^11*((b^4*f)/11 + (6*a^2*
c^2*f)/11 + (4*b^3*c*d)/11 + (12*a*b*c^2*d)/11 + (12*a*b^2*c*f)/11) + x^7*(
(6*a^2*b^2*f)/7 + (4*a*b^3*d)/7 + (4*a^3*c*f)/7 + (12*a^2*b*c*d)/7) + x^13*
((6*b^2*c^2*d)/13 + (4*a*c^3*d)/13 + (4*b^3*c*f)/13 + (12*a*b*c^2*f)/13) +
(a^4*e*x^2)/2 + (c^4*e*x^18)/18 + (c^4*f*x^19)/19 + (e*x^10*(b^4 + 6*a^2*c^
2 + 12*a*b^2*c))/10 + a^4*d*x + (a^2*e*x^6*(2*a*c + 3*b^2))/3 + (c^2*e*x^14
*(2*a*c + 3*b^2))/7 + a^3*b*e*x^4 + (b*c^3*e*x^16)/4 + (a*b*e*x^8*(3*a*c +
b^2))/2 + (b*c*e*x^12*(3*a*c + b^2))/3
```

3.61 $\int (a+bx^2+cx^4)^2 (ad+aex+(bd+af)x^2+bex^3+($

Optimal result

Integrand size = 63, antiderivative size = 259

$$\begin{aligned}
&\int (a+bx^2+cx^4)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx \\
&= a^3dx+\frac{1}{2}a^3ex^2+\frac{1}{3}a^2(3bd+af)x^3+\frac{3}{4}a^2bex^4+\frac{3}{5}a\left(b^2d+acd+abf\right)x^5 \\
&\quad+\frac{1}{2}a\left(b^2+ac\right)ex^6+\frac{1}{7}\left(b^3d+6abcd+3ab^2f+3a^2cf\right)x^7+\frac{1}{8}b\left(b^2+6ac\right)ex^8 \\
&\quad+\frac{1}{9}\left(3b^2cd+3ac^2d+b^3f+6abcf\right)x^9+\frac{3}{10}c\left(b^2+ac\right)ex^{10} \\
&\quad+\frac{3}{11}c\left(bcd+b^2f+acf\right)x^{11}+\frac{1}{4}bc^2ex^{12}+\frac{1}{13}c^2(cd+3bf)x^{13}+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}
\end{aligned}$$

```
[Out] a^3*d*x+1/2*a^3*e*x^2+1/3*a^2*(a*f+3*b*d)*x^3+3/4*a^2*b*e*x^4+3/5*a*(a*b*f+
a*c*d+b^2*d)*x^5+1/2*a*(a*c+b^2)*e*x^6+1/7*(3*a^2*c*f+3*a*b^2*f+6*a*b*c*d+b
^3*d)*x^7+1/8*b*(6*a*c+b^2)*e*x^8+1/9*(6*a*b*c*f+3*a*c^2*d+b^3*f+3*b^2*c*d)
*x^9+3/10*c*(a*c+b^2)*e*x^10+3/11*c*(a*c*f+b^2*f+b*c*d)*x^11+1/4*b*c^2*e*x^
12+1/13*c^2*(3*b*f+c*d)*x^13+1/14*c^3*e*x^14+1/15*c^3*f*x^15
```

Rubi [A] (verified)

Time = 0.22 (sec) , antiderivative size = 259, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.016$, Rules used

$= \{1685\}$

$$\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$= a^3dx+\frac{1}{2}a^3ex^2+\frac{1}{7}x^7\left(3a^2cf+3ab^2f+6abcd+b^3d\right)+\frac{1}{3}a^2x^3(af+3bd)+\frac{3}{4}a^2bex^4$$
$$+\frac{3}{11}cx^{11}\left(acf+b^2f+bcd\right)+\frac{3}{5}ax^5\left(abf+acd+b^2d\right)+\frac{3}{10}cex^{10}\left(ac+b^2\right)$$
$$+\frac{1}{8}bex^8\left(6ac+b^2\right)+\frac{1}{2}aex^6\left(ac+b^2\right)+\frac{1}{9}x^9\left(6abcf+3ac^2d+b^3f+3b^2cd\right)$$
$$+\frac{1}{13}c^2x^{13}(3bf+cd)+\frac{1}{4}bc^2ex^{12}+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}$$

```
[In] Int[(a + b*x^2 + c*x^4)^2*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d +
 b*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^3*d*x + (a^3*e*x^2)/2 + (a^2*(3*b*d + a*f)*x^3)/3 + (3*a^2*b*e*x^4)/4 + (
3*a*(b^2*d + a*c*d + a*b*f)*x^5)/5 + (a*(b^2 + a*c)*e*x^6)/2 + ((b^3*d + 6*
a*b*c*d + 3*a*b^2*f + 3*a^2*c*f)*x^7)/7 + (b*(b^2 + 6*a*c)*e*x^8)/8 + ((3*b
^2*c*d + 3*a*c^2*d + b^3*f + 6*a*b*c*f)*x^9)/9 + (3*c*(b^2 + a*c)*e*x^10)/1
0 + (3*c*(b*c*d + b^2*f + a*c*f)*x^11)/11 + (b*c^2*e*x^12)/4 + (c^2*(c*d +
3*b*f)*x^13)/13 + (c^3*e*x^14)/14 + (c^3*f*x^15)/15
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\text{integral} = \int \left(a^3d+a^3ex+a^2(3bd+af)x^2+3a^2bex^3+3a\left(b^2d+acd+abf\right)x^4\right.$$
$$+3a\left(b^2+ac\right)ex^5+\left(b^3d+6abcd+3ab^2f+3a^2cf\right)x^6+b\left(b^2+6ac\right)ex^7$$
$$+\left(3b^2cd+3ac^2d+b^3f+6abcf\right)x^8+3c\left(b^2+ac\right)ex^9+3c\left(bcd+b^2f+acf\right)x^{10}$$
$$\left.+3bc^2ex^{11}+c^2(cd+3bf)x^{12}+c^3ex^{13}+c^3fx^{14}\right)\,dx$$
$$= a^3dx+\frac{1}{2}a^3ex^2+\frac{1}{3}a^2(3bd+af)x^3+\frac{3}{4}a^2bex^4+\frac{3}{5}a\left(b^2d+acd+abf\right)x^5$$
$$+\frac{1}{2}a\left(b^2+ac\right)ex^6+\frac{1}{7}\left(b^3d+6abcd+3ab^2f+3a^2cf\right)x^7+\frac{1}{8}b\left(b^2+6ac\right)ex^8$$
$$+\frac{1}{9}\left(3b^2cd+3ac^2d+b^3f+6abcf\right)x^9+\frac{3}{10}c\left(b^2+ac\right)ex^{10}$$
$$+\frac{3}{11}c\left(bcd+b^2f+acf\right)x^{11}+\frac{1}{4}bc^2ex^{12}+\frac{1}{13}c^2(cd+3bf)x^{13}+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 259, normalized size of antiderivative = 1.00

$$\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$= a^3dx+\frac{1}{2}a^3ex^2+\frac{1}{3}a^2(3bd+af)x^3+\frac{3}{4}a^2bex^4+\frac{3}{5}a\left(b^2d+acd+abf\right)x^5$$
$$+\frac{1}{2}a\left(b^2+ac\right)ex^6+\frac{1}{7}\left(b^3d+6abcd+3ab^2f+3a^2cf\right)x^7+\frac{1}{8}b\left(b^2+6ac\right)ex^8$$
$$+\frac{1}{9}\left(3b^2cd+3ac^2d+b^3f+6abcf\right)x^9+\frac{3}{10}c\left(b^2+ac\right)ex^{10}$$
$$+\frac{3}{11}c\left(bcd+b^2f+acf\right)x^{11}+\frac{1}{4}bc^2ex^{12}+\frac{1}{13}c^2(cd+3bf)x^{13}+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)^2*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 +
(c*d + b*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^3*d*x + (a^3*e*x^2)/2 + (a^2*(3*b*d + a*f)*x^3)/3 + (3*a^2*b*e*x^4)/4 + (
3*a*(b^2*d + a*c*d + a*b*f)*x^5)/5 + (a*(b^2 + a*c)*e*x^6)/2 + ((b^3*d + 6*
a*b*c*d + 3*a*b^2*f + 3*a^2*c*f)*x^7)/7 + (b*(b^2 + 6*a*c)*e*x^8)/8 + ((3*b
^2*c*d + 3*a*c^2*d + b^3*f + 6*a*b*c*f)*x^9)/9 + (3*c*(b^2 + a*c)*e*x^10)/1
0 + (3*c*(b*c*d + b^2*f + a*c*f)*x^11)/11 + (b*c^2*e*x^12)/4 + (c^2*(c*d +
3*b*f)*x^13)/13 + (c^3*e*x^14)/14 + (c^3*f*x^15)/15
```

Maple [A] (verified)

Time = 27.40 (sec) , antiderivative size = 259, normalized size of antiderivative = 1.00

method	result
norman	$\left(\frac{1}{3}a^3f+a^2bd\right)x^3+\left(\frac{3}{10}ac^2e+\frac{3}{10}b^2ce\right)x^{10}+\left(\frac{1}{2}a^2ce+\frac{1}{2}ab^2e\right)x^6+\left(\frac{3}{13}bc^2f+\frac{1}{13}c^3d\right)x^{13}+ ($
risch	$a^3dx+\frac{3}{4}a^2bex^4+\frac{1}{2}ab^2ex^6+\frac{1}{8}b^3ex^8+\frac{1}{4}bc^2ex^{12}+\frac{1}{2}a^3ex^2+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}+\frac{3}{13}x^{13}b$
parallelrisch	$a^3dx+\frac{3}{4}a^2bex^4+\frac{1}{2}ab^2ex^6+\frac{1}{8}b^3ex^8+\frac{1}{4}bc^2ex^{12}+\frac{1}{2}a^3ex^2+\frac{1}{14}c^3ex^{14}+\frac{1}{15}c^3fx^{15}+\frac{3}{13}x^{13}b$
gosper	$x(24024c^3fx^{14}+25740c^3ex^{13}+83160bc^2fx^{12}+27720c^3dx^{12}+90090bc^2ex^{11}+98280ac^2fx^{10}+98280b^2cfx^{10}+98280bc^2dx^{10}$
default	$\frac{c^3fx^{15}}{15}+\frac{c^3ex^{14}}{14}+\frac{(2bc^2f+c^2(bf+cd))x^{13}}{13}+\frac{bc^2ex^{12}}{4}+\frac{((2ac+b^2)cf+2bc(bf+cd)+c^2(af+bd))x^{11}}{11}+ ((2ac+b^2)ec-$

```
[In] int((c*x^4+b*x^2+a)^2*(d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^
5+c*f*x^6),x,method=_RETURNVERBOSE)
```

```
[Out] (1/3*a^3*f+a^2*b*d)*x^3+(3/10*a*c^2*e+3/10*b^2*c*e)*x^10+(1/2*a^2*c*e+1/2*a
*b^2*e)*x^6+(3/13*b*c^2*f+1/13*c^3*d)*x^13+(3/4*a*b*c*e+1/8*b^3*e)*x^8+(3/1
1*a*c^2*f+3/11*b^2*c*f+3/11*b*c^2*d)*x^11+(3/5*a^2*b*f+3/5*a^2*c*d+3/5*a*b^
2*d)*x^5+(3/7*a^2*c*f+3/7*a*b^2*f+6/7*a*b*c*d+1/7*b^3*d)*x^7+(2/3*a*b*c*f+1
```

```
/3*a*c^2*d+1/9*b^3*f+1/3*b^2*c*d)*x^9+a^3*d*x+1/2*a^3*e*x^2+1/14*c^3*e*x^14
+1/15*c^3*f*x^15+3/4*a^2*b*e*x^4+1/4*b*c^2*e*x^12
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 251, normalized size of antiderivative = 0.97

$$\begin{aligned}\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx &= \frac{1}{15}\,c^3fx^{15}+\frac{1}{14}\,c^3ex^{14}+\frac{1}{4}\,bc^2ex^{12}+\frac{1}{13}\,\left(c^3d+3\,bc^2f\right)x^{13} \\ &+\frac{3}{10}\,\left(b^2c+ac^2\right)ex^{10}+\frac{3}{11}\,\left(bc^2d+\left(b^2c+ac^2\right)f\right)x^{11} \\ &+\frac{1}{8}\,\left(b^3+6\,abc\right)ex^8+\frac{1}{9}\,\left(3\,\left(b^2c+ac^2\right)d+\left(b^3+6\,abc\right)f\right)x^9 \\ &+\frac{3}{4}\,a^2bex^4+\frac{1}{2}\,\left(ab^2+a^2c\right)ex^6+\frac{1}{7}\,\left(\left(b^3+6\,abc\right)d+3\,\left(ab^2+a^2c\right)f\right)x^7 \\ &+\frac{1}{2}\,a^3ex^2+\frac{3}{5}\,\left(a^2bf+\left(ab^2+a^2c\right)d\right)x^5+a^3dx+\frac{1}{3}\,\left(3\,a^2bd+a^3f\right)x^3\end{aligned}$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="fricas")
```

```
[Out] 1/15*c^3*f*x^15 + 1/14*c^3*e*x^14 + 1/4*b*c^2*e*x^12 + 1/13*(c^3*d + 3*b*c^
2*f)*x^13 + 3/10*(b^2*c + a*c^2)*e*x^10 + 3/11*(b*c^2*d + (b^2*c + a*c^2)*f
)*x^11 + 1/8*(b^3 + 6*a*b*c)*e*x^8 + 1/9*(3*(b^2*c + a*c^2)*d + (b^3 + 6*a*
b*c)*f)*x^9 + 3/4*a^2*b*e*x^4 + 1/2*(a*b^2 + a^2*c)*e*x^6 + 1/7*((b^3 + 6*a
*b*c)*d + 3*(a*b^2 + a^2*c)*f)*x^7 + 1/2*a^3*e*x^2 + 3/5*(a^2*b*f + (a*b^2
+ a^2*c)*d)*x^5 + a^3*d*x + 1/3*(3*a^2*b*d + a^3*f)*x^3
```

Sympy [A] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 309, normalized size of antiderivative = 1.19

$$\begin{aligned}\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx &= a^3dx+\frac{a^3ex^2}{2}+\frac{3a^2bex^4}{4}+\frac{bc^2ex^{12}}{4}+\frac{c^3ex^{14}}{14}+\frac{c^3fx^{15}}{15}+x^{13}\cdot\left(\frac{3bc^2f}{13}+\frac{c^3d}{13}\right)+x^{11} \\ &\cdot\left(\frac{3ac^2f}{11}+\frac{3b^2cf}{11}+\frac{3bc^2d}{11}\right)+x^{10}\cdot\left(\frac{3ac^2e}{10}+\frac{3b^2ce}{10}\right)+x^9\cdot\left(\frac{2abcf}{3}+\frac{ac^2d}{3}+\frac{b^3f}{9}+\frac{b^2cd}{3}\right) \\ &+x^8\cdot\left(\frac{3abce}{4}+\frac{b^3e}{8}\right)+x^7\cdot\left(\frac{3a^2cf}{7}+\frac{3ab^2f}{7}+\frac{6abcd}{7}+\frac{b^3d}{7}\right) \\ &+x^6\left(\frac{a^2ce}{2}+\frac{ab^2e}{2}\right)+x^5\cdot\left(\frac{3a^2bf}{5}+\frac{3a^2cd}{5}+\frac{3ab^2d}{5}\right)+x^3\left(\frac{a^3f}{3}+a^2bd\right)\end{aligned}$$

```
[In] integrate((c*x**4+b*x**2+a)**2*(a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)
*x**4+c*e*x**5+c*f*x**6),x)
```

```
[Out] a**3*d*x + a**3*e*x**2/2 + 3*a**2*b*e*x**4/4 + b*c**2*e*x**12/4 + c**3*e*x*
*14/14 + c**3*f*x**15/15 + x**13*(3*b*c**2*f/13 + c**3*d/13) + x**11*(3*a*c
**2*f/11 + 3*b**2*c*f/11 + 3*b*c**2*d/11) + x**10*(3*a*c**2*e/10 + 3*b**2*c
*e/10) + x**9*(2*a*b*c*f/3 + a*c**2*d/3 + b**3*f/9 + b**2*c*d/3) + x**8*(3*
a*b*c*e/4 + b**3*e/8) + x**7*(3*a**2*c*f/7 + 3*a*b**2*f/7 + 6*a*b*c*d/7 + b
**3*d/7) + x**6*(a**2*c*e/2 + a*b**2*e/2) + x**5*(3*a**2*b*f/5 + 3*a**2*c*d
/5 + 3*a*b**2*d/5) + x**3*(a**3*f/3 + a**2*b*d)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 251, normalized size of antiderivative = 0.97

$$\begin{aligned}
&\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx \\
&= \frac{1}{15}\,c^3fx^{15} + \frac{1}{14}\,c^3ex^{14} + \frac{1}{4}\,bc^2ex^{12} + \frac{1}{13}\,\left(c^3d + 3\,bc^2f\right)x^{13} \\
&\quad + \frac{3}{10}\,\left(b^2c + ac^2\right)ex^{10} + \frac{3}{11}\,\left(bc^2d + \left(b^2c + ac^2\right)f\right)x^{11} \\
&\quad + \frac{1}{8}\,\left(b^3 + 6\,abc\right)ex^8 + \frac{1}{9}\,\left(3\,\left(b^2c + ac^2\right)d + \left(b^3 + 6\,abc\right)f\right)x^9 \\
&\quad + \frac{3}{4}\,a^2bex^4 + \frac{1}{2}\,\left(ab^2 + a^2c\right)ex^6 + \frac{1}{7}\,\left(\left(b^3 + 6\,abc\right)d + 3\,\left(ab^2 + a^2c\right)f\right)x^7 \\
&\quad + \frac{1}{2}\,a^3ex^2 + \frac{3}{5}\,\left(a^2bf + \left(ab^2 + a^2c\right)d\right)x^5 + a^3dx + \frac{1}{3}\,\left(3\,a^2bd + a^3f\right)x^3
\end{aligned}$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="maxima")
```

```
[Out] 1/15*c^3*f*x^15 + 1/14*c^3*e*x^14 + 1/4*b*c^2*e*x^12 + 1/13*(c^3*d + 3*b*c^
2*f)*x^13 + 3/10*(b^2*c + a*c^2)*e*x^10 + 3/11*(b*c^2*d + (b^2*c + a*c^2)*f
)*x^11 + 1/8*(b^3 + 6*a*b*c)*e*x^8 + 1/9*(3*(b^2*c + a*c^2)*d + (b^3 + 6*a*
b*c)*f)*x^9 + 3/4*a^2*b*e*x^4 + 1/2*(a*b^2 + a^2*c)*e*x^6 + 1/7*((b^3 + 6*a
*b*c)*d + 3*(a*b^2 + a^2*c)*f)*x^7 + 1/2*a^3*e*x^2 + 3/5*(a^2*b*f + (a*b^2
+ a^2*c)*d)*x^5 + a^3*d*x + 1/3*(3*a^2*b*d + a^3*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 285, normalized size of antiderivative = 1.10

$$\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= \frac{1}{15}c^3fx^{15}+\frac{1}{14}c^3ex^{14}+\frac{1}{13}c^3dx^{13}+\frac{3}{13}bc^2fx^{13}+\frac{1}{4}bc^2ex^{12}+\frac{3}{11}bc^2dx^{11}+\frac{3}{11}b^2cfx^{11} + \frac{3}{11}ac^2fx^{11}+\frac{3}{10}b^2cex^{10}+\frac{3}{10}ac^2ex^{10}+\frac{1}{3}b^2cdx^9+\frac{1}{3}ac^2dx^9+\frac{1}{9}b^3fx^9+\frac{2}{3}abcfx^9 + \frac{1}{8}b^3ex^8+\frac{3}{4}abcex^8+\frac{1}{7}b^3dx^7+\frac{6}{7}abcdx^7+\frac{3}{7}ab^2fx^7+\frac{3}{7}a^2cfx^7+\frac{1}{2}ab^2ex^6+\frac{1}{2}a^2cex^6 + \frac{3}{5}ab^2dx^5+\frac{3}{5}a^2cdx^5+\frac{3}{5}a^2bfx^5+\frac{3}{4}a^2bex^4+a^2bdx^3+\frac{1}{3}a^3fx^3+\frac{1}{2}a^3ex^2+a^3dx$$

```
[In] integrate((c*x^4+b*x^2+a)^2*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+
c*e*x^5+c*f*x^6),x, algorithm="giac")
```

```
[Out] 1/15*c^3*f*x^15 + 1/14*c^3*e*x^14 + 1/13*c^3*d*x^13 + 3/13*b*c^2*f*x^13 + 1
/4*b*c^2*e*x^12 + 3/11*b*c^2*d*x^11 + 3/11*b^2*c*f*x^11 + 3/11*a*c^2*f*x^11
 + 3/10*b^2*c*e*x^10 + 3/10*a*c^2*e*x^10 + 1/3*b^2*c*d*x^9 + 1/3*a*c^2*d*x^
9 + 1/9*b^3*f*x^9 + 2/3*a*b*c*f*x^9 + 1/8*b^3*e*x^8 + 3/4*a*b*c*e*x^8 + 1/7
*b^3*d*x^7 + 6/7*a*b*c*d*x^7 + 3/7*a*b^2*f*x^7 + 3/7*a^2*c*f*x^7 + 1/2*a*b^
2*e*x^6 + 1/2*a^2*c*e*x^6 + 3/5*a*b^2*d*x^5 + 3/5*a^2*c*d*x^5 + 3/5*a^2*b*f
*x^5 + 3/4*a^2*b*e*x^4 + a^2*b*d*x^3 + 1/3*a^3*f*x^3 + 1/2*a^3*e*x^2 + a^3*
d*x
```

Mupad [B] (verification not implemented)

Time = 8.11 (sec) , antiderivative size = 246, normalized size of antiderivative = 0.95

$$\int \left(a+bx^2+cx^4\right)^2 \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= x^3\left(\frac{f\,a^3}{3}+b\,d\,a^2\right)+x^{13}\left(\frac{d\,c^3}{13}+\frac{3\,b\,f\,c^2}{13}\right)+x^5\left(\frac{3\,f\,a^2\,b}{5}+\frac{3\,c\,d\,a^2}{5}+\frac{3\,d\,a\,b^2}{5}\right) + x^{11}\left(\frac{3\,f\,b^2\,c}{11}+\frac{3\,d\,b\,c^2}{11}+\frac{3\,a\,f\,c^2}{11}\right)+x^7\left(\frac{3\,c\,f\,a^2}{7}+\frac{3\,f\,a\,b^2}{7}+\frac{6\,c\,d\,a\,b}{7}+\frac{d\,b^3}{7}\right) + x^9\left(\frac{f\,b^3}{9}+\frac{d\,b^2\,c}{3}+\frac{2\,a\,f\,b\,c}{3}+\frac{a\,d\,c^2}{3}\right)+\frac{a^3\,e\,x^2}{2}+\frac{c^3\,e\,x^{14}}{14}+\frac{c^3\,f\,x^{15}}{15}+a^3\,d\,x + \frac{a\,e\,x^6\,\left(b^2+a\,c\right)}{2}+\frac{b\,e\,x^8\,\left(b^2+6\,a\,c\right)}{8}+\frac{3\,c\,e\,x^{10}\,\left(b^2+a\,c\right)}{10}+\frac{3\,a^2\,b\,e\,x^4}{4}+\frac{b\,c^2\,e\,x^{12}}{4}$$

```
[In] int((a + b*x^2 + c*x^4)^2*(a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x
+ b*e*x^3 + c*e*x^5 + c*f*x^6),x)
```

```
[Out] x^3*((a^3*f)/3 + a^2*b*d) + x^13*((c^3*d)/13 + (3*b*c^2*f)/13) + x^5*((3*a*
b^2*d)/5 + (3*a^2*c*d)/5 + (3*a^2*b*f)/5) + x^11*((3*b*c^2*d)/11 + (3*a*c^2
*f)/11 + (3*b^2*c*f)/11) + x^7*((b^3*d)/7 + (3*a*b^2*f)/7 + (3*a^2*c*f)/7 +
 (6*a*b*c*d)/7) + x^9*((b^3*f)/9 + (a*c^2*d)/3 + (b^2*c*d)/3 + (2*a*b*c*f)/
3) + (a^3*e*x^2)/2 + (c^3*e*x^14)/14 + (c^3*f*x^15)/15 + a^3*d*x + (a*e*x^6
*(a*c + b^2))/2 + (b*e*x^8*(6*a*c + b^2))/8 + (3*c*e*x^10*(a*c + b^2))/10 +
 (3*a^2*b*e*x^4)/4 + (b*c^2*e*x^12)/4
```

3.62 $\int (a+bx^2+cx^4) (ad+aex+(bd+af)x^2+bex^3+($

Optimal result

Integrand size = 61, antiderivative size = 154

$$\int (a+bx^2+cx^4) \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6$$
$$+\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}$$

```
[Out] a^2*d*x+1/2*a^2*e*x^2+1/3*a*(a*f+2*b*d)*x^3+1/2*a*b*e*x^4+1/5*(2*a*b*f+2*a*
c*d+b^2*d)*x^5+1/6*(2*a*c+b^2)*e*x^6+1/7*(2*a*c*f+b^2*f+2*b*c*d)*x^7+1/4*b*
c*e*x^8+1/9*c*(2*b*f+c*d)*x^9+1/10*c^2*e*x^10+1/11*c^2*f*x^11
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.016, Rules used = {1685}

$$\int (a+bx^2+cx^4) \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx$$
$$= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{7}x^7\left(2acf+b^2f+2bcd\right)+\frac{1}{5}x^5\left(2abf+2acd+b^2d\right)+\frac{1}{6}ex^6\left(2ac+b^2\right)$$
$$+\frac{1}{3}ax^3(af+2bd)+\frac{1}{2}abex^4+\frac{1}{9}cx^9(2bf+cd)+\frac{1}{4}bcex^8+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}$$

```
[In] Int[(a + b*x^2 + c*x^4)*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b
*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*b*e*x^4)/2 + ((b^2*d
 + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2 + 2*a*c)*e*x^6)/6 + ((2*b*c*d + b^2*f
```

```
+ 2*a*c*f)*x^7)/7 + (b*c*e*x^8)/4 + (c*(c*d + 2*b*f)*x^9)/9 + (c^2*e*x^10)/
10 + (c^2*f*x^11)/11
```

Rule 1685

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Int[Expa
ndIntegrand[Pq*(a + b*x^2 + c*x^4)^p, x], x] /; FreeQ[{a, b, c}, x] && Poly
Q[Pq, x] && IGtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \left(a^2d+a^2ex+a(2bd+af)x^2+2abex^3+\left(b^2d+2acd+2abf\right)x^4+\left(b^2+2ac\right)ex^5\right.\\ &\qquad \left.+\left(2bcd+b^2f+2acf\right)x^6+2bcex^7+c(cd+2bf)x^8+c^2ex^9+c^2fx^{10}\right)\,dx\\ &= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6\\ &\quad +\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \left(a+bx^2+cx^4\right)\left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx\\ &= a^2dx+\frac{1}{2}a^2ex^2+\frac{1}{3}a(2bd+af)x^3+\frac{1}{2}abex^4+\frac{1}{5}\left(b^2d+2acd+2abf\right)x^5+\frac{1}{6}\left(b^2+2ac\right)ex^6\\ &\quad +\frac{1}{7}\left(2bcd+b^2f+2acf\right)x^7+\frac{1}{4}bcex^8+\frac{1}{9}c(cd+2bf)x^9+\frac{1}{10}c^2ex^{10}+\frac{1}{11}c^2fx^{11}\end{aligned}$$

```
[In] Integrate[(a + b*x^2 + c*x^4)*(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c
*d + b*f)*x^4 + c*e*x^5 + c*f*x^6),x]
```

```
[Out] a^2*d*x + (a^2*e*x^2)/2 + (a*(2*b*d + a*f)*x^3)/3 + (a*b*e*x^4)/2 + ((b^2*d
 + 2*a*c*d + 2*a*b*f)*x^5)/5 + ((b^2 + 2*a*c)*e*x^6)/6 + ((2*b*c*d + b^2*f
+ 2*a*c*f)*x^7)/7 + (b*c*e*x^8)/4 + (c*(c*d + 2*b*f)*x^9)/9 + (c^2*e*x^10)/
10 + (c^2*f*x^11)/11
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 141, normalized size of antiderivative = 0.92

method	result
norman	$\frac{c^2 f x^{11}}{11}+\frac{c^2 e x^{10}}{10}+\left(\frac{2}{9} f b c+\frac{1}{9} c^2 d\right) x^9+\frac{b c e x^8}{4}+\left(\frac{2}{7} a c f+\frac{1}{7} b^2 f+\frac{2}{7} b c d\right) x^7+\left(\frac{1}{3} a c e+\frac{1}{6} b^2 e\right) x^6$ -
risch	$\frac{1}{11} c^2 f x^{11}+\frac{1}{10} c^2 e x^{10}+\frac{2}{9} x^9 f b c+\frac{1}{9} c^2 d x^9+\frac{1}{4} b c e x^8+\frac{2}{7} x^7 a c f+\frac{1}{7} x^7 b^2 f+\frac{2}{7} x^7 b c d+\frac{1}{3} x^6 a c e$
parallelrisch	$\frac{1}{11} c^2 f x^{11}+\frac{1}{10} c^2 e x^{10}+\frac{2}{9} x^9 f b c+\frac{1}{9} c^2 d x^9+\frac{1}{4} b c e x^8+\frac{2}{7} x^7 a c f+\frac{1}{7} x^7 b^2 f+\frac{2}{7} x^7 b c d+\frac{1}{3} x^6 a c e$
gosper	$\frac{x(1260 c^2 f x^{10}+1386 c^2 e x^9+3080 x^8 f b c+1540 x^8 c^2 d+3465 b c e x^7+3960 x^6 a c f+1980 x^6 b^2 f+3960 x^6 b c d+4620 x^5 a c e+2310 x^5 b^2}{13860}$
default	$\frac{c^2 f x^{11}}{11}+\frac{c^2 e x^{10}}{10}+\frac{(f b c+c(b f+c d)) x^9}{9}+\frac{b c e x^8}{4}+\frac{(a c f+b(b f+c d)+c(a f+b d)) x^7}{7}+\frac{(2 a c e+b^2 e) x^6}{6}+\frac{(a(b f+c d)+b}{}$

[In] int((c*x^4+b*x^2+a)*(d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6),x,method=_RETURNVERBOSE)

[Out] 1/11*c^2*f*x^11+1/10*c^2*e*x^10+(2/9*f*b*c+1/9*c^2*d)*x^9+1/4*b*c*e*x^8+(2/7*a*c*f+1/7*b^2*f+2/7*b*c*d)*x^7+(1/3*a*c*e+1/6*b^2*e)*x^6+(2/5*a*b*f+2/5*a*c*d+1/5*b^2*d)*x^5+1/2*a*b*e*x^4+(1/3*f*a^2+2/3*d*a*b)*x^3+1/2*a^2*e*x^2+a^2*d*x

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\begin{aligned}
&\int \left(a+bx^2+cx^4\right)\left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx \\
&=\frac{1}{11}c^2fx^{11}+\frac{1}{10}c^2ex^{10}+\frac{1}{4}bcex^8+\frac{1}{9}\left(c^2d+2\,bcf\right)x^9 \\
&\quad+\frac{1}{6}\left(b^2+2\,ac\right)ex^6+\frac{1}{7}\left(2\,bcd+\left(b^2+2\,ac\right)f\right)x^7+\frac{1}{2}\,abex^4 \\
&\quad+\frac{1}{5}\left(2\,abf+\left(b^2+2\,ac\right)d\right)x^5+\frac{1}{2}\,a^2ex^2+a^2dx+\frac{1}{3}\left(2\,abd+a^2f\right)x^3
\end{aligned}$$

[In] integrate((c*x^4+b*x^2+a)*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6),x, algorithm="fricas")

[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/4*b*c*e*x^8 + 1/9*(c^2*d + 2*b*c*f)*x^9 + 1/6*(b^2 + 2*a*c)*e*x^6 + 1/7*(2*b*c*d + (b^2 + 2*a*c)*f)*x^7 + 1/2*a*b*e*x^4 + 1/5*(2*a*b*f + (b^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + a^2*d*x + 1/3*(2*a*b*d + a^2*f)*x^3

Sympy [A] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.07

$$\int \left(a+bx^2+cx^4\right)\left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$=a^2dx+\frac{a^2ex^2}{2}+\frac{abex^4}{2}+\frac{bcex^8}{4}+\frac{c^2ex^{10}}{10}+\frac{c^2fx^{11}}{11}+x^9\cdot\left(\frac{2bcf}{9}+\frac{c^2d}{9}\right)+x^7\cdot\left(\frac{2acf}{7}+\frac{b^2f}{7}+\frac{2bcd}{7}\right)+x^6\left(\frac{ace}{3}+\frac{b^2e}{6}\right)+x^5\cdot\left(\frac{2abf}{5}+\frac{2acd}{5}+\frac{b^2d}{5}\right)+x^3\left(\frac{a^2f}{3}+\frac{2abd}{3}\right)$$

```
[In] integrate((c*x**4+b*x**2+a)*(a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)*x*
*4+c*e*x**5+c*f*x**6),x)
```

```
[Out] a**2*d*x + a**2*e*x**2/2 + a*b*e*x**4/2 + b*c*e*x**8/4 + c**2*e*x**10/10 +
c**2*f*x**11/11 + x**9*(2*b*c*f/9 + c**2*d/9) + x**7*(2*a*c*f/7 + b**2*f/7
+ 2*b*c*d/7) + x**6*(a*c*e/3 + b**2*e/6) + x**5*(2*a*b*f/5 + 2*a*c*d/5 + b*
*2*d/5) + x**3*(a**2*f/3 + 2*a*b*d/3)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\int \left(a+bx^2+cx^4\right)\left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right)\,dx$$
$$=\frac{1}{11}c^2fx^{11}+\frac{1}{10}c^2ex^{10}+\frac{1}{4}bcex^8+\frac{1}{9}\left(c^2d+2\,bcf\right)x^9+\frac{1}{6}\left(b^2+2\,ac\right)ex^6+\frac{1}{7}\left(2\,bcd+\left(b^2+2\,ac\right)f\right)x^7+\frac{1}{2}abex^4+\frac{1}{5}\left(2\,abf+\left(b^2+2\,ac\right)d\right)x^5+\frac{1}{2}a^2ex^2+a^2dx+\frac{1}{3}\left(2\,abd+a^2f\right)x^3$$

```
[In] integrate((c*x^4+b*x^2+a)*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*
e*x^5+c*f*x^6),x, algorithm="maxima")
```

```
[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/4*b*c*e*x^8 + 1/9*(c^2*d + 2*b*c*f)*x
^9 + 1/6*(b^2 + 2*a*c)*e*x^6 + 1/7*(2*b*c*d + (b^2 + 2*a*c)*f)*x^7 + 1/2*a*
b*e*x^4 + 1/5*(2*a*b*f + (b^2 + 2*a*c)*d)*x^5 + 1/2*a^2*e*x^2 + a^2*d*x + 1
/3*(2*a*b*d + a^2*f)*x^3
```

Giac [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 151, normalized size of antiderivative = 0.98

$$\int \left(a+bx^2+cx^4\right) \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx = \frac{1}{11}c^2fx^{11}+\frac{1}{10}c^2ex^{10}+\frac{1}{9}c^2dx^9+\frac{2}{9}bcfx^9+\frac{1}{4}bcex^8+\frac{2}{7}bcdx^7+\frac{1}{7}b^2fx^7+\frac{2}{7}acfx^7+\frac{1}{6}b^2ex^6+\frac{1}{3}acex^6+\frac{1}{5}b^2dx^5+\frac{2}{5}acdx^5+\frac{2}{5}abfx^5+\frac{1}{2}abex^4+\frac{2}{3}abdx^3+\frac{1}{3}a^2fx^3+\frac{1}{2}a^2ex^2+a^2dx$$

```
[In] integrate((c*x^4+b*x^2+a)*(a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*
e*x^5+c*f*x^6),x, algorithm="giac")
```

```
[Out] 1/11*c^2*f*x^11 + 1/10*c^2*e*x^10 + 1/9*c^2*d*x^9 + 2/9*b*c*f*x^9 + 1/4*b*c
*e*x^8 + 2/7*b*c*d*x^7 + 1/7*b^2*f*x^7 + 2/7*a*c*f*x^7 + 1/6*b^2*e*x^6 + 1/
3*a*c*e*x^6 + 1/5*b^2*d*x^5 + 2/5*a*c*d*x^5 + 2/5*a*b*f*x^5 + 1/2*a*b*e*x^4
 + 2/3*a*b*d*x^3 + 1/3*a^2*f*x^3 + 1/2*a^2*e*x^2 + a^2*d*x
```

Mupad [B] (verification not implemented)

Time = 0.07 (sec) , antiderivative size = 138, normalized size of antiderivative = 0.90

$$\int \left(a+bx^2+cx^4\right) \left(ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6\right) dx = x^5\left(\frac{d\,b^2}{5}+\frac{2\,a\,f\,b}{5}+\frac{2\,a\,c\,d}{5}\right)+x^7\left(\frac{f\,b^2}{7}+\frac{2\,c\,d\,b}{7}+\frac{2\,a\,c\,f}{7}\right)+x^3\left(\frac{f\,a^2}{3}+\frac{2\,b\,d\,a}{3}\right)+x^9\left(\frac{d\,c^2}{9}+\frac{2\,b\,f\,c}{9}\right)+\frac{a^2\,e\,x^2}{2}+\frac{c^2\,e\,x^{10}}{10}+\frac{c^2\,f\,x^{11}}{11}+\frac{e\,x^6\,\left(b^2+2\,a\,c\right)}{6}+a^2\,d\,x+\frac{a\,b\,e\,x^4}{2}+\frac{b\,c\,e\,x^8}{4}$$

```
[In] int((a + b*x^2 + c*x^4)*(a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x +
b*e*x^3 + c*e*x^5 + c*f*x^6),x)
```

```
[Out] x^5*((b^2*d)/5 + (2*a*c*d)/5 + (2*a*b*f)/5) + x^7*((b^2*f)/7 + (2*b*c*d)/7
+ (2*a*c*f)/7) + x^3*((a^2*f)/3 + (2*a*b*d)/3) + x^9*((c^2*d)/9 + (2*b*c*f)
/9) + (a^2*e*x^2)/2 + (c^2*e*x^10)/10 + (c^2*f*x^11)/11 + (e*x^6*(2*a*c + b
^2))/6 + a^2*d*x + (a*b*e*x^4)/2 + (b*c*e*x^8)/4
```

3.63 $\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx$

Optimal result

Integrand size = 63, antiderivative size = 20

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx=dx+\frac{ex^2}{2}+\frac{fx^3}{3}$$

```
[Out] d*x+1/2*e*x^2+1/3*f*x^3
```

Rubi [A] (verified)

Time = 0.02 (sec) , antiderivative size = 20, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.016, Rules used = {1600}

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx=dx+\frac{ex^2}{2}+\frac{fx^3}{3}$$

```
[In] Int[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4),x]
```

```
[Out] d*x + (e*x^2)/2 + (f*x^3)/3
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}\text{integral}&=\int \left(d+ex+fx^2\right)\,dx\\&=dx+\frac{ex^2}{2}+\frac{fx^3}{3}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.00 (sec) , antiderivative size = 20, normalized size of antiderivative = 1.00

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{a + bx^2 + cx^4} dx = dx + \frac{ex^2}{2} + \frac{fx^3}{3}$$

```
[In] Integrate[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*
x^5 + c*f*x^6)/(a + b*x^2 + c*x^4),x]
```

```
[Out] d*x + (e*x^2)/2 + (f*x^3)/3
```

Maple [A] (verified)

Time = 0.02 (sec) , antiderivative size = 17, normalized size of antiderivative = 0.85

method	result	size
default	$dx + \frac{1}{2} e\, x^2 + \frac{1}{3} f\, x^3$	17
norman	$dx + \frac{1}{2} e\, x^2 + \frac{1}{3} f\, x^3$	17
risch	$dx + \frac{1}{2} e\, x^2 + \frac{1}{3} f\, x^3$	17
parallelrisch	$dx + \frac{1}{2} e\, x^2 + \frac{1}{3} f\, x^3$	17
parts	$dx + \frac{1}{2} e\, x^2 + \frac{1}{3} f\, x^3$	17
gosper	$\frac{x(2f\, x^2+3ex+6d)}{6}$	18

```
[In] int((d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+
b*x^2+a),x,method=_RETURNVERBOSE)
```

```
[Out] d*x+1/2*e*x^2+1/3*f*x^3
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 16, normalized size of antiderivative = 0.80

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{a + bx^2 + cx^4} dx = \frac{1}{3} fx^3 + \frac{1}{2} ex^2 + dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(
c*x^4+b*x^2+a),x, algorithm="fricas")
```

```
[Out] 1/3*f*x^3 + 1/2*e*x^2 + d*x
```

Sympy [A] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 15, normalized size of antiderivative = 0.75

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx = dx+\frac{ex^2}{2}+\frac{fx^3}{3}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)*x**4+c*e*x**5+c*f*x**6)/(c*x**4+b*x**2+a),x)
```

```
[Out] d*x + e*x**2/2 + f*x**3/3
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 16, normalized size of antiderivative = 0.80

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx = \frac{1}{3}\,fx^3+\frac{1}{2}\,ex^2+dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a),x, algorithm="maxima")
```

```
[Out] 1/3*f*x^3 + 1/2*e*x^2 + d*x
```

Giac [A] (verification not implemented)

none

Time = 0.62 (sec) , antiderivative size = 16, normalized size of antiderivative = 0.80

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{a+bx^2+cx^4}\,dx = \frac{1}{3}\,fx^3+\frac{1}{2}\,ex^2+dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a),x, algorithm="giac")
```

```
[Out] 1/3*f*x^3 + 1/2*e*x^2 + d*x
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 16, normalized size of antiderivative = 0.80

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{a + bx^2 + cx^4} dx = \frac{f x^3}{3} + \frac{e x^2}{2} + d x$$

```
[In] int((a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x + b*e*x^3 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4),x)
```

```
[Out] d*x + (e*x^2)/2 + (f*x^3)/3
```

3.64 $\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^2}\,dx$

Optimal result

Integrand size = 63, antiderivative size = 211

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^2}\,dx = \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[Out] -e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(1/2)+1/2*arctan(x*
2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))*(f+(-b*f+2*c*d)/(-4*a*c+b^2)^
(1/2))*2^(1/2)/c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/2*arctan(x*2^(1/2)*c^
(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*(f+(b*f-2*c*d)/(-4*a*c+b^2)^(1/2))*2^(1
/2)/c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 211, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.127, Rules used

= {1600, 1687, 1180, 211, 12, 1121, 632, 212}

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^2}\,dx$$
$$=\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{2cd-bf}{\sqrt{b^2-4ac}}+f\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}}+\frac{\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)}{\sqrt{2}\sqrt{c}\sqrt{\sqrt{b^2-4ac}+b}}-\frac{e\operatorname{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

```
[In] Int[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((f + (2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b -
Sqrt[b^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + ((f -
(2*c*d - b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b
^2 - 4*a*c]]])/(Sqrt[2]*Sqrt[c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (e*ArcTanh[(
b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/Sqrt[b^2 - 4*a*c]
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{a+bx^2+cx^4}\,dx \\
&= \int \frac{ex}{a+bx^2+cx^4}\,dx + \int \frac{d+fx^2}{a+bx^2+cx^4}\,dx \\
&= e\int \frac{x}{a+bx^2+cx^4}\,dx + \frac{1}{2}\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&\quad + \frac{1}{2}\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right) \\
&= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad - e\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)
\end{aligned}$$

$$= \frac{\left(f+\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\left(f-\frac{2cd-bf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{2}\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\sqrt{b^2-4ac}}$$

Mathematica [A] (verified)

Time = 0.14 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.11

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{\frac{\sqrt{2}\left(2cd+\left(-b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{\sqrt{c}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{2}\left(-2cd+\left(b+\sqrt{b^2-4ac}\right)f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{\sqrt{c}\sqrt{b+\sqrt{b^2-4ac}}} + e\log\left(-b+\sqrt{b^2-4}\right.}{2\sqrt{b^2-4ac}}$$

```
[In] Integrate[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*
x^5 + c*f*x^6)/(a + b*x^2 + c*x^4)^2,x]
```

```
[Out] ((Sqrt[2]*(2*c*d + (-b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/S
qrt[b - Sqrt[b^2 - 4*a*c]]])/(Sqrt[c]*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[
2]*(-2*c*d + (b + Sqrt[b^2 - 4*a*c])*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
 Sqrt[b^2 - 4*a*c]]])/(Sqrt[c]*Sqrt[b + Sqrt[b^2 - 4*a*c]]) + e*Log[-b + Sq
rt[b^2 - 4*a*c] - 2*c*x^2] - e*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(2*Sqr
t[b^2 - 4*a*c])
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.06 (sec) , antiderivative size = 48, normalized size of antiderivative = 0.23

method	result
risch	$\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2b+a\right)}\frac{\left(\_R^2f+\_Re+d\right)\ln\left(x-\_R\right)}{2c\_R^3+\_Rb}\right)}{2}$
default	$4c\left(-\frac{\sqrt{-4ac+b^2}\left(-\frac{e\ln\left(2cx^2+\sqrt{-4ac+b^2}+b\right)}{2}+\frac{\left(f\sqrt{-4ac+b^2}+bf-2cd\right)\sqrt{2}\arctan\left(\frac{cx\sqrt{2}}{\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{2\sqrt{\left(b+\sqrt{-4ac+b^2}\right)c}}\right)}{4c(4ac-b^2)}-\frac{\sqrt{-4ac+b^2}\left(\frac{e\ln\left(-2cx\right.}{}\right.}{}\right.$

[In] int((d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^2,x,method=_RETURNVERBOSE)

[Out] 1/2*sum((_R^2*f+_R*e+d)/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 18.53 (sec) , antiderivative size = 723401, normalized size of antiderivative = 3428.44

$$\int\frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^2}\,dx=\text{Too large to display}$$

[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^2,x, algorithm="fricas")

[Out] Too large to include

Sympy [F(-1)]

Timed out.

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)*x**4+c*e*x**5+c*f*x**6)/(c*x**4+b*x**2+a)**2,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^2}\,dx = \int \frac{cfx^6 + cex^5 + bex^3 + (cd + bf)x^4 + aex + (bd + af)x^2 + ad}{(cx^4 + bx^2 + a)^2}\,dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^2,x, algorithm="maxima")
```

```
[Out] integrate((c*f*x^6 + c*e*x^5 + b*e*x^3 + (c*d + b*f)*x^4 + a*e*x + (b*d + a*f)*x^2 + a*d)/(c*x^4 + b*x^2 + a)^2, x)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1616 vs. $2(173) = 346$.

Time = 1.23 (sec) , antiderivative size = 1616, normalized size of antiderivative = 7.66

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^2,x, algorithm="giac")
```

```
[Out] -1/2*(b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*(b + sqrt(b^2 - 4*a*c))/c)/((b^4 - 8*a*b^2*c - 2*b^3*c + 16*a^2*c^2 + 8*a*b*c^2 + b^2*c^2 - 4*a*c^3)*c^2) - 1/2*(b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*(b - sqrt(b^2 - 4*a*c))/c)/((b^4 - 8*a*b^2*c - 2*b^3*c + 16*a^2*c^2 + 8*a*b*c^2 + b^2*c^2 - 4*a*c^3)*c^2) + 1/4*((sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^4 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3*c - 2*b^4*
```

```
c + 16*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*c^2 + 8*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b
^2*c^2 + 16*a*b^2*c^2 - 2*b^3*c^2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*a*c^3 - 32*a^2*c^3 + 8*a*b*c^3 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sq
rt(b^2 - 4*a*c)*c)*b^3 - 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a*b*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*
c)*b^2*c + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^2
+ 2*(b^2 - 4*a*c)*b^2*c - 8*(b^2 - 4*a*c)*a*c^2 + 2*(b^2 - 4*a*c)*b*c^2)*d
+ 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^
2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a
*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*
a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2
*(b^2 - 4*a*c)*a*c^2)*f)*arctan(2*sqrt(1/2)*x/sqrt((b + sqrt(b^2 - 4*a*c))/
c))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^
2 - 4*a^2*c^3)*abs(c)) + 1/4*((sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^4
- 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2*c - 2*sqrt(2)*sqrt(b*c -
sqrt(b^2 - 4*a*c)*c)*b^3*c + 2*b^4*c + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a
*c)*c)*a^2*c^2 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c^2 + sqrt(2
)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c^2 - 16*a*b^2*c^2 - 2*b^3*c^2 - 4*sq
rt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^3 + 32*a^2*c^3 + 8*a*b*c^3 + sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3 - 4*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c - 2*sqrt(2)*sqrt(b^2 - 4
*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c + sqrt(2)*sqrt(b^2 - 4*a*c)*sqr
t(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b^2*c + 8*(b^2 - 4*a*c
)*a*c^2 + 2*(b^2 - 4*a*c)*b*c^2)*d + 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2
 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*f)*arctan(2*sqrt(1
/2)*x/sqrt((b - sqrt(b^2 - 4*a*c))/c))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c +
16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*abs(c))
```

Mupad [B] (verification not implemented)

Time = 0.64 (sec) , antiderivative size = 3942, normalized size of antiderivative = 18.68

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^2} dx = \text{Too large to display}$$

```
[In] int((a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x + b*e*x^3 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^2,x)
```

```
[Out] symsum(log(c^2*d*e^2 - c^2*d^2*f + c^2*e^3*x - a*c*f^3 - 8*root(16*a*b^4*c*
z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c
^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 +
```

```
32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z +
 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a
*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^
4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^3*b^3*c^2*x + b*c*d*f^2 - 16*roo
t(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z
^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c
^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^
2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*
d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2
*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*a*c^3*d - 4*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*c^3*d^2*x + 4*root(16
*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2 +
 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*d
^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c*
e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e^
2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^2
 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*b^2*c^2*d + 32*root(1
6*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
+ 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2*
d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*c
*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*e
^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f^
2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^3*a*b*c^3*x + 16*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*a*c^3*e*x + 4*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*a*c^2*f^2*x + 2*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
```

```
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b*c^2*e^2*x - 2*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b^2*c*f^2*x - 4*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)^2*b^2*c^2*e*x + 4*roo
t(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z
^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c
^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^
2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*
d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2
*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*b*c^2*d*e - 8*root(
16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3*z^4 - 16*a*b^2*c*d*f*z^2
 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b^2*c*e^2*z^2 - 16*a*b*c^2
*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 + 4*a*b^3*f^2*z^2 + 16*a^2*
c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16*a*c^2*d^2*e*z - 4*a*c*d*
e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3 + b*c*d^2*e^2 + a*b*e^2*f
^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z, k)*a*c^2*e*f + b*c*e*f^2
*x - 2*c^2*d*e*f*x + 4*root(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*
c^3*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*
a*b^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2
 + 4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z -
 16*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f
^3 + b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4,
 z, k)*b*c^2*d*f*x)*root(16*a*b^4*c*z^4 - 128*a^2*b^2*c^2*z^4 + 256*a^3*c^3
*z^4 - 16*a*b^2*c*d*f*z^2 + 64*a^2*c^2*d*f*z^2 - 16*a^2*b*c*f^2*z^2 - 8*a*b
^2*c*e^2*z^2 - 16*a*b*c^2*d^2*z^2 + 32*a^2*c^2*e^2*z^2 + 4*b^3*c*d^2*z^2 +
4*a*b^3*f^2*z^2 + 16*a^2*c*e*f^2*z + 4*b^2*c*d^2*e*z - 4*a*b^2*e*f^2*z - 16
*a*c^2*d^2*e*z - 4*a*c*d*e^2*f + 2*a*c*d^2*f^2 - 2*b*c*d^3*f - 2*a*b*d*f^3
+ b*c*d^2*e^2 + a*b*e^2*f^2 + a*c*e^4 + b^2*d^2*f^2 + c^2*d^4 + a^2*f^4, z,
 k), k, 1, 4)
```

3.65 $\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx$

Optimal result

Integrand size = 63, antiderivative size = 368

$$\begin{aligned}
&\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx \\
&= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} \\
&\quad + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{2ce\,\mathrm{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}
\end{aligned}$$

```
[Out] -1/2*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+1/2*x*(b^2*d-2*a*c*d-a*b*f+
c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)+2*c*e*arctanh((2*c*x^2+b
)/(-4*a*c+b^2)^(1/2))/(-4*a*c+b^2)^(3/2)+1/4*arctan(x*2^(1/2)*c^(1/2)/(b-(-
4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b*d-2*a*f+(4*a*b*f-12*a*c*d+b^2*d)/(-4*a*
c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2)+1/4*arcta
n(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(b*d-2*a*f+(-4*a*
b*f+12*a*c*d-b^2*d)/(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)*2^(1/2)/(b+(-4*a*c+b
^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 0.56 (sec) , antiderivative size = 368, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}} = 0.159$, Rules used = {1600, 1687, 1192, 1180, 211, 12, 1121, 628, 632, 212}

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a\,(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{4abf-12acd+b^2d}{\sqrt{b^2-4ac}}-2af+bd\right)}{2\sqrt{2}a\,(b^2-4ac)\sqrt{\sqrt{b^2-4ac}+b}}+\frac{2ce\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$
$$+\frac{x(cx^2(bd-2af)-abf-2acd+b^2d)}{2a\,(b^2-4ac)\,(a+bx^2+cx^4)}-\frac{e(b+2cx^2)}{2\,(b^2-4ac)\,(a+bx^2+cx^4)}$$

```
[In] Int[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] -1/2*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)) + (x*(b^2*d - 2*
a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(2*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x^4
)) + (Sqrt[c]*(b*d - 2*a*f + (b^2*d - 12*a*c*d + 4*a*b*f)/Sqrt[b^2 - 4*a*c]
)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^
2 - 4*a*c)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sqrt[c]*(b*d - 2*a*f - (b^2*d -
12*a*c*d + 4*a*b*f)/Sqrt[b^2 - 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
Sqrt[b^2 - 4*a*c]]])/(2*Sqrt[2]*a*(b^2 - 4*a*c)*Sqrt[b + Sqrt[b^2 - 4*a*c]]
) + (2*c*e*ArcTanh[(b + 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(3/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
```

```
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 && !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2}{(a+bx^2+cx^4)^2}\,dx$$

$$= \int \frac{ex}{(a+bx^2+cx^4)^2}\,dx + \int \frac{d+fx^2}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} - \frac{\int \frac{-b^2d+6acd-abf-c(bd-2af)x^2}{a+bx^2+cx^4}\,dx}{2a(b^2-4ac)} + e\int \frac{x}{(a+bx^2+cx^4)^2}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right) + \frac{\left(c\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)} + \frac{\left(c\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{4a(b^2-4ac)}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} - \frac{(ce)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{(2ce)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{b^2-4ac}$$

$$= -\frac{e(b+2cx^2)}{2(b^2-4ac)(a+bx^2+cx^4)} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{2a(b^2-4ac)(a+bx^2+cx^4)} + \frac{\sqrt{c}\left(bd-2af+\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{c}\left(bd-2af-\frac{b^2d-12acd+4abf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{2\sqrt{2}a(b^2-4ac)\sqrt{b+\sqrt{b^2-4ac}}} + \frac{2ce\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{3/2}}$$

Mathematica [A] (verified)

Time = 0.74 (sec) , antiderivative size = 398, normalized size of antiderivative = 1.08

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx$$
$$= \frac{1}{4}\left(\frac{2ab(e+fx)-2bdx(b+cx^2)+4acx(d+x(e+fx))}{a(-b^2+4ac)(a+bx^2+cx^4)} + \frac{\sqrt{2}\sqrt{c}\left(b^2d+b\left(\sqrt{b^2-4ac}d+4af\right)-2a\left(6cd+\sqrt{b^2-4ac}f\right)\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b-\sqrt{b^2-4ac}}} + \frac{\sqrt{2}\sqrt{c}\left(-b^2d+12acd+b\sqrt{b^2-4ac}d-4abf-2a\sqrt{b^2-4ac}f\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{a(b^2-4ac)^{3/2}\sqrt{b+\sqrt{b^2-4ac}}} - \frac{4ce\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{3/2}} + \frac{4ce\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{3/2}}\right)$$

```
[In] Integrate[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*
x^5 + c*f*x^6)/(a + b*x^2 + c*x^4)^3,x]
```

```
[Out] ((2*a*b*(e + f*x) - 2*b*d*x*(b + c*x^2) + 4*a*c*x*(d + x*(e + f*x)))/(a*(-b
^2 + 4*a*c)*(a + b*x^2 + c*x^4)) + (Sqrt[2]*Sqrt[c]*(b^2*d + b*(Sqrt[b^2 -
4*a*c]*d + 4*a*f) - 2*a*(6*c*d + Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt
[c]*x)/Sqrt[b - Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b - Sqrt[b
^2 - 4*a*c]]) + (Sqrt[2]*Sqrt[c]*(-(b^2*d) + 12*a*c*d + b*Sqrt[b^2 - 4*a*c]
*d - 4*a*b*f - 2*a*Sqrt[b^2 - 4*a*c]*f)*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b +
 Sqrt[b^2 - 4*a*c]]])/(a*(b^2 - 4*a*c)^(3/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) -
 (4*c*e*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(3/2) + (4*c*e
*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(3/2))/4
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.24 (sec) , antiderivative size = 232, normalized size of antiderivative = 0.63

method	result
risch	$\frac{\frac{c(2af-bd)x^3}{2a\left(4ac-b^2\right)}+\frac{c\,x^2 e}{4ac-b^2}+\frac{\left(abf+2acd-b^2 d\right)x}{2a\left(4ac-b^2\right)}+\frac{be}{8ac-2b^2}}{c\,x^4+b\,x^2+a}+\frac{\left(\sum_{\_R=\mathrm{RootOf}\left(c\_Z^4+\_Z^2 b+a\right)}\frac{\left(\frac{c(2af-bd)\_R^2}{a\left(4ac-b^2\right)}+\frac{4ce\_R}{4ac-b^2}-\frac{abf-6acd+b^2 d}{a\left(4ac-b^2\right)}\right)}{2c\_R^3+\_Rb}\right.}{4}$
default	$16c^2\left(-\frac{-\frac{\left(-4acd\sqrt{-4ac+b^2}+b^2 d\sqrt{-4ac+b^2}+8a^2cf-2a\,b^2 f-4abcd+b^3 d\right)x}{16ac}-\frac{e\left(4ac-b^2\right)}{8c}}{x^2+\frac{b}{2c}-\frac{\sqrt{-4ac+b^2}}{2c}}+\frac{2ae\sqrt{-4ac+b^2}\,\ln\left(-2c\,x^2+\sqrt{-4ac+b^2}-b\right)+\frac{\left(4\sqrt{-\right.}{}}{}}{4c(4ac-b^2)^2}\right.$

[In] int((d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^3,x,method=_RETURNVERBOSE)

[Out] (1/2*c*(2*a*f-b*d)/a/(4*a*c-b^2)*x^3+c/(4*a*c-b^2)*x^2*e+1/2*(a*b*f+2*a*c*d-b^2*d)/a/(4*a*c-b^2)*x+1/2/(4*a*c-b^2)*b*e)/(c*x^4+b*x^2+a)+1/4*sum((c*(2*a*f-b*d)/a/(4*a*c-b^2)*_R^2+4*c/(4*a*c-b^2)*e*_R-(a*b*f-6*a*c*d+b^2*d)/a/(4*a*c-b^2))/(2*_R^3*c+_R*b)*ln(x-_R),_R=RootOf(_Z^4*c+_Z^2*b+a))

Fricas [F(-1)]

Timed out.

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx=\text{Timed out}$$

[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^3,x, algorithm="fricas")

[Out] Timed out

Sympy [F(-1)]

Timed out.

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^3} dx = \text{Timed out}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)*x**4+c*e*x**5+c*f*x**6)/(c*x**4+b*x**2+a)**3,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^3} dx \\ = \int \frac{cfx^6 + cex^5 + bex^3 + (cd + bf)x^4 + aex + (bd + af)x^2 + ad}{(cx^4 + bx^2 + a)^3} dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^3,x, algorithm="maxima")
```

```
[Out] -1/2*(2*a*c*e*x^2 - (b*c*d - 2*a*c*f)*x^3 + a*b*e + (a*b*f - (b^2 - 2*a*c)*d)*x)/((a*b^2*c - 4*a^2*c^2)*x^4 + a^2*b^2 - 4*a^3*c + (a*b^3 - 4*a^2*b*c)*x^2) - 1/2*integrate((4*a*c*e*x - a*b*f - (b*c*d - 2*a*c*f)*x^2 - (b^2 - 6*a*c)*d)/(c*x^4 + b*x^2 + a), x)/(a*b^2 - 4*a^2*c)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5159 vs. 2(320) = 640.

Time = 2.82 (sec) , antiderivative size = 5159, normalized size of antiderivative = 14.02

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^3} dx = \text{Too large to display}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+b*x^2+a)^3,x, algorithm="giac")
```

```
[Out] 1/2*(b*c*d*x^3 - 2*a*c*f*x^3 - 2*a*c*e*x^2 + b^2*d*x - 2*a*c*d*x - a*b*f*x - a*b*e)/((c*x^4 + b*x^2 + a)*(a*b^2 - 4*a^2*c)) - 1/16*((2*b^3*c^2 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 -
```

4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b
^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4
*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^2*c)^2*f - 2*(sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 2*a
*b^6*c + 64*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 20*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2
 - 4*a*c)*c)*a*b^4*c^2 + 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*c + sqrt(b^2 -
4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 -
10*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 128*a^3*b^2*c^3 +
24*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 192*a^4*c^4 + 2*(b^2 -
 4*a*c)*a*b^4*c - 20*(b^2 - 4*a*c)*a^2*b^2*c^2 + 48*(b^2 - 4*a*c)*a^3*c^3)*
d*abs(a*b^2 - 4*a^2*c) - 2*(sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5
 - 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b*c
 + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*a^2*b^5*c + 16*sqrt(2)*sqrt(b*c + sqr
t(b^2 - 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3
*b^2*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + 16*a^3*b^3
*c^2 - 4*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 32*a^4*b*c^3 +
 2*(b^2 - 4*a*c)*a^2*b^3*c - 8*(b^2 - 4*a*c)*a^3*b*c^2)*f*abs(a*b^2 - 4*a^2
*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c^4 - 384*a^5*b*c^5 - s
qrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^7 + 20*sqrt(
2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c + 2*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c - 112*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - 32*sqrt(2)*s
qrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 - sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^2 + 192*sqrt(2)*sqr
t(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^5*b*c^3 + 96*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 16*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 48*sqrt(2)*sqrt(
b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^
2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*a^4*b*c^4)*d +
4*(2*a^3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a
*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt
(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
 sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^3*b^4*c^2 + 8*(b^2 - 4
*a*c)*a^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c + sqrt((
a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2
*c - 4*a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 48*a^5*b^2*c^2 +

16*a^4*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 - 8*a^4*b^2*c^3 + 16*a^5*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/16*((2*b^3*c^2 - 8*a*b*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^3 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^2*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b*c^2 - 2*(b^2 - 4*a*c)*b*c^2)*(a*b^2 - 4*a^2*c)^2*d - 2*(2*a*b^2*c^2 - 8*a^2*c^3 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b*c - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*c^2 - 2*(b^2 - 4*a*c)*a*c^2)*(a*b^2 - 4*a^2*c)^2*f - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 14*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*a*b^6*c + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 28*a^2*b^4*c^2 - 96*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 48*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 10*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 128*a^3*b^2*c^3 + 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 192*a^4*c^4 - 2*(b^2 - 4*a*c)*a*b^4*c + 20*(b^2 - 4*a*c)*a^2*b^2*c^2 - 48*(b^2 - 4*a*c)*a^3*c^3)*d*abs(a*b^2 - 4*a^2*c) - 2*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5 - 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*a^2*b^5*c + 16*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^2 + 8*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 16*a^3*b^3*c^2 - 4*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 32*a^4*b*c^3 - 2*(b^2 - 4*a*c)*a^2*b^3*c + 8*(b^2 - 4*a*c)*a^3*b*c^2)*f*abs(a*b^2 - 4*a^2*c) + (2*a^2*b^7*c^2 - 40*a^3*b^5*c^3 + 224*a^4*b^3*c^4 - 384*a^5*b*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^7 + 20*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^6*c - 112*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c^2 + 192*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b*c^3 + 96*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^2*c^3 + 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^3 - 48*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^4 - 2*(b^2 - 4*a*c)*a^2*b^5*c^2 + 32*(b^2 - 4*a*c)*a^3*b^3*c^3 - 96*(b^2 - 4*a*c)*a^4*b*c^4)*d + 4*(2*a^3*b^6*c^2 - 16*a^4*b^4*c^3 + 32*a^5*b^2*c^4 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^6 + 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^4*c + 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^5*c - 16*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^5*b^2*c^2 - 8*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b^3*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b

```
^2 - 4*a*c)*c)*a^3*b^4*c^2 + 4*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^
2 - 4*a*c)*c)*a^4*b^2*c^3 - 2*(b^2 - 4*a*c)*a^3*b^4*c^2 + 8*(b^2 - 4*a*c)*a
^4*b^2*c^3)*f)*arctan(2*sqrt(1/2)*x/sqrt((a*b^3 - 4*a^2*b*c - sqrt((a*b^3 -
 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*
a^2*c^2)))/((a^3*b^6 - 12*a^4*b^4*c - 2*a^3*b^5*c + 48*a^5*b^2*c^2 + 16*a^4
*b^3*c^2 + a^3*b^4*c^2 - 64*a^6*c^3 - 32*a^5*b*c^3 - 8*a^4*b^2*c^3 + 16*a^5
*c^4)*abs(a*b^2 - 4*a^2*c)*abs(c)) - 1/4*((b^3*c^2 - 4*a*b*c^3 - 2*b^2*c^3
+ b*c^4 + (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqrt(b^2 - 4*a*c))*e*abs(a*b^
2 - 4*a^2*c) - (a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b^4*c^3 + 16*a^3*b*c^4 + 8*
a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 - (a*b^4*c^2 - 4*a^2*b^2*c^3 - 2*a*b^
3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e)*log(x^2 + 1/2*(a*b^3 - 4*a^2*b*c +
 sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*c)*(a*b^2*c - 4*a^2*c^2)))
/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2*a*b^3*c + 16*a^3*c^2 + 8*
a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2 - 4*a^2*c)) - 1/4*((b^3*c^
2 - 4*a*b*c^3 - 2*b^2*c^3 + b*c^4 - (b^2*c^2 - 4*a*c^3 - 2*b*c^3 + c^4)*sqr
t(b^2 - 4*a*c))*e*abs(a*b^2 - 4*a^2*c) - (a*b^5*c^2 - 8*a^2*b^3*c^3 - 2*a*b
^4*c^3 + 16*a^3*b*c^4 + 8*a^2*b^2*c^4 + a*b^3*c^4 - 4*a^2*b*c^5 - (a*b^4*c^
2 - 4*a^2*b^2*c^3 - 2*a*b^3*c^3 + a*b^2*c^4)*sqrt(b^2 - 4*a*c))*e)*log(x^2
+ 1/2*(a*b^3 - 4*a^2*b*c - sqrt((a*b^3 - 4*a^2*b*c)^2 - 4*(a^2*b^2 - 4*a^3*
c)*(a*b^2*c - 4*a^2*c^2)))/(a*b^2*c - 4*a^2*c^2))/((a*b^4 - 8*a^2*b^2*c - 2
*a*b^3*c + 16*a^3*c^2 + 8*a^2*b*c^2 + a*b^2*c^2 - 4*a^2*c^3)*c^2*abs(a*b^2
- 4*a^2*c))
```

Mupad [B] (verification not implemented)

Time = 8.62 (sec) , antiderivative size = 4707, normalized size of antiderivative = 12.79

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^3}\,dx = \text{Too large to display}$$

```
[In] int((a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x + b*e*x^3 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^3,x)
```

```
[Out] symsum(log((5*b^3*c^4*d^3 + 8*a^3*c^4*f^3 - 96*a^2*c^5*d*e^2 + 72*a^2*c^5*d
^2*f - 3*b^4*c^3*d^2*f + 6*a^2*b^2*c^3*f^3 - 36*a*b*c^5*d^3 + 16*a*b^2*c^4*
d*e^2 + 18*a*b^2*c^4*d^2*f + 3*a*b^3*c^3*d*f^2 - 60*a^2*b*c^4*d*f^2 + 16*a^
2*b*c^4*e^2*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) -
 root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3
*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 -
256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072
*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 +
 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 81
92*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z
^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4
*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10
```

*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*((32*a*b^5*c^3*d*e - 512*a^4*c^5*e*f + 1024*a^3*b*c^5*d*e - 384*a^2*b^3*c^4*d*e + 32*a^2*b^4*c^3*e*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^10*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*((x*(1024*a^5*c^6*e - 16*a^2*b^6*c^3*e + 192*a^3*b^4*c^4*e - 768*a^4*b^2*c^5*e))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) - (6144*a^5*c^6*d - 288*a^2*b^6*c^3*d + 1920*a^3*b^4*c^4*d - 5632*a^4*b^2*c^5*d + 16*a^2*b^7*c^2*f - 192*a^3*b^5*c^3*f + 768*a^4*b^3*c^4*f + 16*a*b^8*c^2*d - 1024*a^5*b*c^5*f)/(8*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2)) + (root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 - 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 3072*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2 + 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c

*d^2*z^2 - 8192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*
b^2*c^4*e^2*z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 6144
0*a^4*b^3*c^4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^
2 - 32*a*b^10*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^1
1*d^2*z^2 - 4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*
d*e*f*z - 768*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^
2*e*z + 1536*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*
c^4*d^2*e*z + 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c
^5*d^2*e*z + 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^
2*f - 192*a^3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3
 - 960*a^2*b*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 20
16*a^2*b*c^4*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3
*c^2*e^2*f^2 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2
*f^2 - 16*b^5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c
*d^2*f^2 - 9*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c
^4*e^4 - 25*b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k)*x*(4096*a^6*b*c^6 + 16*a^
2*b^9*c^2 - 256*a^3*b^7*c^3 + 1536*a^4*b^5*c^4 - 4096*a^5*b^3*c^5))/(2*(a^2
*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) + (x*(b^6*c^3*d^2 - 28
8*a^3*c^6*d^2 + 32*a^4*c^5*f^2 - 18*a*b^4*c^4*d^2 + 64*a^3*b*c^5*e^2 + 128*
a^2*b^2*c^5*d^2 - 16*a^2*b^3*c^4*e^2 + 10*a^2*b^4*c^3*f^2 - 48*a^3*b^2*c^4*
f^2 + 2*a*b^5*c^3*d*f + 160*a^3*b*c^5*d*f - 48*a^2*b^3*c^4*d*f))/(2*(a^2*b^
6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))) - (x*(16*a^2*c^5*e^3 - b^
3*c^4*d^2*e + 12*a*b*c^5*d^2*e - 24*a^2*c^5*d*e*f + 8*a^2*b*c^4*e*f^2 - 2*a
*b^2*c^4*d*e*f))/(2*(a^2*b^6 - 64*a^5*c^3 - 12*a^3*b^4*c + 48*a^4*b^2*c^2))
)*root(1572864*a^8*b^2*c^5*z^4 - 983040*a^7*b^4*c^4*z^4 + 327680*a^6*b^6*c^
3*z^4 - 61440*a^5*b^8*c^2*z^4 + 6144*a^4*b^10*c*z^4 - 1048576*a^9*c^6*z^4 -
 256*a^3*b^12*z^4 + 576*a^2*b^8*c*d*f*z^2 + 24576*a^5*b^2*c^4*d*f*z^2 - 307
2*a^3*b^6*c^2*d*f*z^2 + 2048*a^4*b^4*c^3*d*f*z^2 + 12288*a^6*b*c^4*f^2*z^2
+ 61440*a^5*b*c^5*d^2*z^2 - 49152*a^6*c^5*d*f*z^2 + 432*a*b^9*c*d^2*z^2 - 8
192*a^5*b^3*c^3*f^2*z^2 + 1536*a^4*b^5*c^2*f^2*z^2 + 24576*a^5*b^2*c^4*e^2*
z^2 - 6144*a^4*b^4*c^3*e^2*z^2 + 512*a^3*b^6*c^2*e^2*z^2 - 61440*a^4*b^3*c^
4*d^2*z^2 + 24064*a^3*b^5*c^3*d^2*z^2 - 4608*a^2*b^7*c^2*d^2*z^2 - 32*a*b^1
0*d*f*z^2 - 32768*a^6*c^5*e^2*z^2 - 16*a^2*b^9*f^2*z^2 - 16*b^11*d^2*z^2 -
4096*a^4*b*c^4*d*e*f*z + 64*a*b^7*c*d*e*f*z + 3072*a^3*b^3*c^3*d*e*f*z - 76
8*a^2*b^5*c^2*d*e*f*z + 32*a^2*b^6*c*e*f^2*z - 672*a*b^6*c^2*d^2*e*z + 1536
*a^4*b^2*c^3*e*f^2*z - 384*a^3*b^4*c^2*e*f^2*z - 15872*a^3*b^2*c^4*d^2*e*z
+ 4992*a^2*b^4*c^3*d^2*e*z - 2048*a^5*c^4*e*f^2*z + 18432*a^4*c^5*d^2*e*z +
 32*b^8*c*d^2*e*z - 32*a*b^4*c^2*d*e^2*f + 192*a^2*b^2*c^3*d*e^2*f - 192*a^
3*b*c^3*e^2*f^2 + 198*a*b^4*c^2*d^2*f^2 + 144*a^2*b^3*c^2*d*f^3 - 960*a^2*b
*c^4*d^2*e^2 + 240*a*b^3*c^3*d^2*e^2 + 768*a^3*c^4*d*e^2*f + 2016*a^2*b*c^4
*d^3*f - 496*a*b^3*c^3*d^3*f + 224*a^3*b*c^3*d*f^3 - 16*a^2*b^3*c^2*e^2*f^2
 - 960*a^2*b^2*c^3*d^2*f^2 - 18*a*b^5*c*d*f^3 - 288*a^3*c^4*d^2*f^2 - 16*b^
5*c^2*d^2*e^2 - 24*a^3*b^2*c^2*f^4 + 30*b^5*c^2*d^3*f - 9*b^6*c*d^2*f^2 - 9
*a^2*b^4*c*f^4 + 360*a*b^2*c^4*d^4 - 16*a^4*c^3*f^4 - 256*a^3*c^4*e^4 - 25*
b^4*c^3*d^4 - 1296*a^2*c^5*d^4, z, k), k, 1, 4) + ((b*e)/(2*(4*a*c - b^2))

```
+ (c*e*x^2)/(4*a*c - b^2) + (x*(2*a*c*d - b^2*d + a*b*f))/(2*a*(4*a*c - b^2
)) - (c*x^3*(b*d - 2*a*f))/(2*a*(4*a*c - b^2)))/(a + b*x^2 + c*x^4)
```

3.66 $\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4} dx$

Optimal result

Integrand size = 63, antiderivative size = 621

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4} dx$$
$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)-ab^2\left(30cd-\sqrt{b^2-4ac}f\right)+4a^2c(42cd\ldots\right.}{8\sqrt{2}a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}$$

```
[Out] -1/4*e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+1/4*x*(b^2*d-2*a*c*d-a*b*
f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^2+3/2*c*e*(2*c*x^2+b)/
(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)+1/8*x*(3*b^4*d-25*a*b^2*c*d+28*a^2*c^2*d+a*b
^3*f+8*a^2*b*c*f+c*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)*x^2)/a^2/(-4*a*c
+b^2)^2/(c*x^4+b*x^2+a)-6*c^2*e*arctanh((2*c*x^2+b)/(-4*a*c+b^2)^(1/2))/(-4
*a*c+b^2)^(5/2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/2))
*c^(1/2)*(3*b^4*d+b^3*(a*f+3*d*(-4*a*c+b^2)^(1/2))-4*a*b*c*(13*a*f+6*d*(-4*
```

```
a*c+b^2)^(1/2))-a*b^2*(30*c*d-f*(-4*a*c+b^2)^(1/2))+4*a^2*c*(42*c*d+5*f*(-4
*a*c+b^2)^(1/2)))/a^2/(-4*a*c+b^2)^(5/2)*2^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/
2)+1/16*arctan(x*2^(1/2)*c^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(1/2))*c^(1/2)*(3*b
^3*d-24*a*b*c*d+a*b^2*f+20*a^2*c*f+(52*a^2*b*c*f-168*a^2*c^2*d-a*b^3*f+30*a
*b^2*c*d-3*b^4*d)/(-4*a*c+b^2)^(1/2))/a^2/(-4*a*c+b^2)^2*2^(1/2)/(b+(-4*a*c
+b^2)^(1/2))^(1/2)
```

Rubi [A] (verified)

Time = 2.64 (sec) , antiderivative size = 621, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.159, Rules used = {1600, 1687, 1192, 1180, 211, 12, 1121, 628, 632, 212}

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4}\,dx$$

$$= \frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{\sqrt{b^2-4ac}+b}}\right)\left(-\frac{-52a^2bcf+168a^2c^2d+ab^3f-30ab^2cd+3b^4d}{\sqrt{b^2-4ac}}+20a^2cf+ab^2f-24abcd+3b^3d\right)}{8\sqrt{2}a^2\left(b^2-4ac\right)^2\sqrt{\sqrt{b^2-4ac}+b}}$$

$$+\frac{\sqrt{c}\arctan\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)\left(4a^2c\left(5f\sqrt{b^2-4ac}+42cd\right)-ab^2\left(30cd-f\sqrt{b^2-4ac}\right)-4abc\left(6d\sqrt{b^2-4ac}\right.\right.}{8\sqrt{2}a^2\left(b^2-4ac\right)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}}$$

$$+\frac{x(cx^2(20a^2cf+ab^2f-24abcd+3b^3d)+8a^2bcf+28a^2c^2d+ab^3f-25ab^2cd+3b^4d)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}$$

$$-\frac{6c^2e\,\text{arctanh}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{\left(b^2-4ac\right)^{5/2}}+\frac{x(cx^2(bd-2af)-abf-2acd+b^2d)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

$$+\frac{3ce(b+2cx^2)}{2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)}-\frac{e(b+2cx^2)}{4\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2}$$

```
[In] Int[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^4,x]
```

```
[Out] -1/4*(e*(b + 2*c*x^2))/((b^2 - 4*a*c)*(a + b*x^2 + c*x^4)^2) + (x*(b^2*d -
2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(4*a*(b^2 - 4*a*c)*(a + b*x^2 + c*x
^4)^2) + (3*c*e*(b + 2*c*x^2))/(2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (x
*(3*b^4*d - 25*a*b^2*c*d + 28*a^2*c^2*d + a*b^3*f + 8*a^2*b*c*f + c*(3*b^3*
d - 24*a*b*c*d + a*b^2*f + 20*a^2*c*f)*x^2))/(8*a^2*(b^2 - 4*a*c)^2*(a + b*
x^2 + c*x^4)) + (Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d + a*f) - 4*a
*b*c*(6*Sqrt[b^2 - 4*a*c]*d + 13*a*f) - a*b^2*(30*c*d - Sqrt[b^2 - 4*a*c]*f
) + 4*a^2*c*(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sq
rt[b - Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[2]*a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqr
t[b^2 - 4*a*c]]) + (Sqrt[c]*(3*b^3*d - 24*a*b*c*d + a*b^2*f + 20*a^2*c*f -
(3*b^4*d - 30*a*b^2*c*d + 168*a^2*c^2*d + a*b^3*f - 52*a^2*b*c*f)/Sqrt[b^2
- 4*a*c])*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + Sqrt[b^2 - 4*a*c]]])/(8*Sqrt[
```

```
2]*a^2*(b^2 - 4*a*c)^2*Sqrt[b + Sqrt[b^2 - 4*a*c]]) - (6*c^2*e*ArcTanh[(b +
 2*c*x^2)/Sqrt[b^2 - 4*a*c]])/(b^2 - 4*a*c)^(5/2)
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 211

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[a/b, 2]/a)*ArcTan[x/R
t[a/b, 2]], x] /; FreeQ[{a, b}, x] && PosQ[a/b]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 628

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^(p + 1)/((p + 1)*(b^2 - 4*a*c))), x] - Dist[2*c*((2*p +
 3)/((p + 1)*(b^2 - 4*a*c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2] && Int
egerQ[4*p]
```

Rule 632

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> Dist[-2, Subst[I
nt[1/Simp[b^2 - 4*a*c - x^2, x], x], x, b + 2*c*x], x] /; FreeQ[{a, b, c},
x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1180

```
Int[((d_) + (e_.)*(x_)^2)/((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4), x_Symbol] :
> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 + (2*c*d - b*e)/(2*q), Int[1/(b/2
 - q/2 + c*x^2), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(b/2 + q/2
+ c*x^2), x], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && Ne
Q[c*d^2 - a*e^2, 0] && PosQ[b^2 - 4*a*c]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{\left(a+bx^2+cx^4\right)^3}\,dx \\
&= \int \frac{ex}{\left(a+bx^2+cx^4\right)^3}\,dx + \int \frac{d+fx^2}{\left(a+bx^2+cx^4\right)^3}\,dx \\
&= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} \\
&\quad - \frac{\int \frac{-3b^2d+14acd-abf-5c(bd-2af)x^2}{\left(a+bx^2+cx^4\right)^2}\,dx}{4a\left(b^2-4ac\right)} + e\int \frac{x}{\left(a+bx^2+cx^4\right)^3}\,dx \\
&= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^2} \\
&\quad + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)\,x^2)}{8a^2\left(b^2-4ac\right)^2\left(a+bx^2+cx^4\right)} \\
&\quad + \frac{\int \frac{3b^4d-27ab^2cd+84a^2c^2d+ab^3f-16a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2}{a+bx^2+cx^4}\,dx}{8a^2\left(b^2-4ac\right)^2} \\
&\quad + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{\left(a+bx+cx^2\right)^3}\,dx,x,x^2\right)
\end{aligned}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$- \frac{(3ce)\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^2}\,dx, x, x^2\right)}{2(b^2-4ac)}$$
$$+ \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^2}$$
$$+ \frac{\left(c\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^2}\,dx}{16a^2(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$+ \frac{(3c^2e)\text{Subst}\left(\int \frac{1}{a+bx+cx^2}\,dx, x, x^2\right)}{(b^2-4ac)^2}$$

$$= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2}$$
$$+ \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}}$$
$$+ \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}}$$
$$- \frac{(6c^2e)\text{Subst}\left(\int \frac{1}{b^2-4ac-x^2}\,dx, x, b+2cx^2\right)}{(b^2-4ac)^2}$$

$$
\begin{aligned}
&= -\frac{e(b+2cx^2)}{4(b^2-4ac)(a+bx^2+cx^4)^2} \\
&\quad + \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{4a(b^2-4ac)(a+bx^2+cx^4)^2} + \frac{3ce(b+2cx^2)}{2(b^2-4ac)^2(a+bx^2+cx^4)} \\
&\quad + \frac{x(3b^4d-25ab^2cd+28a^2c^2d+ab^3f+8a^2bcf+c(3b^3d-24abcd+ab^2f+20a^2cf)x^2)}{8a^2(b^2-4ac)^2(a+bx^2+cx^4)} \\
&\quad + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf+\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b-\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\sqrt{c}\left(3b^3d-24abcd+ab^2f+20a^2cf-\frac{3b^4d-30ab^2cd+168a^2c^2d+ab^3f-52a^2bcf}{\sqrt{b^2-4ac}}\right)\tan^{-1}\left(\frac{\sqrt{2}\sqrt{c}x}{\sqrt{b+\sqrt{b^2-4ac}}}\right)}{8\sqrt{2}a^2(b^2-4ac)^2\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad - \frac{6c^2e\tanh^{-1}\left(\frac{b+2cx^2}{\sqrt{b^2-4ac}}\right)}{(b^2-4ac)^{5/2}}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 2.03 (sec) , antiderivative size = 625, normalized size of antiderivative = 1.01

$$
\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4}\,dx
$$

$$
\begin{aligned}
&= \frac{1}{16}\left(\frac{4ab(e+fx)-4bdx(b+cx^2)+8acx(d+x(e+fx))}{a(-b^2+4ac)(a+bx^2+cx^4)^2}\right. \\
&\quad + \frac{6b^3dx(b+cx^2)+2abx(-25bcd+b^2f-24c^2dx^2+bcfx^2)+8a^2c(b(3e+2fx)+cx(7d+6ex+5fx^2))}{a^2(b^2-4ac)^2(a+bx^2+cx^4)} \\
&\quad + \frac{\sqrt{2}\sqrt{c}\left(3b^4d+b^3\left(3\sqrt{b^2-4ac}d+af\right)-4abc\left(6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(-30cd+\sqrt{b^2-4ac}f\right)+4a^2 \cdots\right.}{a^2(b^2-4ac)^{5/2}\sqrt{b-\sqrt{b^2-4ac}}} \\
&\quad + \frac{\sqrt{2}\sqrt{c}\left(-3b^4d+b^3\left(3\sqrt{b^2-4ac}d-af\right)+4abc\left(-6\sqrt{b^2-4ac}d+13af\right)+ab^2\left(30cd+\sqrt{b^2-4ac}f\right)+4 \cdots\right.}{a^2(b^2-4ac)^{5/2}\sqrt{b+\sqrt{b^2-4ac}}} \\
&\quad \left. + \frac{48c^2e\log\left(-b+\sqrt{b^2-4ac}-2cx^2\right)}{(b^2-4ac)^{5/2}} - \frac{48c^2e\log\left(b+\sqrt{b^2-4ac}+2cx^2\right)}{(b^2-4ac)^{5/2}}\right)
\end{aligned}
$$

```
[In] Integrate[(a*d + a*e*x + (b*d + a*f)*x^2 + b*e*x^3 + (c*d + b*f)*x^4 + c*e*
x^5 + c*f*x^6)/(a + b*x^2 + c*x^4)^4,x]
```

```
[Out] ((4*a*b*(e + f*x) - 4*b*d*x*(b + c*x^2) + 8*a*c*x*(d + x*(e + f*x)))/(a*(-b
^2 + 4*a*c)*(a + b*x^2 + c*x^4)^2) + (6*b^3*d*x*(b + c*x^2) + 2*a*b*x*(-25*
b*c*d + b^2*f - 24*c^2*d*x^2 + b*c*f*x^2) + 8*a^2*c*(b*(3*e + 2*f*x) + c*x*
```

```
(7*d + 6*e*x + 5*f*x^2)))/(a^2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x^4)) + (Sqrt
[2]*Sqrt[c]*(3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d + a*f) - 4*a*b*c*(6*Sqrt[
b^2 - 4*a*c]*d + 13*a*f) + a*b^2*(-30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2*c*
(42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b - Sqrt[
b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b - Sqrt[b^2 - 4*a*c]]) + (Sq
rt[2]*Sqrt[c]*(-3*b^4*d + b^3*(3*Sqrt[b^2 - 4*a*c]*d - a*f) + 4*a*b*c*(-6*S
qrt[b^2 - 4*a*c]*d + 13*a*f) + a*b^2*(30*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2
*c*(-42*c*d + 5*Sqrt[b^2 - 4*a*c]*f))*ArcTan[(Sqrt[2]*Sqrt[c]*x)/Sqrt[b + S
qrt[b^2 - 4*a*c]]])/(a^2*(b^2 - 4*a*c)^(5/2)*Sqrt[b + Sqrt[b^2 - 4*a*c]]) +
 (48*c^2*e*Log[-b + Sqrt[b^2 - 4*a*c] - 2*c*x^2])/(b^2 - 4*a*c)^(5/2) - (48
*c^2*e*Log[b + Sqrt[b^2 - 4*a*c] + 2*c*x^2])/(b^2 - 4*a*c)^(5/2))/16
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 0.56 (sec) , antiderivative size = 607, normalized size of antiderivative = 0.98

method	result
risch	$\frac{\frac{c^{2}\left(20a^{2}cf+a\,b^{2}f-24abcd+3b^{3}d\right)x^{7}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{3c^{3}e\,x^{6}}{16a^{2}c^{2}-8a\,b^{2}c+b^{4}}+\frac{c\left(28a^{2}bcf+28a^{2}c^{2}d+2a\,b^{3}f-49a\,b^{2}cd+6d\,b^{4}\right)x^{5}}{8a^{2}\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{9b\,c^{2}e\,x^{4}}{2\left(16a^{2}c^{2}-8a\,b^{2}c+b^{4}\right)}+\frac{(36a^{3}c^{2}f \cdots}{\cdots}}{(c\,x^{4}+b\,x^{2} \cdots}$
default	Expression too large to display

```
[In] int((d*a+a*e*x+(a*f+b*d)*x^2+e*x^3*b+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(c*x^4+
b*x^2+a)^4,x,method=_RETURNVERBOSE)
```

```
[Out] (1/8*c^2*(20*a^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+
b^4)*x^7+3*c^3*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^6+1/8/a^2*c*(28*a^2*b*c*f+28*
a^2*c^2*d+2*a*b^3*f-49*a*b^2*c*d+6*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)*x^5+9/
2*b*c^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^4+1/8*(36*a^3*c^2*f+5*a^2*b^2*c*f-4*
a^2*b*c^2*d+a*b^4*f-20*a*b^3*c*d+3*b^5*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*x^
3+(5*a*c+b^2)*c*e/(16*a^2*c^2-8*a*b^2*c+b^4)*x^2+1/8*(16*a^2*b*c*f+44*a^2*c
^2*d-a*b^3*f-37*a*b^2*c*d+5*b^4*d)/(16*a^2*c^2-8*a*b^2*c+b^4)/a*x+1/4*b*(10
*a*c-b^2)*e/(16*a^2*c^2-8*a*b^2*c+b^4))/(c*x^4+b*x^2+a)^2+1/16*sum((c*(20*a
^2*c*f+a*b^2*f-24*a*b*c*d+3*b^3*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4)*_R^2+48*c
^2*e/(16*a^2*c^2-8*a*b^2*c+b^4)*_R-(16*a^2*b*c*f-84*a^2*c^2*d-a*b^3*f+27*a*
b^2*c*d-3*b^4*d)/a^2/(16*a^2*c^2-8*a*b^2*c+b^4))/(2*_R^3*c+_R*b)*ln(x-_R),_
R=RootOf(_Z^4*c+_Z^2*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4}\,dx = \text{Timed out}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(
c*x^4+b*x^2+a)^4,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4}\,dx = \text{Timed out}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x**2+b*e*x**3+(b*f+c*d)*x**4+c*e*x**5+c*f*x*
*6)/(c*x**4+b*x**2+a)**4,x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{ad+aex+(bd+af)x^2+bex^3+(cd+bf)x^4+cex^5+cfx^6}{(a+bx^2+cx^4)^4}\,dx = \int \frac{cfx^6+cex^5+bex^3+(cd+bf)x^4+aex+(bd+af)x^2+ad}{(cx^4+bx^2+a)^4}\,dx$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(
c*x^4+b*x^2+a)^4,x, algorithm="maxima")
```

```
[Out] 1/8*(24*a^2*c^3*e*x^6 + 36*a^2*b*c^2*e*x^4 + (3*(b^3*c^2 - 8*a*b*c^3)*d + (
a*b^2*c^2 + 20*a^2*c^3)*f)*x^7 + ((6*b^4*c - 49*a*b^2*c^2 + 28*a^2*c^3)*d +
 2*(a*b^3*c + 14*a^2*b*c^2)*f)*x^5 + 8*(a^2*b^2*c + 5*a^3*c^2)*e*x^2 + ((3*
b^5 - 20*a*b^3*c - 4*a^2*b*c^2)*d + (a*b^4 + 5*a^2*b^2*c + 36*a^3*c^2)*f)*x
^3 - 2*(a^2*b^3 - 10*a^3*b*c)*e + ((5*a*b^4 - 37*a^2*b^2*c + 44*a^3*c^2)*d
- (a^2*b^3 - 16*a^3*b*c)*f)*x)/((a^2*b^4*c^2 - 8*a^3*b^2*c^3 + 16*a^4*c^4)*
x^8 + a^4*b^4 - 8*a^5*b^2*c + 16*a^6*c^2 + 2*(a^2*b^5*c - 8*a^3*b^3*c^2 + 1
6*a^4*b*c^3)*x^6 + (a^2*b^6 - 6*a^3*b^4*c + 32*a^5*c^3)*x^4 + 2*(a^3*b^5 -
8*a^4*b^3*c + 16*a^5*b*c^2)*x^2) + 1/8*integrate((48*a^2*c^2*e*x + (3*(b^3*
c - 8*a*b*c^2)*d + (a*b^2*c + 20*a^2*c^2)*f)*x^2 + 3*(b^4 - 9*a*b^2*c + 28*
a^2*c^2)*d + (a*b^3 - 16*a^2*b*c)*f)/(c*x^4 + b*x^2 + a), x)/(a^2*b^4 - 8*a
^3*b^2*c + 16*a^4*c^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5280 vs. 2(561) = 1122.

Time = 1.60 (sec) , antiderivative size = 5280, normalized size of antiderivative = 8.50

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^4} dx = \text{Too large to display}$$

```
[In] integrate((a*d+a*e*x+(a*f+b*d)*x^2+b*e*x^3+(b*f+c*d)*x^4+c*e*x^5+c*f*x^6)/(
c*x^4+b*x^2+a)^4,x, algorithm="giac")
```

```
[Out] -3*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*a*c)*e*log(x^2 + 1/2*(a
^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*
b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^
2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3))/((b^8 - 16*a*b^
6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*c^2 - 256*a^3*b^2*c^3 -
 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a^3*b*c^4 + 48*a^2*b^2*c
^4 - 64*a^3*c^5)*c^2) - 3*(b^2*c^4 - 4*a*c^5 - 2*b*c^5 + c^6)*sqrt(b^2 - 4*
a*c)*e*log(x^2 + 1/2*(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5
- 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a
^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^
4*c^3))/((b^8 - 16*a*b^6*c - 2*b^7*c + 96*a^2*b^4*c^2 + 24*a*b^5*c^2 + b^6*
c^2 - 256*a^3*b^2*c^3 - 96*a^2*b^3*c^3 - 12*a*b^4*c^3 + 256*a^4*c^4 + 128*a
^3*b*c^4 + 48*a^2*b^2*c^4 - 64*a^3*c^5)*c^2) + 1/32*(3*(sqrt(2)*sqrt(b*c +
sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6
*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7*c - 2*b^8*c + 116*sqrt(2
)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + 26*sqrt(2)*sqrt(b*c + sqrt(
b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c^2
 + 34*a*b^6*c^2 + 2*b^7*c^2 - 368*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a
^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 - 13*s
qrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^3 - 232*a^2*b^4*c^3 - 30*a*b
^5*c^3 + 448*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^4 + 224*sqrt(2)*
sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 64*sqrt(2)*sqrt(b*c + sqrt(b^2
- 4*a*c)*c)*a^2*b^2*c^4 + 736*a^3*b^2*c^4 + 176*a^2*b^3*c^4 - 112*sqrt(2)*s
qrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^5 - 896*a^4*c^5 - 352*a^3*b*c^5 - sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^7 + 15*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 2*sqrt(2)*sqrt(b^2
- 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*b^6*c - 88*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - 22*sqrt(2)*sqrt(b^2 - 4*a*
c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - sqrt(2)*sqrt(b^2 - 4*a*c)*sq
rt(b*c + sqrt(b^2 - 4*a*c)*c)*b^5*c^2 + 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 - 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c
+ sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 + 2*(b^2 - 4*a*c)*b^6*c - 26*(b^2 - 4*a*c)
```

*a*b^4*c^2 - 2*(b^2 - 4*a*c)*b^5*c^2 + 128*(b^2 - 4*a*c)*a^2*b^2*c^3 + 22*(
b^2 - 4*a*c)*a*b^3*c^3 - 224*(b^2 - 4*a*c)*a^3*c^4 - 88*(b^2 - 4*a*c)*a^2*b
*c^4)*d + (sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(
b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*
c)*c)*a*b^6*c - 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3
*b^3*c^2 + 40*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)
*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + 48*a^2*b^5*c^2 + 2*a*b^6*c^2 -
256*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b
*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c + sqrt(b^2 - 4*
a*c)*c)*a^2*b^3*c^3 - 288*a^3*b^3*c^3 - 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*
c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 + 512*a^4*b*c^4 + 64*a^3*b^2*c^4 + 320*a
^4*c^5 - sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^6 +
22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c + 2*
sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^5*c - 32*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 - 36*sqrt
(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 - sqrt(2)
*sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 - 160*sqrt(2)*
sqrt(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^4*c^3 - 80*sqrt(2)*sqrt
(b^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 + 18*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 + 40*sqrt(2)*sqrt(b
^2 - 4*a*c)*sqrt(b*c + sqrt(b^2 - 4*a*c)*c)*a^3*c^4 + 2*(b^2 - 4*a*c)*a*b^5
c - 40(b^2 - 4*a*c)*a^2*b^3*c^2 - 2*(b^2 - 4*a*c)*a*b^4*c^2 + 128*(b^2 -
4*a*c)*a^3*b*c^3 + 36*(b^2 - 4*a*c)*a^2*b^2*c^3 + 80*(b^2 - 4*a*c)*a^3*c^4)
*f)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 + sqrt(
(a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^
5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^
2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 +
24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b
^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c))
+ 1/32*(3*(sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^8 - 17*sqrt(2)*sqrt(b
*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*
c)*b^7*c + 2*b^8*c + 116*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^
2 + 26*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 + sqrt(2)*sqrt(b*c
- sqrt(b^2 - 4*a*c)*c)*b^6*c^2 - 34*a*b^6*c^2 - 2*b^7*c^2 - 368*sqrt(2)*sq
rt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2
- 4*a*c)*c)*a^2*b^3*c^3 - 13*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4
*c^3 + 232*a^2*b^4*c^3 + 30*a*b^5*c^3 + 448*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4
*a*c)*c)*a^4*c^4 + 224*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 +
64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^4 - 736*a^3*b^2*c^4 -
176*a^2*b^3*c^4 - 112*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^5 + 896
*a^4*c^5 + 352*a^3*b*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 -
4*a*c)*c)*b^7 - 15*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c
)*a*b^5*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^6
*c + 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c
^2 + 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2

+ sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*b^5*c^2 - 176*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 88*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 11*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^3*c^3 + 44*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b*c^4 - 2*(b^2 - 4*a*c)*b^6*c + 26*(b^2 - 4*a*c)*a*b^4*c^2 + 2*(b^2 - 4*a*c)*b^5*c^2 - 128*(b^2 - 4*a*c)*a^2*b^2*c^3 - 22*(b^2 - 4*a*c)*a*b^3*c^3 + 224*(b^2 - 4*a*c)*a^3*c^4 + 88*(b^2 - 4*a*c)*a^2*b*c^4)*d + (sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^7 - 24*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^5*c - 2*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6*c + 2*a*b^7*c + 144*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^3*c^2 + 40*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c^2 + sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c^2 - 48*a^2*b^5*c^2 - 2*a*b^6*c^2 - 256*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*b*c^3 - 128*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^3 - 20*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^3 + 288*a^3*b^3*c^3 + 44*a^2*b^4*c^3 + 64*sqrt(2)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^4 - 512*a^4*b*c^4 - 64*a^3*b^2*c^4 - 320*a^4*c^5 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^6 - 22*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^4*c - 2*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^5*c + 32*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b^2*c^2 + 36*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^3*c^2 + sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a*b^4*c^2 + 160*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^4*c^3 + 80*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*b*c^3 - 18*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^2*b^2*c^3 - 40*sqrt(2)*sqrt(b^2 - 4*a*c)*sqrt(b*c - sqrt(b^2 - 4*a*c)*c)*a^3*c^4 - 2*(b^2 - 4*a*c)*a*b^5*c + 40*(b^2 - 4*a*c)*a^2*b^3*c^2 + 2*(b^2 - 4*a*c)*a*b^4*c^2 - 128*(b^2 - 4*a*c)*a^3*b*c^3 - 36*(b^2 - 4*a*c)*a^2*b^2*c^3 - 80*(b^2 - 4*a*c)*a^3*c^4)*f)*arctan(2*sqrt(1/2)*x/sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2 - sqrt((a^2*b^5 - 8*a^3*b^3*c + 16*a^4*b*c^2)^2 - 4*(a^3*b^4 - 8*a^4*b^2*c + 16*a^5*c^2)*(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/(a^2*b^4*c - 8*a^3*b^2*c^2 + 16*a^4*c^3)))/((a^3*b^8 - 16*a^4*b^6*c - 2*a^3*b^7*c + 96*a^5*b^4*c^2 + 24*a^4*b^5*c^2 + a^3*b^6*c^2 - 256*a^6*b^2*c^3 - 96*a^5*b^3*c^3 - 12*a^4*b^4*c^3 + 256*a^7*c^4 + 128*a^6*b*c^4 + 48*a^5*b^2*c^4 - 64*a^6*c^5)*abs(c)) + 1/8*(3*b^3*c^2*d*x^7 - 24*a*b*c^3*d*x^7 + a*b^2*c^2*f*x^7 + 20*a^2*c^3*f*x^7 + 24*a^2*c^3*e*x^6 + 6*b^4*c*d*x^5 - 49*a*b^2*c^2*d*x^5 + 28*a^2*c^3*d*x^5 + 2*a*b^3*c*f*x^5 + 28*a^2*b*c^2*f*x^5 + 36*a^2*b*c^2*e*x^4 + 3*b^5*d*x^3 - 20*a*b^3*c*d*x^3 - 4*a^2*b*c^2*d*x^3 + a*b^4*f*x^3 + 5*a^2*b^2*c*f*x^3 + 36*a^3*c^2*f*x^3 + 8*a^2*b^2*c*e*x^2 + 40*a^3*c^2*e*x^2 + 5*a*b^4*d*x - 37*a^2*b^2*c*d*x + 44*a^3*c^2*d*x - a^2*b^3*f*x + 16*a^3*b*c*f*x - 2*a^2*b^3*e + 20*a^3*b*c*e)/((a^2*b^4 - 8*a^3*b^2*c + 16*a^4*c^2)*(c*x^4 + b*x^2 + a)^2)

Mupad [B] (verification not implemented)

Time = 9.34 (sec) , antiderivative size = 8689, normalized size of antiderivative = 13.99

$$\int \frac{ad + aex + (bd + af)x^2 + bex^3 + (cd + bf)x^4 + cex^5 + cfx^6}{(a + bx^2 + cx^4)^4} dx = \text{Too large to display}$$

```
[In] int((a*d + x^2*(b*d + a*f) + x^4*(c*d + b*f) + a*e*x + b*e*x^3 + c*e*x^5 +
c*f*x^6)/(a + b*x^2 + c*x^4)^4,x)
```

```
[Out] ((x^2*(5*a*c^2*e + b^2*c*e))/(b^4 + 16*a^2*c^2 - 8*a*b^2*c) - (b^3*e - 10*a
*b*c*e)/(4*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (x^5*(28*a^2*c^3*d + 6*b^4*c*d
 + 2*a*b^3*c*f - 49*a*b^2*c^2*d + 28*a^2*b*c^2*f))/(8*a^2*(b^4 + 16*a^2*c^2
 - 8*a*b^2*c)) + (x*(5*b^4*d + 44*a^2*c^2*d - a*b^3*f - 37*a*b^2*c*d + 16*a
^2*b*c*f))/(8*a*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (3*c^3*e*x^6)/(b^4 + 16*a
^2*c^2 - 8*a*b^2*c) + (x^3*(3*b^5*d + 36*a^3*c^2*f + a*b^4*f - 20*a*b^3*c*d
 - 4*a^2*b*c^2*d + 5*a^2*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) +
 (9*b*c^2*e*x^4)/(2*(b^4 + 16*a^2*c^2 - 8*a*b^2*c)) + (c*x^7*(20*a^2*c^2*f
+ 3*b^3*c*d - 24*a*b*c^2*d + a*b^2*c*f))/(8*a^2*(b^4 + 16*a^2*c^2 - 8*a*b^2
*c)))/(x^4*(2*a*c + b^2) + a^2 + c^2*x^8 + 2*a*b*x^2 + 2*b*c*x^6) + symsum(
log(root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 471859
20*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4
*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 +
3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^1
0*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*
c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2
 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680
*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8
*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330
240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b
*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*
z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^
5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f
^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 1
1206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^
8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2
- 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^
2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 25
6*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216
*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6
*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350
208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c
^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z
- 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^
3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2
```

*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*(root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a

^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*((768*a^2*b^14*c^2*d - 22020096*a^9*c^9*d - 22272*a^3*b^12*c^3*d + 282624*a^4*b^10*c^4*d - 2027520*a^5*b^8*c^5*d + 8847360*a^6*b^6*c^6*d - 23396352*a^7*b^4*c^7*d + 34603008*a^8*b^2*c^8*d + 256*a^3*b^13*c^2*f - 9216*a^4*b^11*c^3*f + 122880*a^5*b^9*c^4*f - 819200*a^6*b^7*c^5*f + 2949120*a^7*b^5*c^6*f - 5505024*a^8*b^3*c^7*f + 4194304*a^9*b*c^8*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(786432*a^9*c^9*e - 768*a^4*b^10*c^4*e + 15360*a^5*b^8*c^5*e - 122880*a^6*b^6*c^6*e + 491520*a^7*b^4*c^7*e - 983040*a^8*b^2*c^8*e))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^

3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k)*x*(4194304*a^11*b*c^9 - 256*a^4*b^15*c^2 + 7168*a^5*b^13*c^3 - 86016*a^6*b^11*c^4 + 573440*a^7*b^9*c^5 - 2293760*a^8*b^7*c^6 + 5505024*a^9*b^5*c^7 - 7340032*a^10*b^3*c^8))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) + (3244032*a^6*b*c^8*d*e - 983040*a^7*c^8*e*f + 4608*a^2*b^9*c^4*d*e - 87552*a^3*b^7*c^5*d*e + 681984*a^4*b^5*c^6*d*e - 2433024*a^5*b^3*c^7*d*e + 1536*a^3*b^8*c^4*e*f - 39936*a^4*b^6*c^5*e*f + 184320*a^5*b^4*c^6*e*f + 49152*a^6*b^2*c^7*e*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) - (x*(225792*a^6*c^9*d^2 + 9*b^12*c^3*d^2 - 12800*a^7*c^8*f^2 - 252*a*b^10*c^4*d^2 - 36864*a^6*b*c^8*e^2 + 3114*a^2*b^8*c^5*d^2 - 21312*a^3*b^6*c^6*d^2 + 88128*a^4*b^4*c^7*d^2 - 211968*a^5*b^2*c^8*d^2 - 2304*a^4*b^5*c^6*e^2 + 18432*a^5*b^3*c^7*e^2 + a^2*b^10*c^3*f^2 - 42*a^3*b^8*c^4*f^2 + 1760*a^4*b^6*c^5*f^2 - 13120*a^5*b^4*c^6*f^2 + 29952*a^6*b^2*c^7*f^2 + 6*a*b^11*c^3*d*f - 109056*a^6*b*c^8*d*f - 210*a^2*b^9*c^4*d*f + 2496*a^3*b^7*c^5*d*f - 18240*a^4*b^5*c^6*d*f + 72192*a^5*b^3*c^7*d*f))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5))) - (567*b^7*c^5*d^3 + 8000*a^5*c^7*f^3 - 10368*a*b^5*c^6*d^3 - 169344*a^3*b*c^8*d^3 - 193536*a^4*c^8*d*e^2 + 141120*a^4*c^8*d^2*f - 315*b^8*c^4*d^2*f + 67824*a^2*b^3*c^7*d^3 - 35*a^2*b^6*c^4*f^3 - 84*a^3*b^4*c^5*f^3 + 12720*a^4*b^2*c^6*f^3 + 6237*a*b^6*c^5*d^2*f - 210*a*b^7*c^4*d*f^2 - 116160*a^4*b*c^7*d*f^2 + 36864*a^4*b*c^7*e^2*f - 6912*a^2*b^4*c^6*d*e^2 + 62208*a^3*b^2*c^7*d*e^2 - 42372*a^2*b^4*c^6*d^2*f + 1764*a^2*b^5*c^5*d*f^2 + 96048*a^3*b^2*c^7*d^2*f + 4608*a^3*b^3*c^6*d*f^2 - 2304*a^3*b^3*c^6*e^2*f)/(512*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8 *c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)) + (x*(6912*a^4*c^8*e^3 - 27*b^7*c^5*d^2*e - 10080*a^4*c^8*d*e*f + 486*a*b^5*c^6*d^2*e + 12096*a^3*b*c^8*d^2*e + 3120*a^4*b*c^7*e*f^2 - 3672*a^2*b^3*c^7*d^2*e - 3*a^2*b^5*c^5*e*f^2 + 96*a^3*b^3*c^6*e*f^2 - 18*a*b^6*c^5*d*e*f + 450*a^2*b^4*c^6*d*e*f - 2448*a^3*b^2*c^7*d*e*f))/(32*(a^4*b^12 + 4096*a^10*c^6 - 24*a^5*b^10*c + 240*a^6*b^8*c^2 - 1280*a^7*b^6*c^3 + 3840*a^8*b^4*c^4 - 6144*a^9*b^2*c^5)))*root(56371445760*a^11*b^8*c^6*z^4 - 503316480*a^8*b^14*c^3*z^4 + 47185920*a^7*b^16*c^2*z^4 - 171798691840*a^14*b^2*c^9*z^4 + 193273528320*a^13*b^4*c^8*z^4 - 128849018880*a^12*b^6*c^7*z^4 - 16911433728*a^10*b^10*c^5*z^4 + 3523215360*a^9*b^12*c^4*z^4 - 2621440*a^6*b^18*c*z^4 + 68719476736*a^15*c^10*z^4 + 65536*a^5*b^20*z^4 - 73728*a^2*b^16*c*d*f*z^2 - 1321205760*a^9*b^2*c^8*d*f*z^2 + 732168192*a^7*b^6*c^6*d*f*z^2 - 366280704*a^6*b^8*c^5*d*f*z^2 - 330301440*a^8*b^4*c^7*d*f*z^2 + 96583680*a^5*b^10*c^4*d*f*z^2 - 15175680*a^4*b^12*c^3*d*f*z^2 + 1428480*a^3*b^14*c^2*d*f*z^2 - 440401920*a^10*b*c^8*f^2*z^2 + 1761607680*a^10*c^9*d*f*z^2 - 14080*a^3*b^15*c*f^2*z^2 + 6936330240*a^8*b^3*c^8*d^2*z^2 + 2464874496*a^6*b^7*c^6*d^2*z^2 - 3963617280*a^9*b*c^9*d^2*z^2 - 1509949440*a^9*b^2*c^8*e^2*z^2 - 5400428544*a^7*b^5*c^7*d^2*z^2 - 94464*a*b^17*c*d^2*z^2 + 754974720*a^8*b^4*c^7*e^2*z^2 - 730054656*a^5*b^9*c^5*d^2*z^2 + 477102080*a^9*b^3*c^7*f^2*z^2 - 174325760*a^8*b^5*c^6*f^2*z^2 - 188743680*a^7*b^6*c^6*e^2*z^2 + 146165760*a^4*b^11*c^4*d^2*z^2 + 11206656*a^7*b^7*c^5*f^2*z^2 + 8929280*a^6*b^9*c^4*f^2*z^2 + 23592960*a^6*b^8*c^5*e^2*z^2 - 2600960*a^5*b^11*c^3*f^2*z^2 + 291840*a^4*b^13*c^2*f^2*z^2 - 19860480*a^3*b^13*c^3*d^2*z^2 - 1179648*a^5*b^10*c^4*e^2*z^2 + 1771776*a^2*b^15*c^2*d^2*z^2 + 1536*a*b^18*d*f*z^2 + 1207959552*a^10*c^9*e^2*z^2 + 256*a^2*b^17*f^2*z^2 + 2304*b^19*d^2*z^2 + 169869312*a^7*b*c^8*d*e*f*z + 9216*a*b^13*c^2*d*e*f*z - 221773824*a^6*b^3*c^7*d*e*f*z + 117964800*a^5*b^5*c^6*d*e*f*z - 32440320*a^4*b^7*c^5*d*e*f*z + 4792320*a^3*b^9*c^4*d*e*f*z - 350208*a^2*b^11*c^3*d*e*f*z - 428544*a*b^12*c^3*d^2*e*z + 1022754816*a^6*b^2*c^8*d^2*e*z - 642318336*a^5*b^4*c^7*d^2*e*z + 223395840*a^4*b^6*c^6*d^2*e*z - 50724864*a^7*b^2*c^7*e*f^2*z + 26542080*a^6*b^4*c^6*e*f^2*z - 46725120*a^3*b^8*c^5*d^2*e*z - 7127040*a^5*b^6*c^5*e*f^2*z + 1013760*a^4*b^8*c^4*e*f^2*z - 69120*a^3*b^10*c^3*e*f^2*z + 1536*a^2*b^12*c^2*e*f^2*z + 5930496*a^2*b^10*c^4*d^2*e*z - 693633024*a^7*c^9*d^2*e*z + 39321600*a^8*c^8*e*f^2*z + 13824*b^14*c^2*d^2*e*z + 13824*a*b^8*c^4*d*e^2*f - 7741440*a^4*b^2*c^7*d*e^2*f + 2903040*a^3*b^4*c^6*d*e^2*f - 387072*a^2*b^6*c^5*d*e^2*f + 37310976*a^3*b^3*c^7*d^3*f + 3870720*a^5*b*c^7*e^2*f^2 + 34836480*a^4*b*c^8*d^2*e^2 - 8068032*a^2*b^5*c^6*d^3*f - 5623296*a^4*b^3*c^6*d*f^3 + 1737792*a^3*b^5*c^5*d*f^3 - 260190*a*b^8*c^4*d^2*f^2 - 211680*a^2*b^7*c^4*d*f^3 - 435456*a*b^7*c^5*d^2*e^2 - 75188736*a^4*b*c^8*d^3*f - 15482880*a^5*c^8*d*e^2*f - 4262400*a^5*b*c^7*d*f^3 + 852768*a*b^7*c^5*d^3*f + 7350*a*b^9*c^3*d*f^3 + 35525376*a^4*b^2*c^7*d^2*f^2 + 645120*a^4*b^3*c^6*e^2*f^2 - 80640*a^3*b^5*c^5*e^2*f^2 + 2304*a^2*b^7*c^4*e^2*f^2 - 15269184*a^3*b^4*c^6*d^2*f^2 + 2870784*a^2*b^6*c^5*d^2*f^2 - 17418240*a^3*b^3*c^7*d^2*e^2 + 3919104*a^2*b^5*c^6*d^2*e^2 + 11025*b^10*c^3*d^2*f^2 + 5644800*a^5*c^8*d^2*f^2 + 20736*b^9*c^4*d^2*e^2 + 492800*a^5*b^2*c^6*f^4 + 351456*a^4*b^4*c^5*f^4 - 43120*a^3*b^

```
6*c^4*f^4 + 1225*a^2*b^8*c^3*f^4 - 27433728*a^3*b^2*c^8*d^4 + 6446304*a^2*b
^4*c^7*d^4 - 39690*b^9*c^4*d^3*f - 734832*a*b^6*c^6*d^4 + 49787136*a^4*c^9*
d^4 + 160000*a^6*c^7*f^4 + 5308416*a^5*c^8*e^4 + 35721*b^8*c^5*d^4, z, k),
k, 1, 4)
```

3.67 $\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 26, antiderivative size = 4

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4} \, dx = \log(2+x)$$

```
[Out] ln(2+x)
```

Rubi [A] (verified)

Time = 0.01 (sec) , antiderivative size = 4, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.077$, Rules used = {1600, 31}

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4} \, dx = \log(x+2)$$

```
[In] Int[(2 - x - 2*x^2 + x^3)/(4 - 5*x^2 + x^4),x]
```

```
[Out] Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{1}{2+x}\,dx \\
&= \log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.00 (sec) , antiderivative size = 4, normalized size of antiderivative = 1.00

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \log(2+x)$$

```
[In] Integrate[(2 - x - 2*x^2 + x^3)/(4 - 5*x^2 + x^4),x]
[Out] Log[2 + x]
```

Maple [A] (verified)

Time = 0.02 (sec) , antiderivative size = 5, normalized size of antiderivative = 1.25

method	result	size
default	$\ln(x+2)$	5
norman	$\ln(x+2)$	5
risch	$\ln(x+2)$	5
parallelrisch	$\ln(x+2)$	5

```
[In] int((x^3-2*x^2-x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 4, normalized size of antiderivative = 1.00

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] log(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 3, normalized size of antiderivative = 0.75

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \log(x+2)$$

```
[In] integrate((x**3-2*x**2-x+2)/(x**4-5*x**2+4),x)
[Out] log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 4, normalized size of antiderivative = 1.00

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] log(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 5, normalized size of antiderivative = 1.25

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \log(|x+2|)$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 0.01 (sec) , antiderivative size = 4, normalized size of antiderivative = 1.00

$$\int \frac{2-x-2x^2+x^3}{4-5x^2+x^4}\,dx = \ln(x+2)$$

```
[In] int(-(x + 2*x^2 - x^3 - 2)/(x^4 - 5*x^2 + 4),x)
[Out] log(x + 2)
```

3.68 $\int \frac{(d+ex)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 14

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = ex+(d-2e)\log(2+x)$$

```
[Out] e*x+(d-2*e)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.02 (sec) , antiderivative size = 14, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.065, Rules used = {1600, 45}

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = (d-2e)\log(x+2)+ex$$

```
[In] Int[((d + e*x)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4),x]
```

```
[Out] e*x + (d - 2*e)*Log[2 + x]
```

Rule 45

```
Int[((a_.) + (b_.)*(x_))^(m_.)*((c_.) + (d_.)*(x_))^(n_.), x_Symbol] :> Int
[ExpandIntegrand[(a + b*x)^m*(c + d*x)^n, x], x] /; FreeQ[{a, b, c, d, n},
x] && NeQ[b*c - a*d, 0] && IGtQ[m, 0] && ( !IntegerQ[n] || (EqQ[c, 0] && Le
Q[7*m + 4*n + 4, 0]) || LtQ[9*m + 5*(n + 1), 0] || GtQ[m + n + 2, 0])
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
```

```
EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex}{2+x}\,dx\\ &= \int \left(e+\frac{d-2e}{2+x}\right)dx\\ &= ex+(d-2e)\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 16, normalized size of antiderivative = 1.14

$$\int \frac{(d+ex)\left(2-x-2x^2+x^3\right)}{4-5x^2+x^4}\,dx = e(2+x)+(d-2e)\log(2+x)$$

```
[In] Integrate[((d + e*x)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4),x]
[Out] e*(2 + x) + (d - 2*e)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.03 (sec) , antiderivative size = 15, normalized size of antiderivative = 1.07

method	result	size
default	$ex+(d-2e)\ln(x+2)$	15
norman	$ex+(d-2e)\ln(x+2)$	15
risch	$ex+\ln(x+2)\,d-2\ln(x+2)\,e$	18
parallelrisch	$ex+\ln(x+2)\,d-2\ln(x+2)\,e$	18

```
[In] int((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] e*x+(d-2*e)*ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 14, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = ex+(d-2\,e)\log{(x+2)}$$

```
[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] e*x + (d - 2*e)*log(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 12, normalized size of antiderivative = 0.86

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = ex+(d-2e)\log{(x+2)}$$

```
[In] integrate((e*x+d)*(x**3-2*x**2-x+2)/(x**4-5*x**2+4),x)
[Out] e*x + (d - 2*e)*log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 14, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = ex+(d-2\,e)\log{(x+2)}$$

```
[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] e*x + (d - 2*e)*log(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 15, normalized size of antiderivative = 1.07

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = ex+(d-2\,e)\log{(|x+2|)}$$

```
[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] e*x + (d - 2*e)*log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 7.86 (sec) , antiderivative size = 14, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \ln{(x+2)}\,\,(d-2\,e)+e\,x$$

```
[In] int(-((d + e*x)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4),x)
[Out] log(x + 2)*(d - 2*e) + e*x
```

3.69 $\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 36, antiderivative size = 31

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4} dx = (e-4f)x + \frac{1}{2}f(2+x)^2 + (d-2e+4f)\log(2+x)$$

```
[Out] (e-4*f)*x+1/2*f*(2+x)^2+(d-2*e+4*f)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 31, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.056, Rules used = {1600, 712}

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4} dx = \log(x+2)(d-2e+4f) + x(e-4f) + \frac{1}{2}f(x+2)^2$$

```
[In] Int[((d + e*x + f*x^2)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4),x]
[Out] (e - 4*f)*x + (f*(2 + x)^2)/2 + (d - 2*e + 4*f)*Log[2 + x]
```

Rule 712

```
Int[((d_.) + (e_.)*(x_))^(m_)*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_
Symbol] :> Int[ExpandIntegrand[(d + e*x)^m*(a + b*x + c*x^2)^p, x], x] /; F
reeQ[{a, b, c, d, e, m}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a
*e^2, 0] && NeQ[2*c*d - b*e, 0] && IntegerQ[p] && (GtQ[p, 0] || (EqQ[a, 0]
&& IntegerQ[m]))
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex+fx^2}{2+x}\,dx\\ &= \int \left(e-4f+\frac{d-2e+4f}{2+x}+f(2+x)\right)dx\\ &= (e-4f)x+\frac{1}{2}f(2+x)^2+(d-2e+4f)\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 30, normalized size of antiderivative = 0.97

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}(2e+f(-6+x))(2+x)+(d-2e+4f)\log(2+x)$$

```
[In] Integrate[((d + e*x + f*x^2)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4),x]
[Out] ((2*e + f*(-6 + x))*(2 + x))/2 + (d - 2*e + 4*f)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.03 (sec) , antiderivative size = 28, normalized size of antiderivative = 0.90

method	result	size
default	$\frac{f\,x^2}{2}+ex-2fx+(d-2e+4f)\ln(x+2)$	28
norman	$(e-2f)\,x+\frac{f\,x^2}{2}+(d-2e+4f)\ln(x+2)$	28
risch	$\frac{f\,x^2}{2}+ex-2fx+\ln(x+2)\,d-2\ln(x+2)\,e+4\ln(x+2)\,f$	35
parallelrisch	$\frac{f\,x^2}{2}+ex-2fx+\ln(x+2)\,d-2\ln(x+2)\,e+4\ln(x+2)\,f$	35

```
[In] int((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] 1/2*f*x^2+e*x-2*f*x+(d-2*e+4*f)*ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 27, normalized size of antiderivative = 0.87

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}\,fx^2+(e-2\,f)x+(d-2\,e+4\,f)\log(x+2)$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="fricas"
)
```

```
[Out] 1/2*f*x^2 + (e - 2*f)*x + (d - 2*e + 4*f)*log(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.08 (sec) , antiderivative size = 26, normalized size of antiderivative = 0.84

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \frac{fx^2}{2}+x(e-2f)+(d-2e+4f)\log(x+2)$$

```
[In] integrate((f*x**2+e*x+d)*(x**3-2*x**2-x+2)/(x**4-5*x**2+4),x)
```

```
[Out] f*x**2/2 + x*(e - 2*f) + (d - 2*e + 4*f)*log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 27, normalized size of antiderivative = 0.87

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}\,fx^2+(e-2\,f)x+(d-2\,e+4\,f)\log(x+2)$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="maxima"
)
```

```
[Out] 1/2*f*x^2 + (e - 2*f)*x + (d - 2*e + 4*f)*log(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 28, normalized size of antiderivative = 0.90

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}fx^2+ex-2fx+(d-2e+4f)\log(|x+2|)$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] 1/2*f*x^2 + e*x - 2*f*x + (d - 2*e + 4*f)*log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 27, normalized size of antiderivative = 0.87

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{4-5x^2+x^4}\,dx = x(e-2f)+\frac{fx^2}{2}+\ln(x+2)\,(d-2e+4f)$$

```
[In] int(-((d + e*x + f*x^2)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4),x)
[Out] x*(e - 2*f) + (f*x^2)/2 + log(x + 2)*(d - 2*e + 4*f)
```

3.70 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 41, antiderivative size = 51

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = (e-4f+12g)x+\frac{1}{2}(f-6g)(2+x)^2+\frac{1}{3}g(2+x)^3+(d-2e+4f-8g)\log(2+x)$$

```
[Out] (e-4*f+12*g)*x+1/2*(f-6*g)*(2+x)^2+1/3*g*(2+x)^3+(d-2*e+4*f-8*g)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 51, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.049$, Rules used = {1600, 1864}

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \log(x+2)(d-2e+4f-8g)+x(e-4f+12g)+\frac{1}{2}(x+2)^2(f-6g)+\frac{1}{3}g(x+2)^3$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4),x]
```

```
[Out] (e - 4*f + 12*g)*x + ((f - 6*g)*(2 + x)^2)/2 + (g*(2 + x)^3)/3 + (d - 2*e +
 4*f - 8*g)*Log[2 + x]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned} \text{integral} &= \int \frac{d+ex+fx^2+gx^3}{2+x}\,dx \\ &= \int \left(e-4f+12g+\frac{d-2e+4f-8g}{2+x}+(f-6g)(2+x)+g(2+x)^2\right)dx \\ &= (e-4f+12g)x+\frac{1}{2}(f-6g)(2+x)^2+\frac{1}{3}g(2+x)^3+(d-2e+4f-8g)\log(2+x) \end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.88

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \frac{1}{6}(2+x)\left(6e+3f(-6+x)+2g\left(22-5x+x^2\right)\right)+(d-2e+4f-8g)\log(2+x)$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^
4),x]
[Out] ((2 + x)*(6*e + 3*f*(-6 + x) + 2*g*(22 - 5*x + x^2)))/6 + (d - 2*e + 4*f -
8*g)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.88

method	result
norman	$\left(\frac{f}{2}-g\right)x^2+(e-2f+4g)\,x+\frac{g\,x^3}{3}+(d-2e+4f-8g)\ln(x+2)$
default	$\frac{g\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+ex-2fx+4gx+(d-2e+4f-8g)\ln(x+2)$
risch	$\frac{g\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+ex-2fx+4gx+\ln(x+2)\,d-2\ln(x+2)\,e+4\ln(x+2)\,f-8\ln(x$
parallelrisch	$\frac{g\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+ex-2fx+4gx+\ln(x+2)\,d-2\ln(x+2)\,e+4\ln(x+2)\,f-8\ln(x$

```
[In] int((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBO
SE)
```

```
[Out] (1/2*f-g)*x^2+(e-2*f+4*g)*x+1/3*g*x^3+(d-2*e+4*f-8*g)*ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 43, normalized size of antiderivative = 0.84

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{3}gx^3+\frac{1}{2}(f-2g)x^2+(e-2f+4g)x+(d-2e+4f-8g)\log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="f
ricas")
[Out] 1/3*g*x^3 + 1/2*(f - 2*g)*x^2 + (e - 2*f + 4*g)*x + (d - 2*e + 4*f - 8*g)*l
og(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.08 (sec) , antiderivative size = 41, normalized size of antiderivative = 0.80

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$$
$$=\frac{gx^3}{3}+x^2\left(\frac{f}{2}-g\right)+x(e-2f+4g)+(d-2e+4f-8g)\log(x+2)$$

```
[In] integrate((x**3-2*x**2-x+2)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] g*x**3/3 + x**2*(f/2 - g) + x*(e - 2*f + 4*g) + (d - 2*e + 4*f - 8*g)*log(x
 + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 43, normalized size of antiderivative = 0.84

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{3}gx^3+\frac{1}{2}(f-2g)x^2+(e-2f+4g)x+(d-2e+4f-8g)\log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="m
axima")
[Out] 1/3*g*x^3 + 1/2*(f - 2*g)*x^2 + (e - 2*f + 4*g)*x + (d - 2*e + 4*f - 8*g)*l
og(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 47, normalized size of antiderivative = 0.92

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{3}\,gx^3+\frac{1}{2}\,fx^2-gx^2+ex-2\,fx+4\,gx+(d-2\,e+4\,f-8\,g)\log(|x+2|)$$

```
[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="g
iac")
```

```
[Out] 1/3*g*x^3 + 1/2*f*x^2 - g*x^2 + e*x - 2*f*x + 4*g*x + (d - 2*e + 4*f - 8*g)
*log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 44, normalized size of antiderivative = 0.86

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$$
$$=x^2\left(\frac{f}{2}-g\right)+x\,(e-2\,f+4\,g)+\frac{g\,x^3}{3}+\ln(x+2)\ (d-2\,e+4\,f-8\,g)$$

```
[In] int(-((d + e*x + f*x^2 + g*x^3)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4),x)
```

```
[Out] x^2*(f/2 - g) + x*(e - 2*f + 4*g) + (g*x^3)/3 + log(x + 2)*(d - 2*e + 4*f -
 8*g)
```

3.71 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 46, antiderivative size = 68

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx &= (e-2f+4g-8h)x+\frac{1}{2}(f-2g+4h)x^2+\frac{1}{3}(g-2h)x^3\\&\quad+\frac{hx^4}{4}+(d-2e+4f-8g+16h)\log(2+x)\end{aligned}$$

```
[Out] (e-2*f+4*g-8*h)*x+1/2*(f-2*g+4*h)*x^2+1/3*(g-2*h)*x^3+1/4*h*x^4+(d-2*e+4*f-
8*g+16*h)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.08 (sec) , antiderivative size = 68, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.043, Rules used = {1600, 1864}

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx &= \log(x+2)(d-2e+4f-8g+16h)+x(e-2f+4g-8h)\\&\quad+\frac{1}{2}x^2(f-2g+4h)+\frac{1}{3}x^3(g-2h)+\frac{hx^4}{4}\end{aligned}$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 +
x^4),x]
```

```
[Out] (e - 2*f + 4*g - 8*h)*x + ((f - 2*g + 4*h)*x^2)/2 + ((g - 2*h)*x^3)/3 + (h*
x^4)/4 + (d - 2*e + 4*f - 8*g + 16*h)*Log[2 + x]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4}{2+x}\,dx\\ &= \int \left(e\left(1-\frac{2(f-2g+4h)}{e}\right)+(f-2g+4h)x+(g-2h)x^2+hx^3 + \frac{d-2e+4f-8g+16h}{2+x}\right)dx\\ &= (e-2f+4g-8h)x+\frac{1}{2}(f-2g+4h)x^2+\frac{1}{3}(g-2h)x^3+\frac{hx^4}{4}+(d-2e+4f-8g+16h)\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 68, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx\\ &= (e-2f+4g-8h)x+\frac{1}{2}(f-2g+4h)x^2+\frac{1}{3}(g-2h)x^3\\ &\quad+\frac{hx^4}{4}+(d-2e+4f-8g+16h)\log(2+x)\end{aligned}$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*
x^2 + x^4),x]

[Out] (e - 2*f + 4*g - 8*h)*x + ((f - 2*g + 4*h)*x^2)/2 + ((g - 2*h)*x^3)/3 + (h*
x^4)/4 + (d - 2*e + 4*f - 8*g + 16*h)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.96

method	result
norman	$\left(\frac{g}{3}-\frac{2h}{3}\right)x^{3}+\left(\frac{f}{2}-g+2h\right)x^{2}+\left(e-2f+4g-8h\right)x+\frac{h\,x^{4}}{4}+\left(d-2e+4f-8g+16h\right)\ln($
default	$\frac{h\,x^{4}}{4}+\frac{g\,x^{3}}{3}-\frac{2h\,x^{3}}{3}+\frac{f\,x^{2}}{2}-g\,x^{2}+2h\,x^{2}+ex-2fx+4gx-8hx+\left(d-2e+4f-8g+16h\right)$
risch	$\frac{h\,x^{4}}{4}+\frac{g\,x^{3}}{3}-\frac{2h\,x^{3}}{3}+\frac{f\,x^{2}}{2}-g\,x^{2}+2h\,x^{2}+ex-2fx+4gx-8hx+\ln\left(x+2\right)d-2\ln\left(x+2\right)$
parallelrisch	$\frac{h\,x^{4}}{4}+\frac{g\,x^{3}}{3}-\frac{2h\,x^{3}}{3}+\frac{f\,x^{2}}{2}-g\,x^{2}+2h\,x^{2}+ex-2fx+4gx-8hx+\ln\left(x+2\right)d-2\ln\left(x+2\right)$

```
[In] int((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETUR
NVERBOSE)
```

```
[Out] (1/3*g-2/3*h)*x^3+(1/2*f-g+2*h)*x^2+(e-2*f+4*g-8*h)*x+1/4*h*x^4+(d-2*e+4*f-
8*g+16*h)*ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.91

$$\int \frac{(2-x-2x^{2}+x^{3})\left(d+ex+fx^{2}+gx^{3}+hx^{4}\right)}{4-5x^{2}+x^{4}}\,dx = \frac{1}{4}\,hx^{4}+\frac{1}{3}\,(g-2\,h)x^{3}+\frac{1}{2}\,(f-2\,g+4\,h)x^{2} + (e-2\,f+4\,g-8\,h)x+(d-2\,e+4\,f-8\,g+16\,h)\log\left(x+2\right)$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algori
thm="fricas")
```

```
[Out] 1/4*h*x^4 + 1/3*(g - 2*h)*x^3 + 1/2*(f - 2*g + 4*h)*x^2 + (e - 2*f + 4*g -
8*h)*x + (d - 2*e + 4*f - 8*g + 16*h)*log(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.10 (sec) , antiderivative size = 63, normalized size of antiderivative = 0.93

$$\int \frac{(2-x-2x^{2}+x^{3})\left(d+ex+fx^{2}+gx^{3}+hx^{4}\right)}{4-5x^{2}+x^{4}}\,dx = \frac{hx^{4}}{4}+x^{3}\left(\frac{g}{3}-\frac{2h}{3}\right)+x^{2}\left(\frac{f}{2}-g+2h\right) + x(e-2f+4g-8h)+(d-2e+4f-8g+16h)\log\left(x+2\right)$$

```
[In] integrate((x**3-2*x**2-x+2)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] h*x**4/4 + x**3*(g/3 - 2*h/3) + x**2*(f/2 - g + 2*h) + x*(e - 2*f + 4*g - 8
*h) + (d - 2*e + 4*f - 8*g + 16*h)*log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.91

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx = \frac{1}{4}hx^4 + \frac{1}{3}(g-2h)x^3 + \frac{1}{2}(f-2g+4h)x^2 + (e-2f+4g-8h)x + (d-2e+4f-8g+16h)\log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algori
thm="maxima")
```

```
[Out] 1/4*h*x^4 + 1/3*(g - 2*h)*x^3 + 1/2*(f - 2*g + 4*h)*x^2 + (e - 2*f + 4*g -
8*h)*x + (d - 2*e + 4*f - 8*g + 16*h)*log(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 72, normalized size of antiderivative = 1.06

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx = \frac{1}{4}hx^4 + \frac{1}{3}gx^3 - \frac{2}{3}hx^3 + \frac{1}{2}fx^2 - gx^2 + 2hx^2 + ex - 2fx + 4gx - 8hx + (d-2e+4f-8g+16h)\log(|x+2|)$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algori
thm="giac")
```

```
[Out] 1/4*h*x^4 + 1/3*g*x^3 - 2/3*h*x^3 + 1/2*f*x^2 - g*x^2 + 2*h*x^2 + e*x - 2*f
*x + 4*g*x - 8*h*x + (d - 2*e + 4*f - 8*g + 16*h)*log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 64, normalized size of antiderivative = 0.94

$$\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$$
$$= x^3\left(\frac{g}{3}-\frac{2\,h}{3}\right)+\ln\left(x+2\right)\,\left(d-2\,e+4\,f-8\,g+16\,h\right)$$
$$+\frac{h\,x^4}{4}+x^2\left(\frac{f}{2}-g+2\,h\right)+x\,\left(e-2\,f+4\,g-8\,h\right)$$

```
[In] int(-((x + 2*x^2 - x^3 - 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2
 + 4),x)
```

```
[Out] x^3*(g/3 - (2*h)/3) + log(x + 2)*(d - 2*e + 4*f - 8*g + 16*h) + (h*x^4)/4 +
 x^2*(f/2 - g + 2*h) + x*(e - 2*f + 4*g - 8*h)
```

3.72 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 51, antiderivative size = 92

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx \\ = (e-2f+4g-8h+16i)x+\frac{1}{2}(f-2g+4h-8i)x^2+\frac{1}{3}(g-2h+4i)x^3 \\ +\frac{1}{4}(h-2i)x^4+\frac{ix^5}{5}+(d-2e+4f-8g+16h-32i)\log(2+x)$$

```
[Out] (e-2*f+4*g-8*h+16*i)*x+1/2*(f-2*g+4*h-8*i)*x^2+1/3*(g-2*h+4*i)*x^3+1/4*(h-2
*i)*x^4+1/5*i*x^5+(d-2*e+4*f-8*g+16*h-32*i)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.11 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.039, Rules used = {1600, 1864}

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx \\ = \log(x+2)(d-2e+4f-8g+16h-32i)+x(e-2f+4g-8h+16i) \\ +\frac{1}{2}x^2(f-2g+4h-8i)+\frac{1}{3}x^3(g-2h+4i)+\frac{1}{4}x^4(h-2i)+\frac{ix^5}{5}$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 -
5*x^2 + x^4),x]
```

```
[Out] (e - 2*f + 4*g - 8*h + 16*i)*x + ((f - 2*g + 4*h - 8*i)*x^2)/2 + ((g - 2*h
+ 4*i)*x^3)/3 + ((h - 2*i)*x^4)/4 + (i*x^5)/5 + (d - 2*e + 4*f - 8*g + 16*h
 - 32*i)*Log[2 + x]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1864

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.))^(p_.), x_Symbol] :> Int[ExpandIntegrand
[Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n}, x] && PolyQ[Pq, x] && (IGtQ[p
, 0] || EqQ[n, 1])
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{2+x}\,dx\\ &= \int \left(e\left(1-\frac{2(f-2g+4h-8i)}{e}\right)+(f-2g+4h-8i)x+(g-2h+4i)x^2\right.\\ &\qquad \left.+(h-2i)x^3+ix^4+\frac{d-2e+4f-8g+16h-32i}{2+x}\right)dx\\ &= (e-2f+4g-8h+16i)x+\frac{1}{2}(f-2g+4h-8i)x^2+\frac{1}{3}(g-2h+4i)x^3\\ &\quad +\frac{1}{4}(h-2i)x^4+\frac{ix^5}{5}+(d-2e+4f-8g+16h-32i)\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.00

$$\begin{aligned}&\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx\\ &= (e-2f+4g-8h+16i)x+\frac{1}{2}(f-2g+4h-8i)x^2+\frac{1}{3}(g-2h+4i)x^3\\ &\quad +\frac{1}{4}(h-2i)x^4+\frac{ix^5}{5}+(d-2e+4f-8g+16h-32i)\log(2+x)\end{aligned}$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))
/(4 - 5*x^2 + x^4),x]

[Out] (e - 2*f + 4*g - 8*h + 16*i)*x + ((f - 2*g + 4*h - 8*i)*x^2)/2 + ((g - 2*h
+ 4*i)*x^3)/3 + ((h - 2*i)*x^4)/4 + (i*x^5)/5 + (d - 2*e + 4*f - 8*g + 16*h
 - 32*i)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 88, normalized size of antiderivative = 0.96

method	result
norman	$\left(\frac{h}{4}-\frac{i}{2}\right) x^4+\left(\frac{g}{3}-\frac{2h}{3}+\frac{4i}{3}\right) x^3+\left(\frac{f}{2}-g+2h-4i\right) x^2+\left(e-2f+4g-8h+16i\right) x+\frac{i\,x^5}{5}+$
default	$\frac{i\,x^5}{5}+\frac{h\,x^4}{4}-\frac{i\,x^4}{2}+\frac{g\,x^3}{3}-\frac{2h\,x^3}{3}+\frac{4i\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+2h\,x^2-4i\,x^2+ex-2fx+4gx-8hx$
risch	$\frac{i\,x^5}{5}+\frac{h\,x^4}{4}-\frac{i\,x^4}{2}+\frac{g\,x^3}{3}-\frac{2h\,x^3}{3}+\frac{4i\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+2h\,x^2-4i\,x^2+ex-2fx+4gx-8hx$
parallelrisch	$\frac{i\,x^5}{5}+\frac{h\,x^4}{4}-\frac{i\,x^4}{2}+\frac{g\,x^3}{3}-\frac{2h\,x^3}{3}+\frac{4i\,x^3}{3}+\frac{f\,x^2}{2}-g\,x^2+2h\,x^2-4i\,x^2+ex-2fx+4gx-8hx$

```
[In] int((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=
_RETURNVERBOSE)
```

```
[Out] (1/4*h-1/2*i)*x^4+(1/3*g-2/3*h+4/3*i)*x^3+(1/2*f-g+2*h-4*i)*x^2+(e-2*f+4*g-
8*h+16*i)*x+1/5*i*x^5+(d-2*e+4*f-8*g+16*h-32*i)*ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.91

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx &= \frac{1}{5}\,ix^5+\frac{1}{4}\,(h-2\,i)x^4+\frac{1}{3}\,(g-2\,h+4\,i)x^3+\frac{1}{2}\,(f-2\,g+4\,h-8\,i)x^2 \\ &\quad+(e-2\,f+4\,g-8\,h+16\,i)x+(d-2\,e+4\,f-8\,g+16\,h-32\,i)\log{(x+2)}\end{aligned}$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,
algorithm="fricas")
```

```
[Out] 1/5*i*x^5 + 1/4*(h - 2*i)*x^4 + 1/3*(g - 2*h + 4*i)*x^3 + 1/2*(f - 2*g + 4*
h - 8*i)*x^2 + (e - 2*f + 4*g - 8*h + 16*i)*x + (d - 2*e + 4*f - 8*g + 16*h
 - 32*i)*log(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.13 (sec) , antiderivative size = 88, normalized size of antiderivative = 0.96

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$=\frac{ix^5}{5}+x^4\left(\frac{h}{4}-\frac{i}{2}\right)+x^3\left(\frac{g}{3}-\frac{2h}{3}+\frac{4i}{3}\right)+x^2\left(\frac{f}{2}-g+2h-4i\right)$$
$$+x(e-2f+4g-8h+16i)+(d-2e+4f-8g+16h-32i)\log(x+2)$$

```
[In] integrate((x**3-2*x**2-x+2)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**
2+4),x)
```

```
[Out] i*x**5/5 + x**4*(h/4 - i/2) + x**3*(g/3 - 2*h/3 + 4*i/3) + x**2*(f/2 - g +
2*h - 4*i) + x*(e - 2*f + 4*g - 8*h + 16*i) + (d - 2*e + 4*f - 8*g + 16*h -
 32*i)*log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.91

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{5}\,ix^5+\frac{1}{4}\,(h-2\,i)x^4+\frac{1}{3}\,(g-2\,h+4\,i)x^3+\frac{1}{2}\,(f-2\,g+4\,h-8\,i)x^2$$
$$+(e-2\,f+4\,g-8\,h+16\,i)x+(d-2\,e+4\,f-8\,g+16\,h-32\,i)\log(x+2)$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,
algorithm="maxima")
```

```
[Out] 1/5*i*x^5 + 1/4*(h - 2*i)*x^4 + 1/3*(g - 2*h + 4*i)*x^3 + 1/2*(f - 2*g + 4*
h - 8*i)*x^2 + (e - 2*f + 4*g - 8*h + 16*i)*x + (d - 2*e + 4*f - 8*g + 16*h
 - 32*i)*log(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 103, normalized size of antiderivative = 1.12

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{5}\,ix^5+\frac{1}{4}\,hx^4-\frac{1}{2}\,ix^4+\frac{1}{3}\,gx^3-\frac{2}{3}\,hx^3+\frac{4}{3}\,ix^3+\frac{1}{2}\,fx^2-gx^2+2\,hx^2-4\,ix^2$$
$$+ex-2\,fx+4\,gx-8\,hx+16\,ix+(d-2\,e+4\,f-8\,g+16\,h-32\,i)\log(|x+2|)$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,
algorithm="giac")
```

```
[Out] 1/5*i*x^5 + 1/4*h*x^4 - 1/2*i*x^4 + 1/3*g*x^3 - 2/3*h*x^3 + 4/3*i*x^3 + 1/2
*f*x^2 - g*x^2 + 2*h*x^2 - 4*i*x^2 + e*x - 2*f*x + 4*g*x - 8*h*x + 16*i*x +
 (d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(abs(x + 2))
```

Mupad [B] (verification not implemented)

Time = 0.02 (sec) , antiderivative size = 87, normalized size of antiderivative = 0.95

$$\begin{aligned}
&\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx \\
&= x^4\left(\frac{h}{4}-\frac{i}{2}\right)+\ln(x+2)\ (d-2\,e+4\,f-8\,g+16\,h-32\,i)+\frac{i\,x^5}{5} \\
&\quad +x^2\left(\frac{f}{2}-g+2\,h-4\,i\right)+x\,(e-2\,f+4\,g-8\,h+16\,i)+x^3\left(\frac{g}{3}-\frac{2\,h}{3}+\frac{4\,i}{3}\right)
\end{aligned}$$

```
[In] int(-((x + 2*x^2 - x^3 - 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4
 - 5*x^2 + 4),x)
```

```
[Out] x^4*(h/4 - i/2) + log(x + 2)*(d - 2*e + 4*f - 8*g + 16*h - 32*i) + (i*x^5)/
5 + x^2*(f/2 - g + 2*h - 4*i) + x*(e - 2*f + 4*g - 8*h + 16*i) + x^3*(g/3 -
 (2*h)/3 + (4*i)/3)
```

3.73 $\int \frac{2-3x+x^2}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 21, antiderivative size = 11

$$\int \frac{2-3x+x^2}{4-5x^2+x^4} \, dx = \log(1+x) - \log(2+x)$$

```
[Out] ln(1+x)-ln(2+x)
```

Rubi [A] (verified)

Time = 0.01 (sec) , antiderivative size = 11, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}} = 0.143$, Rules used = {1600, 630, 31}

$$\int \frac{2-3x+x^2}{4-5x^2+x^4} \, dx = \log(x+1) - \log(x+2)$$

```
[In] Int[(2 - 3*x + x^2)/(4 - 5*x^2 + x^4),x]
```

```
[Out] Log[1 + x] - Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 630

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = Rt[b^2
 - 4*a*c, 2]}, Dist[c/q, Int[1/Simp[b/2 - q/2 + c*x, x], x], x] - Dist[c/q,
 Int[1/Simp[b/2 + q/2 + c*x, x], x], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[b^2 - 4*a*c] && PerfectSquareQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{1}{2+3x+x^2}\,dx\\ &= \int \frac{1}{1+x}\,dx - \int \frac{1}{2+x}\,dx\\ &= \log(1+x)-\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 11, normalized size of antiderivative = 1.00

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = \log(1+x)-\log(2+x)$$

```
[In] Integrate[(2 - 3*x + x^2)/(4 - 5*x^2 + x^4),x]
[Out] Log[1 + x] - Log[2 + x]
```

Maple [A] (verified)

Time = 0.03 (sec) , antiderivative size = 12, normalized size of antiderivative = 1.09

method	result	size
default	$\ln(x+1)-\ln(x+2)$	12
norman	$\ln(x+1)-\ln(x+2)$	12
risch	$\ln(x+1)-\ln(x+2)$	12
parallelrisch	$\ln(x+1)-\ln(x+2)$	12

```
[In] int((x^2-3*x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] ln(x+1)-ln(x+2)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 11, normalized size of antiderivative = 1.00

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = -\log(x+2) + \log(x+1)$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] -log(x + 2) + log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 8, normalized size of antiderivative = 0.73

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = \log(x+1) - \log(x+2)$$

```
[In] integrate((x**2-3*x+2)/(x**4-5*x**2+4),x)
[Out] log(x + 1) - log(x + 2)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 11, normalized size of antiderivative = 1.00

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = -\log(x+2) + \log(x+1)$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] -log(x + 2) + log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 13, normalized size of antiderivative = 1.18

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = -\log(|x+2|) + \log(|x+1|)$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] -log(abs(x + 2)) + log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 8, normalized size of antiderivative = 0.73

$$\int \frac{2-3x+x^2}{4-5x^2+x^4}\,dx = -2\,\mathrm{atanh}(2\,x+3)$$

```
[In] int((x^2 - 3*x + 2)/(x^4 - 5*x^2 + 4),x)
[Out] -2*atanh(2*x + 3)
```

3.74 $\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 26, antiderivative size = 22

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4} dx = (d-e)\log(1+x) - (d-2e)\log(2+x)$$

```
[Out] (d-e)*ln(1+x)-(d-2*e)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.01 (sec) , antiderivative size = 22, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.115, Rules used = {1600, 646, 31}

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4} dx = (d-e)\log(x+1) - (d-2e)\log(x+2)$$

```
[In] Int[((d + e*x)*(2 - 3*x + x^2))/(4 - 5*x^2 + x^4),x]
```

```
[Out] (d - e)*Log[1 + x] - (d - 2*e)*Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
```

```
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex}{2+3x+x^2}\,dx \\ &= -\left((d-2e)\int \frac{1}{2+x}\,dx\right)+(d-e)\int \frac{1}{1+x}\,dx \\ &= (d-e)\log(1+x)-(d-2e)\log(2+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 23, normalized size of antiderivative = 1.05

$$\int \frac{(d+ex)\left(2-3x+x^2\right)}{4-5x^2+x^4}\,dx = (d-e)\log(1+x)+(-d+2e)\log(2+x)$$

```
[In] Integrate[((d + e*x)*(2 - 3*x + x^2))/(4 - 5*x^2 + x^4),x]
[Out] (d - e)*Log[1 + x] + (-d + 2*e)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 24, normalized size of antiderivative = 1.09

method	result	size
default	$(-d+2e)\ln(x+2)+(d-e)\ln(x+1)$	24
norman	$(-d+2e)\ln(x+2)+(d-e)\ln(x+1)$	24
parallelrisch	$\ln(x+1)\,d-\ln(x+1)\,e-\ln(x+2)\,d+2\ln(x+2)\,e$	29
risch	$-\ln(x+2)\,d+2\ln(x+2)\,e+\ln(-x-1)\,d-\ln(-x-1)\,e$	33

```
[In] int((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] (-d+2*e)*ln(x+2)+(d-e)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 22, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4}\,dx = -(d-2\,e)\log(x+2)+(d-e)\log(x+1)$$

```
[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] -(d - 2*e)*log(x + 2) + (d - e)*log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 0.17 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.32

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4}\,dx = (-d+2e)\log\left(x+\frac{4d-6e}{2d-3e}\right)+(d-e)\log(x+1)$$

```
[In] integrate((e*x+d)*(x**2-3*x+2)/(x**4-5*x**2+4),x)
[Out] (-d + 2*e)*log(x + (4*d - 6*e)/(2*d - 3*e)) + (d - e)*log(x + 1)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 22, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4}\,dx = -(d-2\,e)\log(x+2)+(d-e)\log(x+1)$$

```
[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] -(d - 2*e)*log(x + 2) + (d - e)*log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 24, normalized size of antiderivative = 1.09

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4}\,dx = -(d-2\,e)\log(|x+2|)+(d-e)\log(|x+1|)$$

```
[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] -(d - 2*e)*log(abs(x + 2)) + (d - e)*log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 7.89 (sec) , antiderivative size = 22, normalized size of antiderivative = 1.00

$$\int \frac{(d+ex)(2-3x+x^2)}{4-5x^2+x^4}\,dx = \ln(x+1)\ (d-e) - \ln(x+2)\ (d-2\,e)$$

```
[In] int(((d + e*x)*(x^2 - 3*x + 2))/(x^4 - 5*x^2 + 4),x)
[Out] log(x + 1)*(d - e) - log(x + 2)*(d - 2*e)
```

3.75 $\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 29

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx+(d-e+f)\log(1+x)-(d-2e+4f)\log(2+x)$$

```
[Out] f*x+(d-e+f)*ln(1+x)-(d-2*e+4*f)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.129, Rules used = {1600, 1671, 646, 31}

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = \log(x+1)(d-e+f)-\log(x+2)(d-2e+4f)+fx$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4),x]
[Out] f*x + (d - e + f)*Log[1 + x] - (d - 2*e + 4*f)*Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
```

```
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{2+3x+x^2}\,dx \\
&= \int \left(f+\frac{d-2f+(e-3f)x}{2+3x+x^2}\right)dx \\
&= fx+\int \frac{d-2f+(e-3f)x}{2+3x+x^2}\,dx \\
&= fx+(d-e+f)\int \frac{1}{1+x}\,dx-(d-2e+4f)\int \frac{1}{2+x}\,dx \\
&= fx+(d-e+f)\log(1+x)-(d-2e+4f)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 30, normalized size of antiderivative = 1.03

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx+(d-e+f)\log(1+x)+(-d+2e-4f)\log(2+x)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4),x]
[Out] f*x + (d - e + f)*Log[1 + x] + (-d + 2*e - 4*f)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 31, normalized size of antiderivative = 1.07

method	result
default	$fx + (-d + 2e - 4f) \ln(x+2) + (d - e + f) \ln(x+1)$
norman	$fx + (-d + 2e - 4f) \ln(x+2) + (d - e + f) \ln(x+1)$
parallelrisch	$\ln(x+1) d - \ln(x+1) e + \ln(x+1) f - \ln(x+2) d + 2\ln(x+2) e - 4\ln(x+2) f + fx$
risch	$fx + \ln(-x-1) d - \ln(-x-1) e + \ln(-x-1) f - \ln(x+2) d + 2\ln(x+2) e - 4\ln(x+$

```
[In] int((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] f*x+(-d+2*e-4*f)*ln(x+2)+(d-e+f)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx - (d - 2e + 4f)\log(x+2) + (d - e + f)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] f*x - (d - 2*e + 4*f)*log(x + 2) + (d - e + f)*log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 0.30 (sec) , antiderivative size = 44, normalized size of antiderivative = 1.52

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx + (-d + 2e - 4f)\log\left(x + \frac{4d - 6e + 10f}{2d - 3e + 5f}\right) + (d - e + f)\log(x+1)$$

```
[In] integrate((x**2-3*x+2)*(f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] f*x + (-d + 2*e - 4*f)*log(x + (4*d - 6*e + 10*f)/(2*d - 3*e + 5*f)) + (d -
 e + f)*log(x + 1)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx-(d-2e+4f)\log(x+2)+(d-e+f)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] f*x - (d - 2*e + 4*f)*log(x + 2) + (d - e + f)*log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 31, normalized size of antiderivative = 1.07

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = fx-(d-2e+4f)\log(|x+2|) + (d-e+f)\log(|x+1|)$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] f*x - (d - 2*e + 4*f)*log(abs(x + 2)) + (d - e + f)*log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{4-5x^2+x^4}\,dx = f\,x+\ln(x+1)\,(d-e+f)-\ln(x+2)\,(d-2e+4f)$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2))/(x^4 - 5*x^2 + 4),x)
[Out] f*x + log(x + 1)*(d - e + f) - log(x + 2)*(d - 2*e + 4*f)
```

3.76 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 47

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = (f-3g)x+\frac{gx^2}{2}+(d-e+f-g)\log(1+x) - (d-2e+4f-8g)\log(2+x)$$

```
[Out] (f-3*g)*x+1/2*g*x^2+(d-e+f-g)*ln(1+x)-(d-2*e+4*f-8*g)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.111, Rules used = {1600, 1671, 646, 31}

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \log(x+1)(d-e+f-g) - \log(x+2)(d-2e+4f-8g)+x(f-3g)+\frac{gx^2}{2}$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4),x]
```

```
[Out] (f - 3*g)*x + (g*x^2)/2 + (d - e + f - g)*Log[1 + x] - (d - 2*e + 4*f - 8*g
)*Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3}{2+3x+x^2}\,dx \\
&= \int \left(f-3g+gx+\frac{d-2f+6g+(e-3f+7g)x}{2+3x+x^2}\right)dx \\
&= (f-3g)x+\frac{gx^2}{2}+\int \frac{d-2f+6g+(e-3f+7g)x}{2+3x+x^2}\,dx \\
&= (f-3g)x+\frac{gx^2}{2}-(d-2e+4f-8g)\int \frac{1}{2+x}\,dx+(d-e+f-g)\int \frac{1}{1+x}\,dx \\
&= (f-3g)x+\frac{gx^2}{2}+(d-e+f-g)\log(1+x)-(d-2e+4f-8g)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.07 (sec) , antiderivative size = 44, normalized size of antiderivative = 0.94

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = fx+\frac{1}{2}g(-6+x)x+(d-e+f-g)\log(1+x) - (d-2e+4f-8g)\log(2+x)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4),x]
[Out] f*x + (g*(-6 + x)*x)/2 + (d - e + f - g)*Log[1 + x] - (d - 2*e + 4*f - 8*g)
*Log[2 + x]
```

Maple [A] (verified)

Time = 0.05 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00

method	result
default	$\frac{g\,x^{2}}{2}+f x-3 g x+\left(-d+2 e-4 f+8 g\right) \ln \left(x+2\right)+\left(d-e+f-g\right) \ln \left(x+1\right)$
norman	$\left(f-3 g\right) x+\frac{g\,x^{2}}{2}+\left(-d+2 e-4 f+8 g\right) \ln \left(x+2\right)+\left(d-e+f-g\right) \ln \left(x+1\right)$
parallelrisch	$\frac{g\,x^{2}}{2}+f x-3 g x+\ln \left(x+1\right) d-\ln \left(x+1\right) e+\ln \left(x+1\right) f-\ln \left(x+1\right) g-\ln \left(x+2\right) d+2 \ln$
risch	$\frac{g\,x^{2}}{2}+f x-3 g x+\ln \left(-x-1\right) d-\ln \left(-x-1\right) e+\ln \left(-x-1\right) f-\ln \left(-x-1\right) g-\ln \left(x+2\right)$

```
[In] int((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] 1/2*g*x^2+f*x-3*g*x+(-d+2*e-4*f+8*g)*ln(x+2)+(d-e+f-g)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.96

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}\,gx^2+(f-3\,g)x - (d-2\,e+4\,f-8\,g)\log\left(x+2\right) + (d-e+f-g)\log\left(x+1\right)$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="frica
s")
[Out] 1/2*g*x^2 + (f - 3*g)*x - (d - 2*e + 4*f - 8*g)*log(x + 2) + (d - e + f - g
)*log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 0.48 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.40

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \frac{gx^2}{2}+x(f-3g)+(-d+2e-4f+8g)\log\left(x+\frac{4d-6e+10f-18g}{2d-3e+5f-9g}\right) + (d-e+f-g)\log\left(x+1\right)$$

```
[In] integrate((x**2-3*x+2)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] g*x**2/2 + x*(f - 3*g) + (-d + 2*e - 4*f + 8*g)*log(x + (4*d - 6*e + 10*f -
 18*g)/(2*d - 3*e + 5*f - 9*g)) + (d - e + f - g)*log(x + 1)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.96

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}\,gx^2+(f-3\,g)x - (d-2\,e+4\,f-8\,g)\log(x+2) + (d-e+f-g)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxim
a")
```

```
[Out] 1/2*g*x^2 + (f - 3*g)*x - (d - 2*e + 4*f - 8*g)*log(x + 2) + (d - e + f - g
)*log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \frac{1}{2}\,gx^2+fx-3\,gx - (d-2\,e+4\,f-8\,g)\log(|x+2|) + (d-e+f-g)\log(|x+1|)$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac"
)
```

```
[Out] 1/2*g*x^2 + f*x - 3*g*x - (d - 2*e + 4*f - 8*g)*log(abs(x + 2)) + (d - e +
f - g)*log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.96

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{4-5x^2+x^4}\,dx = \ln(x+1)\,(d-e+f-g)+x\,(f-3\,g) + \frac{g\,x^2}{2} - \ln(x+2)\,(d-2\,e+4\,f-8\,g)$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3))/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x + 1)*(d - e + f - g) + x*(f - 3*g) + (g*x^2)/2 - log(x + 2)*(d - 2*e
+ 4*f - 8*g)
```

3.77 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 41, antiderivative size = 66

$$\begin{aligned}&\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx\\&=(f-3g+7h)x+\frac{1}{2}(g-3h)x^2+\frac{hx^3}{3}+(d-e+f-g+h)\log(1+x)\\&\quad-(d-2e+4f-8g+16h)\log(2+x)\end{aligned}$$

```
[Out] (f-3*g+7*h)*x+1/2*(g-3*h)*x^2+1/3*h*x^3+(d-e+f-g+h)*ln(1+x)-(d-2*e+4*f-8*g+
16*h)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.098, Rules used = {1600, 1671, 646, 31}

$$\begin{aligned}&\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx\\&=\log(x+1)(d-e+f-g+h)-\log(x+2)(d-2e+4f-8g+16h)\\&\quad+x(f-3g+7h)+\frac{1}{2}x^2(g-3h)+\frac{hx^3}{3}\end{aligned}$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4),x
]
```

```
[Out] (f - 3*g + 7*h)*x + ((g - 3*h)*x^2)/2 + (h*x^3)/3 + (d - e + f - g + h)*Log
[1 + x] - (d - 2*e + 4*f - 8*g + 16*h)*Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4}{2+3x+x^2}\,dx \\
&= \int \left(f-3g+7h+(g-3h)x+hx^2+\frac{d-2f+6g-14h+(e-3f+7g-15h)x}{2+3x+x^2}\right)dx \\
&= (f-3g+7h)x+\frac{1}{2}(g-3h)x^2+\frac{hx^3}{3}+\int \frac{d-2f+6g-14h+(e-3f+7g-15h)x}{2+3x+x^2}\,dx \\
&= (f-3g+7h)x+\frac{1}{2}(g-3h)x^2+\frac{hx^3}{3}+(d-e+f-g+h)\int \frac{1}{1+x}\,dx \\
&\quad -(d-2e+4f-8g+16h)\int \frac{1}{2+x}\,dx \\
&= (f-3g+7h)x+\frac{1}{2}(g-3h)x^2+\frac{hx^3}{3}+(d-e+f-g+h)\log(1+x) \\
&\quad -(d-2e+4f-8g+16h)\log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 67, normalized size of antiderivative = 1.02

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$$
$$=(f-3g+7h)x+\frac{1}{2}(g-3h)x^2+\frac{hx^3}{3}+(d-e+f-g+h)\log(1+x)$$
$$+(-d+2e-4f+8g-16h)\log(2+x)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 +
x^4),x]
```

```
[Out] (f - 3*g + 7*h)*x + ((g - 3*h)*x^2)/2 + (h*x^3)/3 + (d - e + f - g + h)*Log
[1 + x] + (-d + 2*e - 4*f + 8*g - 16*h)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.98

method	result
norman	$\left(\frac{g}{2}-\frac{3h}{2}\right)x^2+(f-3g+7h)x+\frac{h\,x^3}{3}+(-d+2e-4f+8g-16h)\ln(x+2)+(d-e+f-$
default	$\frac{h\,x^3}{3}+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+fx-3gx+7hx+(-d+2e-4f+8g-16h)\ln(x+2)+(d-e+f-$
parallelrisch	$\frac{h\,x^3}{3}+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+fx-3gx+7hx+\ln(x+1)\,d-\ln(x+1)\,e+\ln(x+1)\,f-\ln(x+1)\,g$
risch	$\frac{h\,x^3}{3}+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+fx-3gx+7hx+\ln(-x-1)\,d-\ln(-x-1)\,e+\ln(-x-1)\,f-\ln(-$

```
[In] int((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVER
BOSE)
```

```
[Out] (1/2*g-3/2*h)*x^2+(f-3*g+7*h)*x+1/3*h*x^3+(-d+2*e-4*f+8*g-16*h)*ln(x+2)+(d-
e+f-g+h)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.94

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{3}hx^3+\frac{1}{2}(g-3h)x^2+(f-3g+7h)x$$
$$-(d-2e+4f-8g+16h)\log(x+2)+(d-e+f-g+h)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"fricas")
```

```
[Out] 1/3*h*x^3 + 1/2*(g - 3*h)*x^2 + (f - 3*g + 7*h)*x - (d - 2*e + 4*f - 8*g +
16*h)*log(x + 2) + (d - e + f - g + h)*log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 0.85 (sec) , antiderivative size = 94, normalized size of antiderivative = 1.42

$$\begin{aligned}&\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx\\&=\frac{hx^3}{3}+x^2\left(\frac{g}{2}-\frac{3h}{2}\right)+x(f-3g+7h)\\&\quad+(-d+2e-4f+8g-16h)\log\left(x+\frac{4d-6e+10f-18g+34h}{2d-3e+5f-9g+17h}\right)\\&\quad+(d-e+f-g+h)\log(x+1)\end{aligned}$$

```
[In] integrate((x**2-3*x+2)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] h*x**3/3 + x**2*(g/2 - 3*h/2) + x*(f - 3*g + 7*h) + (-d + 2*e - 4*f + 8*g -
 16*h)*log(x + (4*d - 6*e + 10*f - 18*g + 34*h)/(2*d - 3*e + 5*f - 9*g + 17
*h)) + (d - e + f - g + h)*log(x + 1)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.94

$$\begin{aligned}&\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx\\&=\frac{1}{3}\,hx^3+\frac{1}{2}\,(g-3\,h)x^2+(f-3\,g+7\,h)x\\&\quad-(d-2\,e+4\,f-8\,g+16\,h)\log(x+2)+(d-e+f-g+h)\log(x+1)\end{aligned}$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"maxima")
```

```
[Out] 1/3*h*x^3 + 1/2*(g - 3*h)*x^2 + (f - 3*g + 7*h)*x - (d - 2*e + 4*f - 8*g +
16*h)*log(x + 2) + (d - e + f - g + h)*log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 67, normalized size of antiderivative = 1.02

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$$
$$=\frac{1}{3}\,hx^3+\frac{1}{2}\,gx^2-\frac{3}{2}\,hx^2+fx-3\,gx+7\,hx$$
$$-(d-2\,e+4\,f-8\,g+16\,h)\log\left(|x+2|\right)+(d-e+f-g+h)\log\left(|x+1|\right)$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"giac")
```

```
[Out] 1/3*h*x^3 + 1/2*g*x^2 - 3/2*h*x^2 + f*x - 3*g*x + 7*h*x - (d - 2*e + 4*f -
8*g + 16*h)*log(abs(x + 2)) + (d - e + f - g + h)*log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 63, normalized size of antiderivative = 0.95

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx$$
$$=x^2\left(\frac{g}{2}-\frac{3\,h}{2}\right)+x\,(f-3\,g+7\,h)-\ln\left(x+2\right)\,\left(d-2\,e+4\,f-8\,g+16\,h\right)$$
$$+\frac{h\,x^3}{3}+\ln\left(x+1\right)\,\left(d-e+f-g+h\right)$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2 + 4),x
)
```

```
[Out] x^2*(g/2 - (3*h)/2) + x*(f - 3*g + 7*h) - log(x + 2)*(d - 2*e + 4*f - 8*g +
 16*h) + (h*x^3)/3 + log(x + 1)*(d - e + f - g + h)
```

3.78 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 46, antiderivative size = 90

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx &= (f-3g+7h-15i)x+\frac{1}{2}(g-3h+7i)x^2+\frac{1}{3}(h-3i)x^3+\frac{ix^4}{4} \\ &\quad +(d-e+f-g+h-i)\log(1+x)-(d-2e+4f-8g+16h-32i)\log(2+x)\end{aligned}$$

```
[Out] (f-3*g+7*h-15*i)*x+1/2*(g-3*h+7*i)*x^2+1/3*(h-3*i)*x^3+1/4*i*x^4+(d-e+f-g+h
-i)*ln(1+x)-(d-2*e+4*f-8*g+16*h-32*i)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 90, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.087, Rules used = {1600, 1671, 646, 31}

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx &= \log(x+1)(d-e+f-g+h-i)-\log(x+2)(d-2e+4f-8g+16h-32i) \\ &\quad +x(f-3g+7h-15i)+\frac{1}{2}x^2(g-3h+7i)+\frac{1}{3}x^3(h-3i)+\frac{ix^4}{4}\end{aligned}$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2
+ x^4),x]
```

```
[Out] (f - 3*g + 7*h - 15*i)*x + ((g - 3*h + 7*i)*x^2)/2 + ((h - 3*i)*x^3)/3 + (i
*x^4)/4 + (d - e + f - g + h - i)*Log[1 + x] - (d - 2*e + 4*f - 8*g + 16*h
- 32*i)*Log[2 + x]
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 1671

```
Int[(Pq_)*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_.), x_Symbol] :> Int[Expand
Integrand[Pq*(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c}, x] && PolyQ[Pq
, x] && IGtQ[p, -2]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{2+3x+x^2}\,dx\\
&= \int \left(f-3g+7h-15i+(g-3h+7i)x+(h-3i)x^2+ix^3 \right.\\
&\qquad \left. +\frac{d-2f+6g-14h+30i+(e-3f+7g-15h+31i)x}{2+3x+x^2}\right)dx\\
&= (f-3g+7h-15i)x+\frac{1}{2}(g-3h+7i)x^2+\frac{1}{3}(h-3i)x^3+\frac{ix^4}{4}\\
&\quad +\int \frac{d-2f+6g-14h+30i+(e-3f+7g-15h+31i)x}{2+3x+x^2}\,dx\\
&= (f-3g+7h-15i)x+\frac{1}{2}(g-3h+7i)x^2+\frac{1}{3}(h-3i)x^3+\frac{ix^4}{4}\\
&\quad -(d-2e+4f-8g+16h-32i)\int \frac{1}{2+x}\,dx+(d-e+f-g+h-i)\int \frac{1}{1+x}\,dx\\
&= (f-3g+7h-15i)x+\frac{1}{2}(g-3h+7i)x^2+\frac{1}{3}(h-3i)x^3+\frac{ix^4}{4}\\
&\quad +(d-e+f-g+h-i)\log(1+x)-(d-2e+4f-8g+16h-32i)\log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 91, normalized size of antiderivative = 1.01

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$= (f-3g+7h-15i)x+\frac{1}{2}(g-3h+7i)x^2+\frac{1}{3}(h-3i)x^3+\frac{ix^4}{4} + (d-e+f-g+h-i)\log(1+x)+(-d+2e-4f+8g-16h+32i)\log(2+x)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 -
5*x^2 + x^4),x]
```

```
[Out] (f - 3*g + 7*h - 15*i)*x + ((g - 3*h + 7*i)*x^2)/2 + ((h - 3*i)*x^3)/3 + (i
*x^4)/4 + (d - e + f - g + h - i)*Log[1 + x] + (-d + 2*e - 4*f + 8*g - 16*h
 + 32*i)*Log[2 + x]
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 88, normalized size of antiderivative = 0.98

method	result
norman	$\left(\frac{h}{3}-i\right)x^3+\left(\frac{g}{2}-\frac{3h}{2}+\frac{7i}{2}\right)x^2+\left(f-3g+7h-15i\right)x+\frac{i\,x^4}{4}+\left(-d+2e-4f+8g-16h-\right.$
default	$\frac{i\,x^4}{4}+\frac{h\,x^3}{3}-i\,x^3+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+\frac{7i\,x^2}{2}+fx-3gx+7hx-15ix+\left(-d+2e-4f+8g-16\right.$
parallelrisch	$\frac{i\,x^4}{4}+\frac{h\,x^3}{3}-i\,x^3+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+\frac{7i\,x^2}{2}+fx-3gx+7hx-15ix+\ln\left(x+1\right)d-\ln\left(x+1\right)e$
risch	$\frac{i\,x^4}{4}+\frac{h\,x^3}{3}-i\,x^3+\frac{g\,x^2}{2}-\frac{3h\,x^2}{2}+\frac{7i\,x^2}{2}+fx-3gx+7hx-15ix+\ln\left(-x-1\right)d-\ln\left(-x-\right.$

```
[In] int((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RET
URNVERBOSE)
```

```
[Out] (1/3*h-i)*x^3+(1/2*g-3/2*h+7/2*i)*x^2+(f-3*g+7*h-15*i)*x+1/4*i*x^4+(-d+2*e-
4*f+8*g-16*h+32*i)*ln(x+2)+(d-e+f-g+h-i)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.93

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx = \frac{1}{4}\,ix^4 + \frac{1}{3}\,(h-3\,i)x^3 + \frac{1}{2}\,(g-3\,h+7\,i)x^2 + (f-3\,g+7\,h-15\,i)x - (d-2\,e+4\,f-8\,g+16\,h-32\,i)\log(x+2) + (d-e+f-g+h-i)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algo
rithm="fricas")
```

```
[Out] 1/4*i*x^4 + 1/3*(h - 3*i)*x^3 + 1/2*(g - 3*h + 7*i)*x^2 + (f - 3*g + 7*h -
15*i)*x - (d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(x + 2) + (d - e + f - g +
 h - i)*log(x + 1)
```

Sympy [A] (verification not implemented)

Time = 1.39 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.36

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx = \frac{ix^4}{4} + x^3\left(\frac{h}{3}-i\right) + x^2\left(\frac{g}{2}-\frac{3h}{2}+\frac{7i}{2}\right) + x(f-3g+7h-15i) + (-d+2e-4f+8g-16h+32i)\log\left(x+\frac{4d-6e+10f-18g+34h-66i}{2d-3e+5f-9g+17h-33i}\right) + (d-e+f-g+h-i)\log(x+1)$$

```
[In] integrate((x**2-3*x+2)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),
x)
```

```
[Out] i*x**4/4 + x**3*(h/3 - i) + x**2*(g/2 - 3*h/2 + 7*i/2) + x*(f - 3*g + 7*h -
 15*i) + (-d + 2*e - 4*f + 8*g - 16*h + 32*i)*log(x + (4*d - 6*e + 10*f - 1
8*g + 34*h - 66*i)/(2*d - 3*e + 5*f - 9*g + 17*h - 33*i)) + (d - e + f - g
+ h - i)*log(x + 1)
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.93

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$= \frac{1}{4}\,ix^4 + \frac{1}{3}\,(h-3\,i)x^3 + \frac{1}{2}\,(g-3\,h+7\,i)x^2 + (f-3\,g+7\,h-15\,i)x$$
$$- (d-2\,e+4\,f-8\,g+16\,h-32\,i)\log(x+2) + (d-e+f-g+h-i)\log(x+1)$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algo
rithm="maxima")
```

```
[Out] 1/4*i*x^4 + 1/3*(h - 3*i)*x^3 + 1/2*(g - 3*h + 7*i)*x^2 + (f - 3*g + 7*h -
15*i)*x - (d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(x + 2) + (d - e + f - g +
 h - i)*log(x + 1)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.06

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$= \frac{1}{4}\,ix^4 + \frac{1}{3}\,hx^3 - ix^3 + \frac{1}{2}\,gx^2 - \frac{3}{2}\,hx^2 + \frac{7}{2}\,ix^2 + fx - 3\,gx + 7\,hx - 15\,ix$$
$$- (d-2\,e+4\,f-8\,g+16\,h-32\,i)\log(|x+2|) + (d-e+f-g+h-i)\log(|x+1|)$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algo
rithm="giac")
```

```
[Out] 1/4*i*x^4 + 1/3*h*x^3 - i*x^3 + 1/2*g*x^2 - 3/2*h*x^2 + 7/2*i*x^2 + f*x - 3
*g*x + 7*h*x - 15*i*x - (d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(abs(x + 2))
 + (d - e + f - g + h - i)*log(abs(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.05 (sec) , antiderivative size = 86, normalized size of antiderivative = 0.96

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx$$
$$= x^3\left(\frac{h}{3}-i\right) - \ln(x+2)\ (d-2e+4f-8g+16h-32i)$$
$$+\ln(x+1)\ (d-e+f-g+h-i) + \frac{i\,x^4}{4} + x^2\left(\frac{g}{2}-\frac{3h}{2}+\frac{7i}{2}\right) + x\,(f-3g+7h-15i)$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4 - 5*x^
2 + 4),x)
```

```
[Out] x^3*(h/3 - i) - log(x + 2)*(d - 2*e + 4*f - 8*g + 16*h - 32*i) + log(x + 1)
*(d - e + f - g + h - i) + (i*x^4)/4 + x^2*(g/2 - (3*h)/2 + (7*i)/2) + x*(f
 - 3*g + 7*h - 15*i)
```

3.79 $\int \frac{2+x}{4-5x^2+x^4} \, dx$

Optimal result

Integrand size = 16, antiderivative size = 29

$$\int \frac{2+x}{4-5x^2+x^4} \, dx = -\frac{1}{2}\log(1-x)+\frac{1}{3}\log(2-x)+\frac{1}{6}\log(1+x)$$

```
[Out] -1/2*ln(1-x)+1/3*ln(2-x)+1/6*ln(1+x)
```

Rubi [A] (verified)

Time = 0.01 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1600, 2083}

$$\int \frac{2+x}{4-5x^2+x^4} \, dx = -\frac{1}{2}\log(1-x)+\frac{1}{3}\log(2-x)+\frac{1}{6}\log(x+1)$$

```
[In] Int[(2 + x)/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/2*Log[1 - x] + Log[2 - x]/3 + Log[1 + x]/6
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2083

```
Int[(P_)^(p_), x_Symbol] :> With[{u = Factor[P]}, Int[ExpandIntegrand[u^p,
x], x] /;  !SumQ[NonfreeFactors[u, x]]] /; PolyQ[P, x] && ILtQ[p, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{1}{2-x-2x^2+x^3}\,dx\\ &= \int \left(\frac{1}{3(-2+x)}-\frac{1}{2(-1+x)}+\frac{1}{6(1+x)}\right)dx\\ &= -\frac{1}{2}\log(1-x)+\frac{1}{3}\log(2-x)+\frac{1}{6}\log(1+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.01 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00

$$\int \frac{2+x}{4-5x^2+x^4}\,dx = -\frac{1}{2}\log(1-x)+\frac{1}{3}\log(2-x)+\frac{1}{6}\log(1+x)$$

```
[In] Integrate[(2 + x)/(4 - 5*x^2 + x^4),x]
[Out] -1/2*Log[1 - x] + Log[2 - x]/3 + Log[1 + x]/6
```

Maple [A] (verified)

Time = 0.04 (sec) , antiderivative size = 20, normalized size of antiderivative = 0.69

method	result	size
default	$\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{2}+\frac{\ln(x-2)}{3}$	20
norman	$\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{2}+\frac{\ln(x-2)}{3}$	20
risch	$\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{2}+\frac{\ln(x-2)}{3}$	20
parallelrisch	$\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{2}+\frac{\ln(x-2)}{3}$	20

```
[In] int((x+2)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] 1/6*ln(x+1)-1/2*ln(x-1)+1/3*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 19, normalized size of antiderivative = 0.66

$$\int \frac{2+x}{4-5x^2+x^4}\,dx = \frac{1}{6}\,\log\left(x+1\right) - \frac{1}{2}\,\log\left(x-1\right) + \frac{1}{3}\,\log\left(x-2\right)$$

```
[In] integrate((2+x)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] 1/6*log(x + 1) - 1/2*log(x - 1) + 1/3*log(x - 2)
```

Sympy [A] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 19, normalized size of antiderivative = 0.66

$$\int \frac{2+x}{4-5x^2+x^4}\,dx = \frac{\log\left(x-2\right)}{3} - \frac{\log\left(x-1\right)}{2} + \frac{\log\left(x+1\right)}{6}$$

```
[In] integrate((2+x)/(x**4-5*x**2+4),x)
[Out] log(x - 2)/3 - log(x - 1)/2 + log(x + 1)/6
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 19, normalized size of antiderivative = 0.66

$$\int \frac{2+x}{4-5x^2+x^4}\,dx = \frac{1}{6}\,\log\left(x+1\right) - \frac{1}{2}\,\log\left(x-1\right) + \frac{1}{3}\,\log\left(x-2\right)$$

```
[In] integrate((2+x)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] 1/6*log(x + 1) - 1/2*log(x - 1) + 1/3*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 22, normalized size of antiderivative = 0.76

$$\int \frac{2+x}{4-5x^2+x^4}\,dx = \frac{1}{6}\,\log\left(|x+1|\right) - \frac{1}{2}\,\log\left(|x-1|\right) + \frac{1}{3}\,\log\left(|x-2|\right)$$

```
[In] integrate((2+x)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] 1/6*log(abs(x + 1)) - 1/2*log(abs(x - 1)) + 1/3*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.04 (sec) , antiderivative size = 19, normalized size of antiderivative = 0.66

$$\int \frac{2+x}{4-5x^2+x^4} \, dx = \frac{\ln(x+1)}{6} - \frac{\ln(x-1)}{2} + \frac{\ln(x-2)}{3}$$

```
[In] int((x + 2)/(x^4 - 5*x^2 + 4),x)
[Out] log(x + 1)/6 - log(x - 1)/2 + log(x - 2)/3
```

3.80 $\int \frac{(2+x)(d+ex)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 21, antiderivative size = 42

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4} dx = -\frac{1}{2}(d+e)\log(1-x) + \frac{1}{3}(d+2e)\log(2-x) + \frac{1}{6}(d-e)\log(1+x)$$

```
[Out] -1/2*(d+e)*ln(1-x)+1/3*(d+2*e)*ln(2-x)+1/6*(d-e)*ln(1+x)
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 42, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.095$, Rules used = {1600, 2099}

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4} dx = -\frac{1}{2}(d+e)\log(1-x) + \frac{1}{3}(d+2e)\log(2-x) + \frac{1}{6}(d-e)\log(x+1)$$

```
[In] Int[((2 + x)*(d + e*x))/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/2*((d + e)*Log[1 - x]) + ((d + 2*e)*Log[2 - x])/3 + ((d - e)*Log[1 + x])
/6
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
```

```
PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex}{2-x-2x^2+x^3}\,dx \\
&= \int \left(\frac{d+2e}{3(-2+x)}+\frac{-d-e}{2(-1+x)}+\frac{d-e}{6(1+x)}\right)dx \\
&= -\frac{1}{2}(d+e)\log(1-x)+\frac{1}{3}(d+2e)\log(2-x)+\frac{1}{6}(d-e)\log(1+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.01 (sec) , antiderivative size = 39, normalized size of antiderivative = 0.93

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \frac{1}{6}(-3(d+e)\log(1-x)+2(d+2e)\log(2-x)+(d-e)\log(1+x))$$

```
[In] Integrate[((2 + x)*(d + e*x))/(4 - 5*x^2 + x^4),x]
[Out] (-3*(d + e)*Log[1 - x] + 2*(d + 2*e)*Log[2 - x] + (d - e)*Log[1 + x])/6
```

Maple [A] (verified)

Time = 0.05 (sec) , antiderivative size = 38, normalized size of antiderivative = 0.90

method	result	size
default	$\left(\frac{d}{6}-\frac{e}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}\right)\ln\left(x-2\right)$	38
norman	$\left(\frac{d}{6}-\frac{e}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}\right)\ln\left(x-2\right)$	38
parallelrisch	$\frac{\ln(x-2)d}{3}+\frac{2\ln(x-2)e}{3}-\frac{\ln(x-1)d}{2}-\frac{\ln(x-1)e}{2}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}$	44
risch	$\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}-\frac{\ln(1-x)d}{2}-\frac{\ln(1-x)e}{2}+\frac{\ln(2-x)d}{3}+\frac{2\ln(2-x)e}{3}$	52

```
[In] int((x+2)*(e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] (1/6*d-1/6*e)*ln(x+1)+(-1/2*d-1/2*e)*ln(x-1)+(1/3*d+2/3*e)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 32, normalized size of antiderivative = 0.76

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \frac{1}{6}\,(d-e)\log\left(x+1\right) - \frac{1}{2}\,(d+e)\log\left(x-1\right) + \frac{1}{3}\,(d+2\,e)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] 1/6*(d - e)*log(x + 1) - 1/2*(d + e)*log(x - 1) + 1/3*(d + 2*e)*log(x - 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 304 vs. 2(37) = 74.

Time = 1.11 (sec) , antiderivative size = 304, normalized size of antiderivative = 7.24

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \frac{(d-e)\log\left(x+\frac{26d^3+66d^2e-9d^2(d-e)+78de^2-12de(d-e)-7d(d-e)^2+46e^3+3e^2(d-e)-8e(d-e)^2}{10d^3+69d^2e+102de^2+35e^3}\right)}{6} - \frac{(d+e)\log\left(x+\frac{26d^3+66d^2e+27d^2(d+e)+78de^2+36de(d+e)-63d(d+e)^2+46e^3-9e^2(d+e)-72e(d+e)^2}{10d^3+69d^2e+102de^2+35e^3}\right)}{2} + \frac{(d+2e)\log\left(x+\frac{26d^3+66d^2e-18d^2(d+2e)+78de^2-24de(d+2e)-28d(d+2e)^2+46e^3+6e^2(d+2e)-32e(d+2e)^2}{10d^3+69d^2e+102de^2+35e^3}\right)}{3}$$

```
[In] integrate((2+x)*(e*x+d)/(x**4-5*x**2+4),x)
[Out] (d - e)*log(x + (26*d**3 + 66*d**2*e - 9*d**2*(d - e) + 78*d*e**2 - 12*d*e*
(d - e) - 7*d*(d - e)**2 + 46*e**3 + 3*e**2*(d - e) - 8*e*(d - e)**2)/(10*d
**3 + 69*d**2*e + 102*d*e**2 + 35*e**3))/6 - (d + e)*log(x + (26*d**3 + 66*
d**2*e + 27*d**2*(d + e) + 78*d*e**2 + 36*d*e*(d + e) - 63*d*(d + e)**2 + 4
6*e**3 - 9*e**2*(d + e) - 72*e*(d + e)**2)/(10*d**3 + 69*d**2*e + 102*d*e**
2 + 35*e**3))/2 + (d + 2*e)*log(x + (26*d**3 + 66*d**2*e - 18*d**2*(d + 2*e
) + 78*d*e**2 - 24*d*e*(d + 2*e) - 28*d*(d + 2*e)**2 + 46*e**3 + 6*e**2*(d
+ 2*e) - 32*e*(d + 2*e)**2)/(10*d**3 + 69*d**2*e + 102*d*e**2 + 35*e**3))/3
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 32, normalized size of antiderivative = 0.76

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \frac{1}{6}\,(d-e)\log{(x+1)} - \frac{1}{2}\,(d+e)\log{(x-1)} + \frac{1}{3}\,(d+2\,e)\log{(x-2)}$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
[Out] 1/6*(d - e)*log(x + 1) - 1/2*(d + e)*log(x - 1) + 1/3*(d + 2*e)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 35, normalized size of antiderivative = 0.83

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \frac{1}{6}\,(d-e)\log{(|x+1|)} - \frac{1}{2}\,(d+e)\log{(|x-1|)} + \frac{1}{3}\,(d+2\,e)\log{(|x-2|)}$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
[Out] 1/6*(d - e)*log(abs(x + 1)) - 1/2*(d + e)*log(abs(x - 1)) + 1/3*(d + 2*e)*l
og(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.78 (sec) , antiderivative size = 38, normalized size of antiderivative = 0.90

$$\int \frac{(2+x)(d+ex)}{4-5x^2+x^4}\,dx = \ln{(x-2)}\,\left(\frac{d}{3}+\frac{2\,e}{3}\right) - \ln{(x-1)}\,\left(\frac{d}{2}+\frac{e}{2}\right) + \ln{(x+1)}\,\left(\frac{d}{6}-\frac{e}{6}\right)$$

```
[In] int(((x + 2)*(d + e*x))/(x^4 - 5*x^2 + 4),x)
[Out] log(x - 2)*(d/3 + (2*e)/3) - log(x - 1)*(d/2 + e/2) + log(x + 1)*(d/6 - e/6
)
```

3.81 $\int \frac{(2+x)(d+ex+fx^2)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 26, antiderivative size = 47

$$\int \frac{(2+x)(d+ex+fx^2)}{4-5x^2+x^4} dx = -\frac{1}{2}(d+e+f)\log(1-x)+\frac{1}{3}(d+2e+4f)\log(2-x) + \frac{1}{6}(d-e+f)\log(1+x)$$

```
[Out] -1/2*(d+e+f)*ln(1-x)+1/3*(d+2*e+4*f)*ln(2-x)+1/6*(d-e+f)*ln(1+x)
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.077, Rules used = {1600, 2099}

$$\int \frac{(2+x)(d+ex+fx^2)}{4-5x^2+x^4} dx = -\frac{1}{2}\log(1-x)(d+e+f)+\frac{1}{3}\log(2-x)(d+2e+4f) + \frac{1}{6}\log(x+1)(d-e+f)$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4),x]
```

```
[Out] -1/2*((d + e + f)*Log[1 - x]) + ((d + 2*e + 4*f)*Log[2 - x])/3 + ((d - e + f)*Log[1 + x])/6
```

<u>Rule 1600</u>

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{2-x-2x^2+x^3}\,dx \\
&= \int \left(\frac{d+2e+4f}{3(-2+x)} + \frac{-d-e-f}{2(-1+x)} + \frac{d-e+f}{6(1+x)}\right)dx \\
&= -\frac{1}{2}(d+e+f)\log(1-x) + \frac{1}{3}(d+2e+4f)\log(2-x) + \frac{1}{6}(d-e+f)\log(1+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 44, normalized size of antiderivative = 0.94

$$
\int \frac{(2+x)\left(d+ex+fx^2\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}(-3(d+e+f)\log(1-x)+2(d+2e+4f)\log(2-x) + (d-e+f)\log(1+x))
$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4),x]
[Out] (-3*(d + e + f)*Log[1 - x] + 2*(d + 2*e + 4*f)*Log[2 - x] + (d - e + f)*Log
[1 + x])/6
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00

method	result
default	$\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}\right)\ln\left(x-2\right)$
norman	$\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}\right)\ln\left(x-2\right)$
parallelrisch	$\frac{\ln(x-2)d}{3}+\frac{2\ln(x-2)e}{3}+\frac{4\ln(x-2)f}{3}-\frac{\ln(x-1)d}{2}-\frac{\ln(x-1)e}{2}-\frac{\ln(x-1)f}{2}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}$
risch	$\frac{\ln(2-x)d}{3}+\frac{2\ln(2-x)e}{3}+\frac{4\ln(2-x)f}{3}-\frac{\ln(1-x)d}{2}-\frac{\ln(1-x)e}{2}-\frac{\ln(1-x)f}{2}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}$

```
[In] int((x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
[Out] (1/6*d-1/6*e+1/6*f)*ln(x+1)+(-1/2*d-1/2*e-1/2*f)*ln(x-1)+(1/3*d+2/3*e+4/3*f
)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 37, normalized size of antiderivative = 0.79

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}\left(d-e+f\right)\log\left(x+1\right) - \frac{1}{2}\left(d+e+f\right)\log\left(x-1\right) + \frac{1}{3}\left(d+2\,e+4\,f\right)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
[Out] 1/6*(d - e + f)*log(x + 1) - 1/2*(d + e + f)*log(x - 1) + 1/3*(d + 2*e + 4*
f)*log(x - 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 716 vs. 2(49) = 98.

Time = 7.75 (sec) , antiderivative size = 716, normalized size of antiderivative = 15.23

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{4-5x^2+x^4}\,dx = \frac{(d-e+f)\log\left(x+\frac{26d^3+66d^2e+132d^2f-9d^2(d-e+f)+78de^2+276def-12de(d-e+f)+222df^2+6df(d-e+f)-7d(d-e+f)^2+46e^3+2\cdots}{10d^3+69d^2e+102d^2f+102de^2+318def+246df^2\cdots}\cdots\right.}{6} - \frac{(d+e+f)\log\left(x+\frac{26d^3+66d^2e+132d^2f+27d^2(d+e+f)+78de^2+276def+36de(d+e+f)+222df^2-18df(d+e+f)-63d(d+e+f)^2+4\cdots}{10d^3+69d^2e+102d^2f+102de^2+318def+\cdots}\cdots\right.}{2} + \frac{(d+2e+4f)\log\left(x+\frac{26d^3+66d^2e+132d^2f-18d^2(d+2e+4f)+78de^2+276def-24de(d+2e+4f)+222df^2+12df(d+2e+4f)-28d(d\cdots}{10d^3+69d^2e+102d^2f+102d\cdots}\cdots\right.}{3}$$

```
[In] integrate((2+x)*(f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] (d - e + f)*log(x + (26*d**3 + 66*d**2*e + 132*d**2*f - 9*d**2*(d - e + f)
+ 78*d*e**2 + 276*d*e*f - 12*d*e*(d - e + f) + 222*d*f**2 + 6*d*f*(d - e +
f) - 7*d*(d - e + f)**2 + 46*e**3 + 204*e**2*f + 3*e**2*(d - e + f) + 282*e
*f**2 + 36*e*f*(d - e + f) - 8*e*(d - e + f)**2 + 116*f**3 + 51*f**2*(d - e
 + f) - 13*f*(d - e + f)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f + 102*d*e**2
 + 318*d*e*f + 246*d*f**2 + 35*e**3 + 174*e**2*f + 285*e*f**2 + 154*f**3))/
6 - (d + e + f)*log(x + (26*d**3 + 66*d**2*e + 132*d**2*f + 27*d**2*(d + e
+ f) + 78*d*e**2 + 276*d*e*f + 36*d*e*(d + e + f) + 222*d*f**2 - 18*d*f*(d
+ e + f) - 63*d*(d + e + f)**2 + 46*e**3 + 204*e**2*f - 9*e**2*(d + e + f)
+ 282*e*f**2 - 108*e*f*(d + e + f) - 72*e*(d + e + f)**2 + 116*f**3 - 153*f
**2*(d + e + f) - 117*f*(d + e + f)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f +
 102*d*e**2 + 318*d*e*f + 246*d*f**2 + 35*e**3 + 174*e**2*f + 285*e*f**2 +
```

```
154*f**3))/2 + (d + 2*e + 4*f)*log(x + (26*d**3 + 66*d**2*e + 132*d**2*f -
18*d**2*(d + 2*e + 4*f) + 78*d*e**2 + 276*d*e*f - 24*d*e*(d + 2*e + 4*f) +
222*d*f**2 + 12*d*f*(d + 2*e + 4*f) - 28*d*(d + 2*e + 4*f)**2 + 46*e**3 + 2
04*e**2*f + 6*e**2*(d + 2*e + 4*f) + 282*e*f**2 + 72*e*f*(d + 2*e + 4*f) -
32*e*(d + 2*e + 4*f)**2 + 116*f**3 + 102*f**2*(d + 2*e + 4*f) - 52*f*(d + 2
*e + 4*f)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f + 102*d*e**2 + 318*d*e*f +
246*d*f**2 + 35*e**3 + 174*e**2*f + 285*e*f**2 + 154*f**3))/3
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 37, normalized size of antiderivative = 0.79

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}\,(d-e+f)\log\left(x+1\right)-\frac{1}{2}\,(d+e+f)\log\left(x-1\right) + \frac{1}{3}\,(d+2\,e+4\,f)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] 1/6*(d - e + f)*log(x + 1) - 1/2*(d + e + f)*log(x - 1) + 1/3*(d + 2*e + 4*
f)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.85

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}\,(d-e+f)\log\left(|x+1|\right)-\frac{1}{2}\,(d+e+f)\log\left(|x-1|\right) + \frac{1}{3}\,(d+2\,e+4\,f)\log\left(|x-2|\right)$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] 1/6*(d - e + f)*log(abs(x + 1)) - 1/2*(d + e + f)*log(abs(x - 1)) + 1/3*(d
+ 2*e + 4*f)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.07 (sec) , antiderivative size = 47, normalized size of antiderivative = 1.00

$$\int \frac{(2+x)\,(d+ex+fx^2)}{4-5x^2+x^4}\,dx = \ln(x-2)\,\left(\frac{d}{3}+\frac{2\,e}{3}+\frac{4\,f}{3}\right) - \ln(x-1)\,\left(\frac{d}{2}+\frac{e}{2}+\frac{f}{2}\right) + \ln(x+1)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)$$

```
[In] int(((x + 2)*(d + e*x + f*x^2))/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x - 2)*(d/3 + (2*e)/3 + (4*f)/3) - log(x - 1)*(d/2 + e/2 + f/2) + log(x
 + 1)*(d/6 - e/6 + f/6)
```

3.82 $\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 31, antiderivative size = 57

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4} dx = gx - \frac{1}{2}(d+e+f+g)\log(1-x) + \frac{1}{3}(d+2e+4f+8g)\log(2-x) + \frac{1}{6}(d-e+f-g)\log(1+x)$$

```
[Out] g*x-1/2*(d+e+f+g)*ln(1-x)+1/3*(d+2*e+4*f+8*g)*ln(2-x)+1/6*(d-e+f-g)*ln(1+x)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 57, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.065, Rules used = {1600, 2099}

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4} dx = -\frac{1}{2}\log(1-x)(d+e+f+g) + \frac{1}{3}\log(2-x)(d+2e+4f+8g) + \frac{1}{6}\log(x+1)(d-e+f-g) + gx$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4),x]
```

```
[Out] g*x - ((d + e + f + g)*Log[1 - x])/2 + ((d + 2*e + 4*f + 8*g)*Log[2 - x])/3
 + ((d - e + f - g)*Log[1 + x])/6
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3}{2-x-2x^2+x^3}\,dx \\
&= \int \left(g+\frac{d+2e+4f+8g}{3(-2+x)}+\frac{-d-e-f-g}{2(-1+x)}+\frac{d-e+f-g}{6(1+x)}\right)dx \\
&= gx-\frac{1}{2}(d+e+f+g)\log(1-x)+\frac{1}{3}(d+2e+4f+8g)\log(2-x)+\frac{1}{6}(d-e+f \\
&\quad -g)\log(1+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 55, normalized size of antiderivative = 0.96

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}(6gx-3(d+e+f+g)\log(1-x) + 2(d+2e+4f+8g)\log(2-x) + (d-e+f-g)\log(1+x))$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4),x]
[Out] (6*g*x - 3*(d + e + f + g)*Log[1 - x] + 2*(d + 2*e + 4*f + 8*g)*Log[2 - x]
+ (d - e + f - g)*Log[1 + x])/6
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 59, normalized size of antiderivative = 1.04

method	result
default	$gx+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}+\frac{8g}{3}\right)\ln\left(x-\right.$
norman	$gx+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}+\frac{8g}{3}\right)\ln\left(x-\right.$
parallelrisch	$gx+\frac{\ln(x-2)d}{3}+\frac{2\ln(x-2)e}{3}+\frac{4\ln(x-2)f}{3}+\frac{8\ln(x-2)g}{3}-\frac{\ln(x-1)d}{2}-\frac{\ln(x-1)e}{2}-\frac{\ln(x-1)f}{2}-\frac{\ln(x-1)g}{2}+\ln($
risch	$gx+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}-\frac{\ln(x+1)g}{6}-\frac{\ln(1-x)d}{2}-\frac{\ln(1-x)e}{2}-\frac{\ln(1-x)f}{2}-\frac{\ln(1-x)g}{2}+\frac{\ln(2-x}{3}$

```
[In] int((x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
```

```
[Out] g*x+(1/6*d-1/6*e+1/6*f-1/6*g)*ln(x+1)+(-1/2*d-1/2*e-1/2*f-1/2*g)*ln(x-1)+(1
/3*d+2/3*e+4/3*f+8/3*g)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 47, normalized size of antiderivative = 0.82

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = gx+\frac{1}{6}\left(d-e+f-g\right)\log\left(x+1\right) - \frac{1}{2}\left(d+e+f+g\right)\log\left(x-1\right) + \frac{1}{3}\left(d+2\,e+4\,f+8\,g\right)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="fricas")
```

```
[Out] g*x + 1/6*(d - e + f - g)*log(x + 1) - 1/2*(d + e + f + g)*log(x - 1) + 1/3
*(d + 2*e + 4*f + 8*g)*log(x - 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1389 vs. 2(63) = 126.

Time = 54.94 (sec) , antiderivative size = 1389, normalized size of antiderivative = 24.37

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = \text{Too large to display}$$

```
[In] integrate((2+x)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] g*x + (d - e + f - g)*log(x + (26*d**3 + 66*d**2*e + 132*d**2*f + 174*d**2*
g - 9*d**2*(d - e + f - g) + 78*d*e**2 + 276*d*e*f + 444*d*e*g - 12*d*e*(d
- e + f - g) + 222*d*f**2 + 636*d*f*g + 6*d*f*(d - e + f - g) + 510*d*g**2
+ 36*d*g*(d - e + f - g) - 7*d*(d - e + f - g)**2 + 46*e**3 + 204*e**2*f +
```

```
390*e**2*g + 3*e**2*(d - e + f - g) + 282*e*f**2 + 984*e*f*g + 36*e*f*(d -
e + f - g) + 930*e*g**2 + 102*e*g*(d - e + f - g) - 8*e*(d - e + f - g)**2
+ 116*f**3 + 534*f**2*g + 51*f**2*(d - e + f - g) + 924*f*g**2 + 228*f*g*(d
 - e + f - g) - 13*f*(d - e + f - g)**2 + 586*g**3 + 243*g**2*(d - e + f -
g) - 20*g*(d - e + f - g)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f + 213*d**2*
g + 102*d*e**2 + 318*d*e*f + 564*d*e*g + 246*d*f**2 + 894*d*f*g + 750*d*g**
2 + 35*e**3 + 174*e**2*f + 249*e**2*g + 285*e*f**2 + 852*e*f*g + 537*e*g**2
 + 154*f**3 + 717*f**2*g + 966*f*g**2 + 323*g**3))/6 - (d + e + f + g)*log(
x + (26*d**3 + 66*d**2*e + 132*d**2*f + 174*d**2*g + 27*d**2*(d + e + f + g
) + 78*d*e**2 + 276*d*e*f + 444*d*e*g + 36*d*e*(d + e + f + g) + 222*d*f**2
 + 636*d*f*g - 18*d*f*(d + e + f + g) + 510*d*g**2 - 108*d*g*(d + e + f + g
) - 63*d*(d + e + f + g)**2 + 46*e**3 + 204*e**2*f + 390*e**2*g - 9*e**2*(d
 + e + f + g) + 282*e*f**2 + 984*e*f*g - 108*e*f*(d + e + f + g) + 930*e*g*
*2 - 306*e*g*(d + e + f + g) - 72*e*(d + e + f + g)**2 + 116*f**3 + 534*f**
2*g - 153*f**2*(d + e + f + g) + 924*f*g**2 - 684*f*g*(d + e + f + g) - 117
*f*(d + e + f + g)**2 + 586*g**3 - 729*g**2*(d + e + f + g) - 180*g*(d + e
+ f + g)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f + 213*d**2*g + 102*d*e**2 +
318*d*e*f + 564*d*e*g + 246*d*f**2 + 894*d*f*g + 750*d*g**2 + 35*e**3 + 174
*e**2*f + 249*e**2*g + 285*e*f**2 + 852*e*f*g + 537*e*g**2 + 154*f**3 + 717
*f**2*g + 966*f*g**2 + 323*g**3))/2 + (d + 2*e + 4*f + 8*g)*log(x + (26*d**
3 + 66*d**2*e + 132*d**2*f + 174*d**2*g - 18*d**2*(d + 2*e + 4*f + 8*g) + 7
8*d*e**2 + 276*d*e*f + 444*d*e*g - 24*d*e*(d + 2*e + 4*f + 8*g) + 222*d*f**
2 + 636*d*f*g + 12*d*f*(d + 2*e + 4*f + 8*g) + 510*d*g**2 + 72*d*g*(d + 2*e
 + 4*f + 8*g) - 28*d*(d + 2*e + 4*f + 8*g)**2 + 46*e**3 + 204*e**2*f + 390*
e**2*g + 6*e**2*(d + 2*e + 4*f + 8*g) + 282*e*f**2 + 984*e*f*g + 72*e*f*(d
+ 2*e + 4*f + 8*g) + 930*e*g**2 + 204*e*g*(d + 2*e + 4*f + 8*g) - 32*e*(d +
 2*e + 4*f + 8*g)**2 + 116*f**3 + 534*f**2*g + 102*f**2*(d + 2*e + 4*f + 8*
g) + 924*f*g**2 + 456*f*g*(d + 2*e + 4*f + 8*g) - 52*f*(d + 2*e + 4*f + 8*g
)**2 + 586*g**3 + 486*g**2*(d + 2*e + 4*f + 8*g) - 80*g*(d + 2*e + 4*f + 8*
g)**2)/(10*d**3 + 69*d**2*e + 102*d**2*f + 213*d**2*g + 102*d*e**2 + 318*d*
e*f + 564*d*e*g + 246*d*f**2 + 894*d*f*g + 750*d*g**2 + 35*e**3 + 174*e**2*
f + 249*e**2*g + 285*e*f**2 + 852*e*f*g + 537*e*g**2 + 154*f**3 + 717*f**2*
g + 966*f*g**2 + 323*g**3))/3
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 47, normalized size of antiderivative = 0.82

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = gx+\frac{1}{6}\left(d-e+f-g\right)\log\left(x+1\right) - \frac{1}{2}\left(d+e+f+g\right)\log\left(x-1\right) + \frac{1}{3}\left(d+2\,e+4\,f+8\,g\right)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxima")
```

```
[Out] g*x + 1/6*(d - e + f - g)*log(x + 1) - 1/2*(d + e + f + g)*log(x - 1) + 1/3
*(d + 2*e + 4*f + 8*g)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 50, normalized size of antiderivative = 0.88

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = gx+\frac{1}{6}\left(d-e+f-g\right)\log\left(|x+1|\right) - \frac{1}{2}\left(d+e+f+g\right)\log\left(|x-1|\right) + \frac{1}{3}\left(d+2\,e+4\,f+8\,g\right)\log\left(|x-2|\right)$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac")
```

```
[Out] g*x + 1/6*(d - e + f - g)*log(abs(x + 1)) - 1/2*(d + e + f + g)*log(abs(x -
 1)) + 1/3*(d + 2*e + 4*f + 8*g)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.87 (sec) , antiderivative size = 59, normalized size of antiderivative = 1.04

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{4-5x^2+x^4}\,dx = \ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right) - \ln\left(x-1\right)\,\left(\frac{d}{2}+\frac{e}{2}+\frac{f}{2}+\frac{g}{2}\right) + \ln\left(x-2\right)\,\left(\frac{d}{3}+\frac{2\,e}{3}+\frac{4\,f}{3}+\frac{8\,g}{3}\right)+g\,x$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3))/(x^4 - 5*x^2 + 4),x)
```

```
[Out] log(x + 1)*(d/6 - e/6 + f/6 - g/6) - log(x - 1)*(d/2 + e/2 + f/2 + g/2) + l
og(x - 2)*(d/3 + (2*e)/3 + (4*f)/3 + (8*g)/3) + g*x
```

3.83 $\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 74

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx = (g+2h)x+\frac{hx^2}{2}-\frac{1}{2}(d+e+f+g+h)\log(1-x) +\frac{1}{3}(d+2e+4f+8g+16h)\log(2-x) +\frac{1}{6}(d-e+f-g+h)\log(1+x)$$

```
[Out] (g+2*h)*x+1/2*h*x^2-1/2*(d+e+f+g+h)*ln(1-x)+1/3*(d+2*e+4*f+8*g+16*h)*ln(2-x
)+1/6*(d-e+f-g+h)*ln(1+x)
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 74, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.056, Rules used = {1600, 2099}

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx = -\frac{1}{2}\log(1-x)(d+e+f+g+h) +\frac{1}{3}\log(2-x)(d+2e+4f+8g+16h) +\frac{1}{6}\log(x+1)(d-e+f-g+h)+x(g+2h)+\frac{hx^2}{2}$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4),x]
```

```
[Out] (g + 2*h)*x + (h*x^2)/2 - ((d + e + f + g + h)*Log[1 - x])/2 + ((d + 2*e +
4*f + 8*g + 16*h)*Log[2 - x])/3 + ((d - e + f - g + h)*Log[1 + x])/6
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4}{2-x-2x^2+x^3}\,dx \\
&= \int \left(g\left(1+\frac{2h}{g}\right)+\frac{d+2e+4f+8g+16h}{3(-2+x)}+\frac{-d-e-f-g-h}{2(-1+x)}+hx+\frac{d-e+f-g+h}{6(1+x)}\right)dx \\
&= (g+2h)x+\frac{hx^2}{2}-\frac{1}{2}(d+e+f+g+h)\log(1-x) \\
&\quad +\frac{1}{3}(d+2e+4f+8g+16h)\log(2-x)+\frac{1}{6}(d-e+f-g+h)\log(1+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 71, normalized size of antiderivative = 0.96

$$
\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx = \frac{1}{6}\big(6(g+2h)x+3hx^2-3(d+e+f+g+h)\log(1-x)+2(d+2(e+2f+4g+8h))\log(2-x)+(d-e+f-g+h)\log(1+x)\big)
$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4),x]
[Out] (6*(g + 2*h)*x + 3*h*x^2 - 3*(d + e + f + g + h)*Log[1 - x] + 2*(d + 2*(e +
 2*f + 4*g + 8*h))*Log[2 - x] + (d - e + f - g + h)*Log[1 + x])/6
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 78, normalized size of antiderivative = 1.05

method	result
default	$\frac{h\,x^{2}}{2}+gx+2hx+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}-\frac{h}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}\right.$
norman	$\left(g+2h\right)x+\frac{h\,x^{2}}{2}+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}-\frac{h}{2}\right)\ln\left(x-1\right)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}+\frac{8g}{3}+\frac{16h}{3}\right)\ln\left(x-2\right)$
parallelrisch	$\frac{h\,x^{2}}{2}+gx+2hx+\frac{\ln(x-2)d}{3}+\frac{2\ln(x-2)e}{3}+\frac{4\ln(x-2)f}{3}+\frac{8\ln(x-2)g}{3}+\frac{16\ln(x-2)h}{3}-\frac{\ln(x-1)d}{2}-\frac{\ln(x-1)}{2}$
risch	$\frac{h\,x^{2}}{2}+gx+2hx+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}-\frac{\ln(x+1)g}{6}+\frac{\ln(x+1)h}{6}+\frac{\ln(2-x)d}{3}+\frac{2\ln(2-x)e}{3}+$

```
[In] int((x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVERBOSE)
```

```
[Out] 1/2*h*x^2+g*x+2*h*x+(1/6*d-1/6*e+1/6*f-1/6*g+1/6*h)*ln(x+1)+(-1/2*d-1/2*e-1
/2*f-1/2*g-1/2*h)*ln(x-1)+(1/3*d+2/3*e+4/3*f+8/3*g+16/3*h)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.84

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx=\frac{1}{2}\,hx^2+(g+2\,h)x+\frac{1}{6}\,(d-e+f-g+h)\log\left(x+1\right)-\frac{1}{2}\,(d+e+f+g+h)\log\left(x-1\right)+\frac{1}{3}\,(d+2\,e+4\,f+8\,g+16\,h)\log\left(x-2\right)$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="frica
s")
```

```
[Out] 1/2*h*x^2 + (g + 2*h)*x + 1/6*(d - e + f - g + h)*log(x + 1) - 1/2*(d + e +
 f + g + h)*log(x - 1) + 1/3*(d + 2*e + 4*f + 8*g + 16*h)*log(x - 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((2+x)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.84

$$\begin{aligned}\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx &= \frac{1}{2}hx^2+(g+2h)x \\ &\quad +\frac{1}{6}(d-e+f-g+h)\log(x+1) \\ &\quad -\frac{1}{2}(d+e+f+g+h)\log(x-1) \\ &\quad +\frac{1}{3}(d+2e+4f+8g+16h)\log(x-2)\end{aligned}$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="maxim
a")
```

```
[Out] 1/2*h*x^2 + (g + 2*h)*x + 1/6*(d - e + f - g + h)*log(x + 1) - 1/2*(d + e +
 f + g + h)*log(x - 1) + 1/3*(d + 2*e + 4*f + 8*g + 16*h)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 65, normalized size of antiderivative = 0.88

$$\begin{aligned}\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{4-5x^2+x^4}\,dx &= \frac{1}{2}hx^2+gx+2hx \\ &\quad +\frac{1}{6}(d-e+f-g+h)\log(|x+1|) \\ &\quad -\frac{1}{2}(d+e+f+g+h)\log(|x-1|) \\ &\quad +\frac{1}{3}(d+2e+4f+8g+16h)\log(|x-2|)\end{aligned}$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm="giac"
)
```

```
[Out] 1/2*h*x^2 + g*x + 2*h*x + 1/6*(d - e + f - g + h)*log(abs(x + 1)) - 1/2*(d
+ e + f + g + h)*log(abs(x - 1)) + 1/3*(d + 2*e + 4*f + 8*g + 16*h)*log(abs
(x - 2))
```

Mupad [B] (verification not implemented)

Time = 8.03 (sec) , antiderivative size = 78, normalized size of antiderivative = 1.05

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4)}{4-5x^2+x^4}\,dx = x\,(g+2\,h)+\frac{h\,x^2}{2} - \ln(x-1)\,\left(\frac{d}{2}+\frac{e}{2}+\frac{f}{2}+\frac{g}{2}+\frac{h}{2}\right) + \ln(x+1)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right) + \ln(x-2)\,\left(\frac{d}{3}+\frac{2\,e}{3}+\frac{4\,f}{3}+\frac{8\,g}{3}+\frac{16\,h}{3}\right)$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2 + 4),x)
```

```
[Out] x*(g + 2*h) + (h*x^2)/2 - log(x - 1)*(d/2 + e/2 + f/2 + g/2 + h/2) + log(x
+ 1)*(d/6 - e/6 + f/6 - g/6 + h/6) + log(x - 2)*(d/3 + (2*e)/3 + (4*f)/3 +
(8*g)/3 + (16*h)/3)
```

3.84 $\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx$

Optimal result

Integrand size = 41, antiderivative size = 96

$$\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx = (g+2h+5i)x+\frac{1}{2}(h+2i)x^2+\frac{ix^3}{3}-\frac{1}{2}(d+e+f+g+h+i)\log(1-x) + \frac{1}{3}(d+2e+4f+8g+16h+32i)\log(2-x)+\frac{1}{6}(d-e+f-g+h-i)\log(1+x)$$

```
[Out] (g+2*h+5*i)*x+1/2*(h+2*i)*x^2+1/3*i*x^3-1/2*(d+e+f+g+h+i)*ln(1-x)+1/3*(d+2*
e+4*f+8*g+16*h+32*i)*ln(2-x)+1/6*(d-e+f-g+h-i)*ln(1+x)
```

Rubi [A] (verified)

Time = 0.09 (sec) , antiderivative size = 96, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.049, Rules used = {1600, 2099}

$$\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4} dx = -\frac{1}{2}\log(1-x)(d+e+f+g+h+i)+\frac{1}{3}\log(2-x)(d+2e+4f+8g+16h+32i) + \frac{1}{6}\log(x+1)(d-e+f-g+h-i)+x(g+2h+5i)+\frac{1}{2}x^2(h+2i)+\frac{ix^3}{3}$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2 + x^4),x
]
```

```
[Out] (g + 2*h + 5*i)*x + ((h + 2*i)*x^2)/2 + (i*x^3)/3 - ((d + e + f + g + h + i
)*Log[1 - x])/2 + ((d + 2*e + 4*f + 8*g + 16*h + 32*i)*Log[2 - x])/3 + ((d
- e + f - g + h - i)*Log[1 + x])/6
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{2-x-2x^2+x^3}\,dx\\ &= \int \left(g\left(1+\frac{2h+5i}{g}\right)+\frac{d+2e+4f+8g+16h+32i}{3(-2+x)}+\frac{-d-e-f-g-h-i}{2(-1+x)}\right.\\ &\qquad\left.+(h+2i)x+ix^2+\frac{d-e+f-g+h-i}{6(1+x)}\right)dx\\ &= (g+2h+5i)x+\frac{1}{2}(h+2i)x^2+\frac{ix^3}{3}-\frac{1}{2}(d+e+f+g+h+i)\log(1-x)\\ &\quad+\frac{1}{3}(d+2e+4f+8g+16h+32i)\log(2-x)+\frac{1}{6}(d-e+f-g+h-i)\log(1+x)\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 91, normalized size of antiderivative = 0.95

$$\begin{aligned}&\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{4-5x^2+x^4}\,dx\\ &=\frac{1}{6}\big(6(g+2h+5i)x+3(h+2i)x^2+2ix^3-3(d+e+f+g+h+i)\log(1-x)\\ &\qquad+2(d+2e+4(f+2g+4h+8i))\log(2-x)+(d-e+f-g+h-i)\log(1+x)\big)\end{aligned}$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2 +
x^4),x]
```

```
[Out] (6*(g + 2*h + 5*i)*x + 3*(h + 2*i)*x^2 + 2*i*x^3 - 3*(d + e + f + g + h + i
)*Log[1 - x] + 2*(d + 2*e + 4*(f + 2*g + 4*h + 8*i))*Log[2 - x] + (d - e +
f - g + h - i)*Log[1 + x])/6
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 99, normalized size of antiderivative = 1.03

method	result
norman	$\left(\frac{h}{2}+i\right)x^2+(g+2h+5i)x+\frac{i x^3}{3}+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}-\frac{h}{2}-\frac{i}{2}\right)\ln(x-1)+\left(\frac{d}{3}+\frac{2e}{3}+\frac{4f}{3}+\right.$
default	$\frac{i x^3}{3}+\frac{h x^2}{2}+i x^2+gx+2hx+5ix+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right)\ln(x+1)+\left(-\frac{d}{2}-\frac{e}{2}-\frac{f}{2}-\frac{g}{2}\right.$
parallelrisch	$gx+i x^2+\frac{\ln(x-2)d}{3}+\frac{2\ln(x-2)e}{3}-\frac{\ln(x-1)d}{2}-\frac{\ln(x-1)e}{2}+\frac{h x^2}{2}+\frac{i x^3}{3}-\frac{\ln(x+1)i}{6}+\frac{\ln(x+1)f}{6}+2hx+$
risch	$\frac{i x^3}{3}+\frac{h x^2}{2}+i x^2+gx+2hx+5ix+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}-\frac{\ln(x+1)g}{6}+\frac{\ln(x+1)h}{6}-\frac{\ln(x+}{6}$

```
[In] int((x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x,method=_RETURNVER
BOSE)
```

```
[Out] (1/2*h+i)*x^2+(g+2*h+5*i)*x+1/3*i*x^3+(-1/2*d-1/2*e-1/2*f-1/2*g-1/2*h-1/2*i
)*ln(x-1)+(1/3*d+2/3*e+4/3*f+8/3*g+16/3*h+32/3*i)*ln(x-2)+(1/6*d-1/6*e+1/6*
f-1/6*g+1/6*h-1/6*i)*ln(x+1)
```

Fricas [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.85

$$\begin{aligned}&\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{4-5x^2+x^4}\,dx\\&=\frac{1}{3}\,ix^3+\frac{1}{2}\,(h+2\,i)x^2+(g+2\,h+5\,i)x+\frac{1}{6}\,(d-e+f-g+h-i)\log(x+1)\\&\quad-\frac{1}{2}\,(d+e+f+g+h+i)\log(x-1)+\frac{1}{3}\,(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log(x-2)\end{aligned}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"fricas")
```

```
[Out] 1/3*i*x^3 + 1/2*(h + 2*i)*x^2 + (g + 2*h + 5*i)*x + 1/6*(d - e + f - g + h
- i)*log(x + 1) - 1/2*(d + e + f + g + h + i)*log(x - 1) + 1/3*(d + 2*e + 4
*f + 8*g + 16*h + 32*i)*log(x - 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx = \text{Timed out}$$

```
[In] integrate((2+x)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4),x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 82, normalized size of antiderivative = 0.85

$$\begin{aligned}
&\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx \\
&= \frac{1}{3}\,ix^3+\frac{1}{2}\,(h+2\,i)x^2+(g+2\,h+5\,i)x+\frac{1}{6}\,(d-e+f-g+h-i)\log{(x+1)} \\
&\quad -\frac{1}{2}\,(d+e+f+g+h+i)\log{(x-1)}+\frac{1}{3}\,(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log{(x-2)}
\end{aligned}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"maxima")
[Out] 1/3*i*x^3 + 1/2*(h + 2*i)*x^2 + (g + 2*h + 5*i)*x + 1/6*(d - e + f - g + h
- i)*log(x + 1) - 1/2*(d + e + f + g + h + i)*log(x - 1) + 1/3*(d + 2*e + 4
*f + 8*g + 16*h + 32*i)*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 87, normalized size of antiderivative = 0.91

$$\begin{aligned}
&\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{4-5x^2+x^4}\,dx \\
&= \frac{1}{3}\,ix^3+\frac{1}{2}\,hx^2+ix^2+gx+2\,hx+5\,ix+\frac{1}{6}\,(d-e+f-g+h-i)\log{(|x+1|)} \\
&\quad -\frac{1}{2}\,(d+e+f+g+h+i)\log{(|x-1|)} \\
&\quad +\frac{1}{3}\,(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log{(|x-2|)}
\end{aligned}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4),x, algorithm=
"giac")
```

```
[Out] 1/3*i*x^3 + 1/2*h*x^2 + i*x^2 + g*x + 2*h*x + 5*i*x + 1/6*(d - e + f - g +
h - i)*log(abs(x + 1)) - 1/2*(d + e + f + g + h + i)*log(abs(x - 1)) + 1/3*
(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.93 (sec) , antiderivative size = 99, normalized size of antiderivative = 1.03

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{4-5x^2+x^4}\,dx$$
$$= x\,(g+2\,h+5\,i)+\frac{i\,x^3}{3}-\ln\left(x-1\right)\,\left(\frac{d}{2}+\frac{e}{2}+\frac{f}{2}+\frac{g}{2}+\frac{h}{2}+\frac{i}{2}\right)$$
$$+\ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right)$$
$$+\ln\left(x-2\right)\,\left(\frac{d}{3}+\frac{2\,e}{3}+\frac{4\,f}{3}+\frac{8\,g}{3}+\frac{16\,h}{3}+\frac{32\,i}{3}\right)+x^2\,\left(\frac{h}{2}+i\right)$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4 - 5*x^2 + 4),x
)
```

```
[Out] x*(g + 2*h + 5*i) + (i*x^3)/3 - log(x - 1)*(d/2 + e/2 + f/2 + g/2 + h/2 + i
/2) + log(x + 1)*(d/6 - e/6 + f/6 - g/6 + h/6 - i/6) + log(x - 2)*(d/3 + (2
*e)/3 + (4*f)/3 + (8*g)/3 + (16*h)/3 + (32*i)/3) + x^2*(h/2 + i)
```

3.85 $\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 26, antiderivative size = 46

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12(2+x)} - \frac{1}{18}\log(1-x) + \frac{1}{48}\log(2-x) + \frac{1}{6}\log(1+x) - \frac{19}{144}\log(2+x)$$

```
[Out] 1/12/(2+x)-1/18*ln(1-x)+1/48*ln(2-x)+1/6*ln(1+x)-19/144*ln(2+x)
```

Rubi [A] (verified)

Time = 0.03 (sec) , antiderivative size = 46, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.077, Rules used = {1600, 2099}

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12(x+2)} - \frac{1}{18}\log(1-x) + \frac{1}{48}\log(2-x) + \frac{1}{6}\log(x+1) - \frac{19}{144}\log(x+2)$$

```
[In] Int[(2 - x - 2*x^2 + x^3)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] 1/(12*(2 + x)) - Log[1 - x]/18 + Log[2 - x]/48 + Log[1 + x]/6 - (19*Log[2 +
 x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{1}{(2+x)^2\,(2-x-2x^2+x^3)}\,dx \\
&= \int \left(\frac{1}{48(-2+x)} - \frac{1}{18(-1+x)} + \frac{1}{6(1+x)} - \frac{1}{12(2+x)^2} - \frac{19}{144(2+x)}\right) dx \\
&= \frac{1}{12(2+x)} - \frac{1}{18}\log(1-x) + \frac{1}{48}\log(2-x) + \frac{1}{6}\log(1+x) - \frac{19}{144}\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 42, normalized size of antiderivative = 0.91

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(\frac{12}{2+x} + 24\log(-1-x) - 8\log(1-x) + 3\log(2-x) - 19\log(2+x)\right)$$

```
[In] Integrate[(2 - x - 2*x^2 + x^3)/(4 - 5*x^2 + x^4)^2,x]
[Out] (12/(2 + x) + 24*Log[-1 - x] - 8*Log[1 - x] + 3*Log[2 - x] - 19*Log[2 + x])
/144
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 33, normalized size of antiderivative = 0.72

method	result	size
default	$\frac{1}{12x+24}-\frac{19\ln(x+2)}{144}+\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{18}+\frac{\ln(x-2)}{48}$	33
risch	$\frac{1}{12x+24}-\frac{19\ln(x+2)}{144}+\frac{\ln(x+1)}{6}-\frac{\ln(x-1)}{18}+\frac{\ln(x-2)}{48}$	33
norman	$\frac{-\frac{1}{6}x^2-\frac{1}{12}x+\frac{1}{12}x^3+\frac{1}{6}}{x^4-5x^2+4}+\frac{\ln(x-2)}{48}-\frac{\ln(x-1)}{18}+\frac{\ln(x+1)}{6}-\frac{19\ln(x+2)}{144}$	54
parallelrisch	$\frac{3\ln(x-2)x-8\ln(x-1)x+24\ln(x+1)x-19\ln(x+2)x+12+6\ln(x-2)-16\ln(x-1)+48\ln(x+1)-38\ln(x+2)}{144x+288}$	62

```
[In] int((x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] 1/12/(x+2)-19/144*ln(x+2)+1/6*ln(x+1)-1/18*ln(x-1)+1/48*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 45, normalized size of antiderivative = 0.98

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = -\frac{19\,(x+2)\log\left(x+2\right)-24\,(x+2)\log\left(x+1\right)+8\,(x+2)\log\left(x-1\right)-3\,(x+2)\log\left(x-2\right)-12}{144\,(x+2)}$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
[Out] -1/144*(19*(x + 2)*log(x + 2) - 24*(x + 2)*log(x + 1) + 8*(x + 2)*log(x - 1
) - 3*(x + 2)*log(x - 2) - 12)/(x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.14 (sec) , antiderivative size = 34, normalized size of antiderivative = 0.74

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{\log\left(x-2\right)}{48}-\frac{\log\left(x-1\right)}{18}+\frac{\log\left(x+1\right)}{6}-\frac{19\log\left(x+2\right)}{144}+\frac{1}{12x+24}$$

```
[In] integrate((x**3-2*x**2-x+2)/(x**4-5*x**2+4)**2,x)
[Out] log(x - 2)/48 - log(x - 1)/18 + log(x + 1)/6 - 19*log(x + 2)/144 + 1/(12*x
+ 24)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 32, normalized size of antiderivative = 0.70

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12\,(x+2)}-\frac{19}{144}\log\left(x+2\right)+\frac{1}{6}\log\left(x+1\right)-\frac{1}{18}\log\left(x-1\right)+\frac{1}{48}\log\left(x-2\right)$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
[Out] 1/12/(x + 2) - 19/144*log(x + 2) + 1/6*log(x + 1) - 1/18*log(x - 1) + 1/48*
log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 36, normalized size of antiderivative = 0.78

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12\,(x+2)} - \frac{19}{144}\,\log\left(|x+2|\right) + \frac{1}{6}\,\log\left(|x+1|\right) - \frac{1}{18}\,\log\left(|x-1|\right) + \frac{1}{48}\,\log\left(|x-2|\right)$$

```
[In] integrate((x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/12/(x + 2) - 19/144*log(abs(x + 2)) + 1/6*log(abs(x + 1)) - 1/18*log(abs(
x - 1)) + 1/48*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 32, normalized size of antiderivative = 0.70

$$\int \frac{2-x-2x^2+x^3}{(4-5x^2+x^4)^2}\,dx = \frac{\ln\left(x+1\right)}{6} - \frac{\ln\left(x-1\right)}{18} + \frac{\ln\left(x-2\right)}{48} - \frac{19\,\ln\left(x+2\right)}{144} + \frac{1}{12\,\left(x+2\right)}$$

```
[In] int(-(x + 2*x^2 - x^3 - 2)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x + 1)/6 - log(x - 1)/18 + log(x - 2)/48 - (19*log(x + 2))/144 + 1/(12*
(x + 2))
```

3.86 $\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 31, antiderivative size = 71

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e}{12(2+x)} - \frac{1}{18}(d+e)\log(1-x) + \frac{1}{48}(d+2e)\log(2-x) + \frac{1}{6}(d-e)\log(1+x) - \frac{1}{144}(19d-26e)\log(2+x)$$

```
[Out] 1/12*(d-2*e)/(2+x)-1/18*(d+e)*ln(1-x)+1/48*(d+2*e)*ln(2-x)+1/6*(d-e)*ln(1+x
)-1/144*(19*d-26*e)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.12 (sec) , antiderivative size = 71, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.065, Rules used = {1600, 6874}

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e}{12(x+2)} - \frac{1}{18}(d+e)\log(1-x) + \frac{1}{48}(d+2e)\log(2-x) + \frac{1}{6}(d-e)\log(x+1) - \frac{1}{144}(19d-26e)\log(x+2)$$

```
[In] Int[((d + e*x)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d - 2*e)/(12*(2 + x)) - ((d + e)*Log[1 - x])/18 + ((d + 2*e)*Log[2 - x])/4
8 + ((d - e)*Log[1 + x])/6 - ((19*d - 26*e)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
```

```
EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex}{(2+x)^2\,(2-x-2x^2+x^3)}\,dx \\
&= \int \left(\frac{d+2e}{48(-2+x)}+\frac{-d-e}{18(-1+x)}+\frac{d-e}{6(1+x)}+\frac{-d+2e}{12(2+x)^2}+\frac{-19d+26e}{144(2+x)}\right)dx \\
&= \frac{d-2e}{12(2+x)}-\frac{1}{18}(d+e)\log(1-x)+\frac{1}{48}(d+2e)\log(2-x) \\
&\quad +\frac{1}{6}(d-e)\log(1+x)-\frac{1}{144}(19d-26e)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.03 (sec) , antiderivative size = 66, normalized size of antiderivative = 0.93

$$\int \frac{(d+ex)\,(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(\frac{12(d-2e)}{2+x}+24(d-e)\log(-1-x) -8(d+e)\log(1-x)+3(d+2e)\log(2-x) +(-19d+26e)\log(2+x)\right)$$

```
[In] Integrate[((d + e*x)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(d - 2*e))/(2 + x) + 24*(d - e)*Log[-1 - x] - 8*(d + e)*Log[1 - x] + 3
*(d + 2*e)*Log[2 - x] + (-19*d + 26*e)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 64, normalized size of antiderivative = 0.90

method	result
default	$\left(-\frac{19d}{144}+\frac{13e}{72}\right)\ln\left(x+2\right)-\frac{-\frac{d}{12}+\frac{e}{6}}{x+2}+\left(\frac{d}{6}-\frac{e}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{18}-\frac{e}{18}\right)\ln\left(x-1\right)+\left(\frac{d}{48}+\frac{e}{24}\right)\mathrm{l}$
risch	$\frac{d}{12x+24}-\frac{e}{6(x+2)}-\frac{\ln(x-1)d}{18}-\frac{\ln(x-1)e}{18}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}-\frac{19\ln(-x-2)d}{144}+\frac{13\ln(-x-2)e}{72}+\frac{\ln(2-x)}{48}$
norman	$\frac{\left(-\frac{d}{12}+\frac{e}{6}\right)x+\left(\frac{d}{12}-\frac{e}{6}\right)x^3+\left(-\frac{d}{6}+\frac{e}{3}\right)x^2+\frac{d}{6}-\frac{e}{3}}{x^4-5x^2+4}+\left(-\frac{19d}{144}+\frac{13e}{72}\right)\ln\left(x+2\right)+\left(-\frac{d}{18}-\frac{e}{18}\right)\ln\left(x-1\right)+\left(\frac{d}{6}\right.$
parallelrisch	$\frac{3\ln(x-2)xd+6\ln(x-2)xe-8\ln(x-1)xd-8\ln(x-1)xe+24\ln(x+1)xd-24\ln(x+1)xe-19\ln(x+2)xd+26\ln(x+2)xe+6\ln(x-2}{144x+288}$

[In] int((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)

[Out] (-19/144*d+13/72*e)*ln(x+2)-(-1/12*d+1/6*e)/(x+2)+(1/6*d-1/6*e)*ln(x+1)+(-1/18*d-1/18*e)*ln(x-1)+(1/48*d+1/24*e)*ln(x-2)

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 93, normalized size of antiderivative = 1.31

$$\int \frac{(d+ex)\left(2-x-2x^2+x^3\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{\left((19\,d-26\,e)x+38\,d-52\,e\right)\log\left(x+2\right)-24\left((d-e)x+2\,d-2\,e\right)\log\left(x+1\right)+8\left((d+e)x+2\,d+2\right.}{144\left(x+2\right)}$$

[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="fricas")

[Out] -1/144*(((19*d - 26*e)*x + 38*d - 52*e)*log(x + 2) - 24*((d - e)*x + 2*d - 2*e)*log(x + 1) + 8*((d + e)*x + 2*d + 2*e)*log(x - 1) - 3*((d + 2*e)*x + 2*d + 4*e)*log(x - 2) - 12*d + 24*e)/(x + 2)

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1188 vs. 2(60) = 120.

Time = 7.10 (sec) , antiderivative size = 1188, normalized size of antiderivative = 16.73

$$\int \frac{(d+ex)\left(2-x-2x^2+x^3\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \text{Too large to display}$$

[In] integrate((e*x+d)*(x**3-2*x**2-x+2)/(x**4-5*x**2+4)**2,x)

[Out] (d - 2*e)/(12*x + 24) + (d - e)*log(x + (-1534775*d**6 + 8032360*d**5*e - 984027*d**5*(d - e) - 12991180*d**4*e**2 + 11797266*d**4*e*(d - e) + 3567168*d**4*(d - e)**2 + 1075200*d**3*e**3 - 32721528*d**3*e**2*(d - e) - 8725248*d**3*e*(d - e)**2 - 247104*d**3*(d - e)**3 + 16959280*d**2*e**4 + 38977296

```
*d**2*e**3*(d - e) - 2820096*d**2*e**2*(d - e)**2 - 10357632*d**2*e*(d - e)
**3 - 15836800*d*e**5 - 21294960*d*e**4*(d - e) + 15436800*d*e**3*(d - e)**
2 + 16277760*d*e**2*(d - e)**3 + 4283840*e**6 + 3876000*e**5*(d - e) - 6865
920*e**4*(d - e)**2 - 4078080*e**3*(d - e)**3)/(801262*d**6 - 4662251*d**5*
e + 7296938*d**4*e**2 + 1388616*d**3*e**3 - 12447440*d**2*e**4 + 9990800*d*
e**5 - 2380000*e**6))/6 - (d + e)*log(x + (-1534775*d**6 + 8032360*d**5*e +
 328009*d**5*(d + e) - 12991180*d**4*e**2 - 3932422*d**4*e*(d + e) + 396352
*d**4*(d + e)**2 + 1075200*d**3*e**3 + 10907176*d**3*e**2*(d + e) - 969472*
d**3*e*(d + e)**2 + 9152*d**3*(d + e)**3 + 16959280*d**2*e**4 - 12992432*d*
*2*e**3*(d + e) - 313344*d**2*e**2*(d + e)**2 + 383616*d**2*e*(d + e)**3 -
15836800*d*e**5 + 7098320*d*e**4*(d + e) + 1715200*d*e**3*(d + e)**2 - 6028
80*d*e**2*(d + e)**3 + 4283840*e**6 - 1292000*e**5*(d + e) - 762880*e**4*(d
 + e)**2 + 151040*e**3*(d + e)**3)/(801262*d**6 - 4662251*d**5*e + 7296938*
d**4*e**2 + 1388616*d**3*e**3 - 12447440*d**2*e**4 + 9990800*d*e**5 - 23800
00*e**6))/18 + (d + 2*e)*log(x + (-1534775*d**6 + 8032360*d**5*e - 984027*d
**5*(d + 2*e)/8 - 12991180*d**4*e**2 + 5898633*d**4*e*(d + 2*e)/4 + 55737*d
**4*(d + 2*e)**2 + 1075200*d**3*e**3 - 4090191*d**3*e**2*(d + 2*e) - 136332
*d**3*e*(d + 2*e)**2 - 3861*d**3*(d + 2*e)**3/8 + 16959280*d**2*e**4 + 4872
162*d**2*e**3*(d + 2*e) - 44064*d**2*e**2*(d + 2*e)**2 - 80919*d**2*e*(d +
2*e)**3/4 - 15836800*d*e**5 - 2661870*d*e**4*(d + 2*e) + 241200*d*e**3*(d +
 2*e)**2 + 63585*d*e**2*(d + 2*e)**3/2 + 4283840*e**6 + 484500*e**5*(d + 2*
e) - 107280*e**4*(d + 2*e)**2 - 7965*e**3*(d + 2*e)**3)/(801262*d**6 - 4662
251*d**5*e + 7296938*d**4*e**2 + 1388616*d**3*e**3 - 12447440*d**2*e**4 + 9
990800*d*e**5 - 2380000*e**6))/48 - (19*d - 26*e)*log(x + (-1534775*d**6 +
8032360*d**5*e + 328009*d**5*(19*d - 26*e)/8 - 12991180*d**4*e**2 - 1966211
*d**4*e*(19*d - 26*e)/4 + 6193*d**4*(19*d - 26*e)**2 + 1075200*d**3*e**3 +
1363397*d**3*e**2*(19*d - 26*e) - 15148*d**3*e*(19*d - 26*e)**2 + 143*d**3*
(19*d - 26*e)**3/8 + 16959280*d**2*e**4 - 1624054*d**2*e**3*(19*d - 26*e) -
 4896*d**2*e**2*(19*d - 26*e)**2 + 2997*d**2*e*(19*d - 26*e)**3/4 - 1583680
0*d*e**5 + 887290*d*e**4*(19*d - 26*e) + 26800*d*e**3*(19*d - 26*e)**2 - 23
55*d*e**2*(19*d - 26*e)**3/2 + 4283840*e**6 - 161500*e**5*(19*d - 26*e) - 1
1920*e**4*(19*d - 26*e)**2 + 295*e**3*(19*d - 26*e)**3)/(801262*d**6 - 4662
251*d**5*e + 7296938*d**4*e**2 + 1388616*d**3*e**3 - 12447440*d**2*e**4 + 9
990800*d*e**5 - 2380000*e**6))/144
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 57, normalized size of antiderivative = 0.80

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{1}{144}(19\,d-26\,e)\log(x+2) + \frac{1}{6}(d-e)\log(x+1) - \frac{1}{18}(d+e)\log(x-1) + \frac{1}{48}(d+2\,e)\log(x-2) + \frac{d-2\,e}{12\,(x+2)}$$

```
[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] -1/144*(19*d - 26*e)*log(x + 2) + 1/6*(d - e)*log(x + 1) - 1/18*(d + e)*log
(x - 1) + 1/48*(d + 2*e)*log(x - 2) + 1/12*(d - 2*e)/(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 61, normalized size of antiderivative = 0.86

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{1}{144}(19\,d-26\,e)\log(|x+2|) + \frac{1}{6}(d-e)\log(|x+1|) - \frac{1}{18}(d+e)\log(|x-1|) + \frac{1}{48}(d+2\,e)\log(|x-2|) + \frac{d-2\,e}{12\,(x+2)}$$

```
[In] integrate((e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] -1/144*(19*d - 26*e)*log(abs(x + 2)) + 1/6*(d - e)*log(abs(x + 1)) - 1/18*(
d + e)*log(abs(x - 1)) + 1/48*(d + 2*e)*log(abs(x - 2)) + 1/12*(d - 2*e)/(x
 + 2)
```

Mupad [B] (verification not implemented)

Time = 7.95 (sec) , antiderivative size = 64, normalized size of antiderivative = 0.90

$$\int \frac{(d+ex)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = \frac{\frac{d}{12}-\frac{e}{6}}{x+2} + \ln(x+1)\left(\frac{d}{6}-\frac{e}{6}\right) - \ln(x-1)\left(\frac{d}{18}+\frac{e}{18}\right) + \ln(x-2)\left(\frac{d}{48}+\frac{e}{24}\right) - \ln(x+2)\left(\frac{19\,d}{144}-\frac{13\,e}{72}\right)$$

```
[In] int(-((d + e*x)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] (d/12 - e/6)/(x + 2) + log(x + 1)*(d/6 - e/6) - log(x - 1)*(d/18 + e/18) +
log(x - 2)*(d/48 + e/24) - log(x + 2)*((19*d)/144 - (13*e)/72)
```

3.87 $\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 36, antiderivative size = 82

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e+4f}{12(2+x)} - \frac{1}{18}(d+e+f)\log(1-x) + \frac{1}{48}(d+2e+4f)\log(2-x) + \frac{1}{6}(d-e+f)\log(1+x) - \frac{1}{144}(19d-26e+28f)\log(2+x)$$

```
[Out] 1/12*(d-2*e+4*f)/(2+x)-1/18*(d+e+f)*ln(1-x)+1/48*(d+2*e+4*f)*ln(2-x)+1/6*(d
-e+f)*ln(1+x)-1/144*(19*d-26*e+28*f)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 82, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.056$, Rules used = {1600, 6874}

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e+4f}{12(x+2)} - \frac{1}{18}\log(1-x)(d+e+f) + \frac{1}{48}\log(2-x)(d+2e+4f) + \frac{1}{6}\log(x+1)(d-e+f) - \frac{1}{144}\log(x+2)(19d-26e+28f)$$

```
[In] Int[((d + e*x + f*x^2)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d - 2*e + 4*f)/(12*(2 + x)) - ((d + e + f)*Log[1 - x])/18 + ((d + 2*e + 4*
f)*Log[2 - x])/48 + ((d - e + f)*Log[1 + x])/6 - ((19*d - 26*e + 28*f)*Log[
2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{(2+x)^2\,(2-x-2x^2+x^3)}\,dx \\
&= \int \left(\frac{d+2e+4f}{48(-2+x)}+\frac{-d-e-f}{18(-1+x)}+\frac{d-e+f}{6(1+x)}+\frac{-d+2e-4f}{12(2+x)^2}+\frac{-19d+26e-28f}{144(2+x)}\right)dx \\
&= \frac{d-2e+4f}{12(2+x)}-\frac{1}{18}(d+e+f)\log(1-x)+\frac{1}{48}(d+2e+4f)\log(2-x) \\
&\quad +\frac{1}{6}(d-e+f)\log(1+x)-\frac{1}{144}(19d-26e+28f)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 77, normalized size of antiderivative = 0.94

$$\int \frac{(d+ex+fx^2)\,(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(\frac{12(d-2e+4f)}{2+x}+24(d-e+f)\log(-1-x)\right.$$
$$-8(d+e+f)\log(1-x)+3(d+2e+4f)\log(2-x)$$
$$\left.+(-19d+26e-28f)\log(2+x)\right)$$

```
[In] Integrate[((d + e*x + f*x^2)*(2 - x - 2*x^2 + x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(d - 2*e + 4*f))/(2 + x) + 24*(d - e + f)*Log[-1 - x] - 8*(d + e + f)*
Log[1 - x] + 3*(d + 2*e + 4*f)*Log[2 - x] + (-19*d + 26*e - 28*f)*Log[2 + x
])/144
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 79, normalized size of antiderivative = 0.96

method	result
default	$\left(\frac{13e}{72}-\frac{7f}{36}-\frac{19d}{144}\right)\ln\left(x+2\right)-\frac{-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}}{x+2}+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{18}-\frac{e}{18}-\frac{f}{18}\right)\ln\left(x-1\right)$
risch	$\frac{d}{12x+24}-\frac{e}{6(x+2)}+\frac{f}{3x+6}+\frac{13\ln(-x-2)e}{72}-\frac{7\ln(-x-2)f}{36}-\frac{19\ln(-x-2)d}{144}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}$
norman	$\frac{\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}\right)x+\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}\right)x^3+\left(\frac{e}{3}-\frac{2f}{3}-\frac{d}{6}\right)x^2-\frac{e}{3}+\frac{2f}{3}+\frac{d}{6}}{x^4-5x^2+4}+\left(-\frac{d}{18}-\frac{e}{18}-\frac{f}{18}\right)\ln\left(x-1\right)+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right)\ln$
parallelrisch	$48f+12d-24e+6\ln(x-2)d+12\ln(x-2)e-16\ln(x-1)d-16\ln(x-1)e-56\ln(x+2)f+48\ln(x+1)f+26\ln(x+2)xe+6\ln(x-2)xe-$

```
[In] int((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (13/72*e-7/36*f-19/144*d)*ln(x+2)-(-1/12*d+1/6*e-1/3*f)/(x+2)+(1/6*d-1/6*e+
1/6*f)*ln(x+1)+(-1/18*d-1/18*e-1/18*f)*ln(x-1)+(1/48*d+1/24*e+1/12*f)*ln(x-
2)
```

Fricas [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 116, normalized size of antiderivative = 1.41

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx =$$
$$-\frac{((19\,d-26\,e+28\,f)x+38\,d-52\,e+56\,f)\log(x+2)-24\,((d-e+f)x+2\,d-2\,e+2\,f)\log(x+1)}{}$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="frica
s")
```

```
[Out] -1/144*(((19*d - 26*e + 28*f)*x + 38*d - 52*e + 56*f)*log(x + 2) - 24*((d -
 e + f)*x + 2*d - 2*e + 2*f)*log(x + 1) + 8*((d + e + f)*x + 2*d + 2*e + 2*
f)*log(x - 1) - 3*((d + 2*e + 4*f)*x + 2*d + 4*e + 8*f)*log(x - 2) - 12*d +
 24*e - 48*f)/(x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)*(x**3-2*x**2-x+2)/(x**4-5*x**2+4)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 68, normalized size of antiderivative = 0.83

$$\begin{aligned}\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = &-\frac{1}{144}(19\,d-26\,e+28\,f)\log(x+2)\\&+\frac{1}{6}(d-e+f)\log(x+1)\\&-\frac{1}{18}(d+e+f)\log(x-1)\\&+\frac{1}{48}(d+2\,e+4\,f)\log(x-2)+\frac{d-2\,e+4\,f}{12\,(x+2)}\end{aligned}$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="maxim
a")
[Out] -1/144*(19*d - 26*e + 28*f)*log(x + 2) + 1/6*(d - e + f)*log(x + 1) - 1/18*
(d + e + f)*log(x - 1) + 1/48*(d + 2*e + 4*f)*log(x - 2) + 1/12*(d - 2*e +
4*f)/(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 72, normalized size of antiderivative = 0.88

$$\begin{aligned}\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = &-\frac{1}{144}(19\,d-26\,e+28\,f)\log(|x+2|)\\&+\frac{1}{6}(d-e+f)\log(|x+1|)\\&-\frac{1}{18}(d+e+f)\log(|x-1|)\\&+\frac{1}{48}(d+2\,e+4\,f)\log(|x-2|)+\frac{d-2\,e+4\,f}{12\,(x+2)}\end{aligned}$$

```
[In] integrate((f*x^2+e*x+d)*(x^3-2*x^2-x+2)/(x^4-5*x^2+4)^2,x, algorithm="giac"
)
```

```
[Out] -1/144*(19*d - 26*e + 28*f)*log(abs(x + 2)) + 1/6*(d - e + f)*log(abs(x + 1
)) - 1/18*(d + e + f)*log(abs(x - 1)) + 1/48*(d + 2*e + 4*f)*log(abs(x - 2)
) + 1/12*(d - 2*e + 4*f)/(x + 2)
```

Mupad [B] (verification not implemented)

Time = 7.90 (sec) , antiderivative size = 79, normalized size of antiderivative = 0.96

$$\int \frac{(d+ex+fx^2)(2-x-2x^2+x^3)}{(4-5x^2+x^4)^2}\,dx = \frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}}{x+2}+\ln(x+1)\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}\right) - \ln(x-1)\left(\frac{d}{18}+\frac{e}{18}+\frac{f}{18}\right) + \ln(x-2)\left(\frac{d}{48}+\frac{e}{24}+\frac{f}{12}\right) - \ln(x+2)\left(\frac{19\,d}{144}-\frac{13\,e}{72}+\frac{7\,f}{36}\right)$$

```
[In] int(-((d + e*x + f*x^2)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] (d/12 - e/6 + f/3)/(x + 2) + log(x + 1)*(d/6 - e/6 + f/6) - log(x - 1)*(d/1
8 + e/18 + f/18) + log(x - 2)*(d/48 + e/24 + f/12) - log(x + 2)*((19*d)/144
 - (13*e)/72 + (7*f)/36)
```

3.88 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 41, antiderivative size = 95

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e+4f-8g}{12(2+x)} - \frac{1}{18}(d+e+f+g)\log(1-x) + \frac{1}{48}(d+2e+4f+8g)\log(2-x) + \frac{1}{6}(d-e+f-g)\log(1+x) - \frac{1}{144}(19d-26e+28f-8g)\log(2+x)$$

```
[Out] 1/12*(d-2*e+4*f-8*g)/(2+x)-1/18*(d+e+f+g)*ln(1-x)+1/48*(d+2*e+4*f+8*g)*ln(2
-x)+1/6*(d-e+f-g)*ln(1+x)-1/144*(19*d-26*e+28*f-8*g)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.15 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.049, Rules used = {1600, 6874}

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2} dx = \frac{d-2e+4f-8g}{12(x+2)} - \frac{1}{18}\log(1-x)(d+e+f+g) + \frac{1}{48}\log(2-x)(d+2e+4f+8g) + \frac{1}{6}\log(x+1)(d-e+f-g) - \frac{1}{144}\log(x+2)(19d-26e+28f-8g)$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4)^2,x
]
```

```
[Out] (d - 2*e + 4*f - 8*g)/(12*(2 + x)) - ((d + e + f + g)*Log[1 - x])/18 + ((d
+ 2*e + 4*f + 8*g)*Log[2 - x])/48 + ((d - e + f - g)*Log[1 + x])/6 - ((19*d
 - 26*e + 28*f - 8*g)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2+gx^3}{(2+x)^2(2-x-2x^2+x^3)}\,dx$$

$$= \int \left(\frac{d+2e+4f+8g}{48(-2+x)} + \frac{-d-e-f-g}{18(-1+x)} + \frac{d-e+f-g}{6(1+x)} + \frac{-d+2e-4f+8g}{12(2+x)^2} + \frac{-19d+26e-28f+8g}{144(2+x)}\right) dx$$

$$= \frac{d-2e+4f-8g}{12(2+x)} - \frac{1}{18}(d+e+f+g)\log(1-x) + \frac{1}{48}(d+2e+4f+8g)\log(2-x) + \frac{1}{6}(d-e+f-g)\log(1+x) - \frac{1}{144}(19d-26e+28f-8g)\log(2+x)$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.95

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(\frac{12(d-2e+4f-8g)}{2+x} + 24(d-e+f-g)\log(-1-x) - 8(d+e+f+g)\log(1-x) + 3(d+2e+4f+8g)\log(2-x) + (-19d+26e-28f+8g)\log(2+x)\right)$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^
4)^2,x]
```

```
[Out] ((12*(d - 2*e + 4*f - 8*g))/(2 + x) + 24*(d - e + f - g)*Log[-1 - x] - 8*(d
 + e + f + g)*Log[1 - x] + 3*(d + 2*e + 4*f + 8*g)*Log[2 - x] + (-19*d + 26
*e - 28*f + 8*g)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 94, normalized size of antiderivative = 0.99

method	result
default	$\left(\frac{13e}{72}-\frac{7f}{36}+\frac{g}{18}-\frac{19d}{144}\right)\ln\left(x+2\right)-\frac{-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}}{x+2}+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)\ln\left(x+1\right)+\left(-\frac{d}{18}-\frac{e}{18}-\right.$
norman	$\frac{\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}\right)x+\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}\right)x^3+\left(\frac{e}{3}-\frac{2f}{3}+\frac{4g}{3}-\frac{d}{6}\right)x^2-\frac{e}{3}+\frac{2f}{3}-\frac{4g}{3}+\frac{d}{6}}{x^4-5x^2+4}+\left(-\frac{d}{18}-\frac{e}{18}-\frac{f}{18}-\frac{g}{18}\right)\ln\left(x-\right.$
risch	$\frac{d}{12x+24}-\frac{e}{6(x+2)}+\frac{f}{3x+6}-\frac{2g}{3(x+2)}+\frac{\ln(x+1)d}{6}-\frac{\ln(x+1)e}{6}+\frac{\ln(x+1)f}{6}-\frac{\ln(x+1)g}{6}-\frac{\ln(x-1)d}{18}-\frac{\ln(x-1)}{18}$
parallelrisch	$\frac{48f-96g+12d-24e+6\ln(x-2)d+12\ln(x-2)e-16\ln(x-1)d-16\ln(x-1)e+24\ln(x-2)xg-8\ln(x-1)xg-24\ln(x+1)xg+8\ln(x}{}$

[In] int((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVER
BOSE)

[Out] (13/72*e-7/36*f+1/18*g-19/144*d)*ln(x+2)-(-1/12*d+1/6*e-1/3*f+2/3*g)/(x+2)+
(1/6*d-1/6*e+1/6*f-1/6*g)*ln(x+1)+(-1/18*d-1/18*e-1/18*f-1/18*g)*ln(x-1)+(1
/48*d+1/24*e+1/12*f+1/6*g)*ln(x-2)

Fricas [A] (verification not implemented)

none

Time = 0.72 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.48

$$\int \frac{(2-x-2x^2+x^3)\left(d+ex+fx^2+gx^3\right)}{\left(4-5x^2+x^4\right)^2}\,dx =$$
$$-\frac{((19\,d-26\,e+28\,f-8\,g)x+38\,d-52\,e+56\,f-16\,g)\log\left(x+2\right)-24\left((d-e+f-g)x+2\,d-2\,e\right.}{}$$

[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm=
"fricas")

[Out] -1/144*(((19*d - 26*e + 28*f - 8*g)*x + 38*d - 52*e + 56*f - 16*g)*log(x +
2) - 24*((d - e + f - g)*x + 2*d - 2*e + 2*f - 2*g)*log(x + 1) + 8*((d + e
+ f + g)*x + 2*d + 2*e + 2*f + 2*g)*log(x - 1) - 3*((d + 2*e + 4*f + 8*g)*x
 + 2*d + 4*e + 8*f + 16*g)*log(x - 2) - 12*d + 24*e - 48*f + 96*g)/(x + 2)

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**3-2*x**2-x+2)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 81, normalized size of antiderivative = 0.85

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{1}{144}\,(19\,d-26\,e+28\,f-8\,g)\log(x+2)+\frac{1}{6}\,(d-e+f-g)\log(x+1) - \frac{1}{18}\,(d+e+f+g)\log(x-1)+\frac{1}{48}\,(d+2\,e+4\,f+8\,g)\log(x-2)+\frac{d-2\,e+4\,f-8\,g}{12\,(x+2)}$$

```
[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm=
"maxima")
[Out] -1/144*(19*d - 26*e + 28*f - 8*g)*log(x + 2) + 1/6*(d - e + f - g)*log(x +
1) - 1/18*(d + e + f + g)*log(x - 1) + 1/48*(d + 2*e + 4*f + 8*g)*log(x - 2
) + 1/12*(d - 2*e + 4*f - 8*g)/(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 85, normalized size of antiderivative = 0.89

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{1}{144}\,(19\,d-26\,e+28\,f-8\,g)\log(|x+2|) + \frac{1}{6}\,(d-e+f-g)\log(|x+1|)-\frac{1}{18}\,(d+e+f+g)\log(|x-1|) + \frac{1}{48}\,(d+2\,e+4\,f+8\,g)\log(|x-2|)+\frac{d-2\,e+4\,f-8\,g}{12\,(x+2)}$$

```
[In] integrate((x^3-2*x^2-x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm=
"giac")
```

```
[Out] -1/144*(19*d - 26*e + 28*f - 8*g)*log(abs(x + 2)) + 1/6*(d - e + f - g)*log
(abs(x + 1)) - 1/18*(d + e + f + g)*log(abs(x - 1)) + 1/48*(d + 2*e + 4*f +
 8*g)*log(abs(x - 2)) + 1/12*(d - 2*e + 4*f - 8*g)/(x + 2)
```

Mupad [B] (verification not implemented)

Time = 7.94 (sec) , antiderivative size = 94, normalized size of antiderivative = 0.99

$$\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2\,g}{3}}{x+2}+\ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}\right)-\ln\left(x-1\right)\,\left(\frac{d}{18}+\frac{e}{18}+\frac{f}{18}+\frac{g}{18}\right)$$
$$+\ln\left(x-2\right)\,\left(\frac{d}{48}+\frac{e}{24}+\frac{f}{12}+\frac{g}{6}\right)-\ln\left(x+2\right)\,\left(\frac{19\,d}{144}-\frac{13\,e}{72}+\frac{7\,f}{36}-\frac{g}{18}\right)$$

```
[In] int(-((d + e*x + f*x^2 + g*x^3)*(x + 2*x^2 - x^3 - 2))/(x^4 - 5*x^2 + 4)^2,
x)
```

```
[Out] (d/12 - e/6 + f/3 - (2*g)/3)/(x + 2) + log(x + 1)*(d/6 - e/6 + f/6 - g/6) -
 log(x - 1)*(d/18 + e/18 + f/18 + g/18) + log(x - 2)*(d/48 + e/24 + f/12 +
g/6) - log(x + 2)*((19*d)/144 - (13*e)/72 + (7*f)/36 - g/18)
```

3.89 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 46, antiderivative size = 106

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx &= \frac{d-2e+4f-8g+16h}{12(2+x)}-\frac{1}{18}(d+e+f+g+h)\log(1-x)\\&\quad+\frac{1}{48}(d+2e+4f+8g+16h)\log(2-x)+\frac{1}{6}(d-e+f-g+h)\log(1+x)\\&\quad-\frac{1}{144}(19d-26e+28f-8g-80h)\log(2+x)\end{aligned}$$

```
[Out] 1/12*(d-2*e+4*f-8*g+16*h)/(2+x)-1/18*(d+e+f+g+h)*ln(1-x)+1/48*(d+2*e+4*f+8*
g+16*h)*ln(2-x)+1/6*(d-e+f-g+h)*ln(1+x)-1/144*(19*d-26*e+28*f-8*g-80*h)*ln(
2+x)
```

Rubi [A] (verified)

Time = 0.19 (sec) , antiderivative size = 106, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.043, Rules used = {1600, 6874}

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx &= \frac{d-2e+4f-8g+16h}{12(x+2)}-\frac{1}{18}\log(1-x)(d+e+f+g+h)\\&\quad+\frac{1}{48}\log(2-x)(d+2e+4f+8g+16h)+\frac{1}{6}\log(x+1)(d-e+f-g+h)\\&\quad-\frac{1}{144}\log(x+2)(19d-26e+28f-8g-80h)\end{aligned}$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 +
x^4)^2,x]
```

```
[Out] (d - 2*e + 4*f - 8*g + 16*h)/(12*(2 + x)) - ((d + e + f + g + h)*Log[1 - x]
)/18 + ((d + 2*e + 4*f + 8*g + 16*h)*Log[2 - x])/48 + ((d - e + f - g + h)*
Log[1 + x])/6 - ((19*d - 26*e + 28*f - 8*g - 80*h)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4}{(2+x)^2\left(2-x-2x^2+x^3\right)}\,dx \\
&= \int \left(\frac{d+2e+4f+8g+16h}{48(-2+x)}+\frac{-d-e-f-g-h}{18(-1+x)}+\frac{d-e+f-g+h}{6(1+x)}\right. \\
&\qquad \left.+\frac{-d+2e-4f+8g-16h}{12(2+x)^2}+\frac{-19d+26e-28f+8g+80h}{144(2+x)}\right)dx \\
&= \frac{d-2e+4f-8g+16h}{12(2+x)}-\frac{1}{18}(d+e+f+g+h)\log(1-x) \\
&\quad +\frac{1}{48}(d+2e+4f+8g+16h)\log(2-x)+\frac{1}{6}(d-e+f-g+h)\log(1+x) \\
&\quad -\frac{1}{144}(19d-26e+28f-8g-80h)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 102, normalized size of antiderivative = 0.96

$$\begin{aligned}
&\int \frac{\left(2-x-2x^2+x^3\right)\left(d+ex+fx^2+gx^3+hx^4\right)}{\left(4-5x^2+x^4\right)^2}\,dx \\
&= \frac{1}{144}\left(\frac{12(d-2e+4f-8g+16h)}{2+x}+24(d-e+f-g+h)\log(-1-x)\right. \\
&\qquad -8(d+e+f+g+h)\log(1-x)+3(d+2(e+2f+4g+8h))\log(2-x) \\
&\qquad \left.+(-19d+26e-28f+8g+80h)\log(2+x)\right)
\end{aligned}$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*
x^2 + x^4)^2,x]
```

```
[Out] ((12*(d - 2*e + 4*f - 8*g + 16*h))/(2 + x) + 24*(d - e + f - g + h)*Log[-1
- x] - 8*(d + e + f + g + h)*Log[1 - x] + 3*(d + 2*(e + 2*f + 4*g + 8*h))*L
og[2 - x] + (-19*d + 26*e - 28*f + 8*g + 80*h)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 109, normalized size of antiderivative = 1.03

method	result
default	$\left(\frac{5h}{9}+\frac{g}{18}-\frac{7f}{36}+\frac{13e}{72}-\frac{19d}{144}\right)\ln\left(x+2\right)-\frac{-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}}{x+2}+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right)\ln\left(x+1\right)+$
norman	$\frac{\left(-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}\right)x+\left(\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}+\frac{4h}{3}\right)x^3+\left(-\frac{8h}{3}+\frac{4g}{3}-\frac{2f}{3}+\frac{e}{3}-\frac{d}{6}\right)x^2+\frac{8h}{3}-\frac{4g}{3}+\frac{2f}{3}-\frac{e}{3}+\frac{d}{6}}{x^4-5x^2+4}+\left(-\frac{d}{18}-\frac{e}{18}-\frac{f}{18}\right.$
risch	$-\frac{\ln(x-1)d}{18}-\frac{\ln(x-1)e}{18}+\frac{\ln(x+1)f}{6}+\frac{\ln(2-x)f}{12}+\frac{5\ln(-x-2)h}{9}+\frac{d}{12x+24}+\frac{\ln(2-x)d}{48}+\frac{\ln(2-x)e}{24}+\frac{\ln(x+1)d}{6}$
parallelrisch	$\frac{48f-96g+12d+192h-24e+48\ln(x-2)xh-8\ln(x-1)xh+24\ln(x+1)xh+80\ln(x+2)xh+6\ln(x-2)d+12\ln(x-2)e-16\ln(x-1)d}{}$

```
[In] int((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RET
URNVERBOSE)
```

```
[Out] (5/9*h+1/18*g-7/36*f+13/72*e-19/144*d)*ln(x+2)-(-1/12*d+1/6*e-1/3*f+2/3*g-4
/3*h)/(x+2)+(1/6*d-1/6*e+1/6*f-1/6*g+1/6*h)*ln(x+1)+(-1/18*d-1/18*e-1/18*f-
1/18*g-1/18*h)*ln(x-1)+(1/48*d+1/24*e+1/12*f+1/6*g+1/3*h)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 3.26 (sec) , antiderivative size = 164, normalized size of antiderivative = 1.55

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = -\frac{((19\,d-26\,e+28\,f-8\,g-80\,h)x+38\,d-52\,e+56\,f-16\,g-160\,h)\log\left(x+2\right)-24\left((d-e+f-g\right.}{}$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algo
rithm="fricas")
```

```
[Out] -1/144*(((19*d - 26*e + 28*f - 8*g - 80*h)*x + 38*d - 52*e + 56*f - 16*g -
160*h)*log(x + 2) - 24*((d - e + f - g + h)*x + 2*d - 2*e + 2*f - 2*g + 2*h
)*log(x + 1) + 8*((d + e + f + g + h)*x + 2*d + 2*e + 2*f + 2*g + 2*h)*log(
x - 1) - 3*((d + 2*e + 4*f + 8*g + 16*h)*x + 2*d + 4*e + 8*f + 16*g + 32*h)
*log(x - 2) - 12*d + 24*e - 48*f + 96*g - 192*h)/(x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**3-2*x**2-x+2)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2
,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.18 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.87

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\ = -\frac{1}{144}(19\,d-26\,e+28\,f-8\,g-80\,h)\log(x+2) \\ +\frac{1}{6}(d-e+f-g+h)\log(x+1)-\frac{1}{18}(d+e+f+g+h)\log(x-1) \\ +\frac{1}{48}(d+2\,e+4\,f+8\,g+16\,h)\log(x-2)+\frac{d-2\,e+4\,f-8\,g+16\,h}{12\,(x+2)}\end{aligned}$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algo
rithm="maxima")
```

```
[Out] -1/144*(19*d - 26*e + 28*f - 8*g - 80*h)*log(x + 2) + 1/6*(d - e + f - g +
h)*log(x + 1) - 1/18*(d + e + f + g + h)*log(x - 1) + 1/48*(d + 2*e + 4*f +
 8*g + 16*h)*log(x - 2) + 1/12*(d - 2*e + 4*f - 8*g + 16*h)/(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 96, normalized size of antiderivative = 0.91

$$\begin{aligned}\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\ = -\frac{1}{144}(19\,d-26\,e+28\,f-8\,g-80\,h)\log(|x+2|) \\ +\frac{1}{6}(d-e+f-g+h)\log(|x+1|)-\frac{1}{18}(d+e+f+g+h)\log(|x-1|) \\ +\frac{1}{48}(d+2\,e+4\,f+8\,g+16\,h)\log(|x-2|)+\frac{d-2\,e+4\,f-8\,g+16\,h}{12\,(x+2)}\end{aligned}$$

```
[In] integrate((x^3-2*x^2-x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algo
rithm="giac")
```

```
[Out] -1/144*(19*d - 26*e + 28*f - 8*g - 80*h)*log(abs(x + 2)) + 1/6*(d - e + f -
 g + h)*log(abs(x + 1)) - 1/18*(d + e + f + g + h)*log(abs(x - 1)) + 1/48*(
d + 2*e + 4*f + 8*g + 16*h)*log(abs(x - 2)) + 1/12*(d - 2*e + 4*f - 8*g + 1
6*h)/(x + 2)
```

Mupad [B] (verification not implemented)

Time = 8.11 (sec) , antiderivative size = 108, normalized size of antiderivative = 1.02

$$
\begin{aligned}
&\int \frac{(2-x-2x^2+x^3)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2\,g}{3}+\frac{4\,h}{3}}{x+2}+\ln\left(x+1\right)\,\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right) \\
&\quad -\ln\left(x-1\right)\,\left(\frac{d}{18}+\frac{e}{18}+\frac{f}{18}+\frac{g}{18}+\frac{h}{18}\right)+\ln\left(x-2\right)\,\left(\frac{d}{48}+\frac{e}{24}+\frac{f}{12}+\frac{g}{6}+\frac{h}{3}\right) \\
&\quad +\ln\left(x+2\right)\,\left(\frac{13\,e}{72}-\frac{19\,d}{144}-\frac{7\,f}{36}+\frac{g}{18}+\frac{5\,h}{9}\right)
\end{aligned}
$$

```
[In] int(-((x + 2*x^2 - x^3 - 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2
 + 4)^2,x)
```

```
[Out] (d/12 - e/6 + f/3 - (2*g)/3 + (4*h)/3)/(x + 2) + log(x + 1)*(d/6 - e/6 + f/
6 - g/6 + h/6) - log(x - 1)*(d/18 + e/18 + f/18 + g/18 + h/18) + log(x - 2)
*(d/48 + e/24 + f/12 + g/6 + h/3) + log(x + 2)*((13*e)/72 - (19*d)/144 - (7
*f)/36 + g/18 + (5*h)/9)
```

3.90 $\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 51, antiderivative size = 122

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2} dx = ix+\frac{d-2e+4f-8g+16h-32i}{12(2+x)}-\frac{1}{18}(d+e+f+g+h+i)\log(1-x) + \frac{1}{48}(d+2e+4f+8g+16h+32i)\log(2-x)+\frac{1}{6}(d-e+f-g+h-i)\log(1+x) - \frac{1}{144}(19d-26e+28f-8g-80h+352i)\log(2+x)$$

```
[Out] i*x+1/12*(d-2*e+4*f-8*g+16*h-32*i)/(2+x)-1/18*(d+e+f+g+h+i)*ln(1-x)+1/48*(d
+2*e+4*f+8*g+16*h+32*i)*ln(2-x)+1/6*(d-e+f-g+h-i)*ln(1+x)-1/144*(19*d-26*e+
28*f-8*g-80*h+352*i)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.039, Rules used = {1600, 6874}

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2} dx = \frac{d-2e+4f-8g+16h-32i}{12(x+2)}-\frac{1}{18}\log(1-x)(d+e+f+g+h+i) + \frac{1}{48}\log(2-x)(d+2e+4f+8g+16h+32i)+\frac{1}{6}\log(x+1)(d-e+f-g+h-i) - \frac{1}{144}\log(x+2)(19d-26e+28f-8g-80h+352i)+ix$$

```
[In] Int[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 -
5*x^2 + x^4)^2,x]
```

```
[Out] i*x + (d - 2*e + 4*f - 8*g + 16*h - 32*i)/(12*(2 + x)) - ((d + e + f + g +
h + i)*Log[1 - x])/18 + ((d + 2*e + 4*f + 8*g + 16*h + 32*i)*Log[2 - x])/48
 + ((d - e + f - g + h - i)*Log[1 + x])/6 - ((19*d - 26*e + 28*f - 8*g - 80
*h + 352*i)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(2+x)^2(2-x-2x^2+x^3)}\,dx \\
&= \int \left(i+\frac{d+2e+4f+8g+16h+32i}{48(-2+x)}+\frac{-d-e-f-g-h-i}{18(-1+x)}+\frac{d-e+f-g+h-i}{6(1+x)}\right. \\
&\quad \left.+\frac{-d+2e-4f+8g-16h+32i}{12(2+x)^2}+\frac{-19d+26e-28f+8g+80h-352i}{144(2+x)}\right)dx \\
&= ix+\frac{d-2e+4f-8g+16h-32i}{12(2+x)}-\frac{1}{18}(d+e+f+g+h+i)\log(1-x) \\
&\quad +\frac{1}{48}(d+2e+4f+8g+16h+32i)\log(2-x)+\frac{1}{6}(d-e+f-g+h-i)\log(1+x) \\
&\quad -\frac{1}{144}(19d-26e+28f-8g-80h+352i)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 118, normalized size of antiderivative = 0.97

$$\begin{aligned}
&\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{1}{144}\left(144ix+\frac{12(d-2(e-2f+4g-8h+16i))}{2+x}-8(d+e+f+g+h+i)\log(1-x)\right. \\
&\quad +3(d+2e+4(f+2g+4h+8i))\log(2-x)+24(d-e+f-g+h-i)\log(1+x) \\
&\quad \left.+(-19d+26e-28f+8g+80h-352i)\log(2+x)\right)
\end{aligned}$$

```
[In] Integrate[((2 - x - 2*x^2 + x^3)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))
/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (144*i*x + (12*(d - 2*(e - 2*f + 4*g - 8*h + 16*i)))/(2 + x) - 8*(d + e + f
 + g + h + i)*Log[1 - x] + 3*(d + 2*e + 4*(f + 2*g + 4*h + 8*i))*Log[2 - x]
 + 24*(d - e + f - g + h - i)*Log[1 + x] + (-19*d + 26*e - 28*f + 8*g + 80*
h - 352*i)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.04

method	result
default	$ix+\left(\frac{5h}{9}-\frac{22i}{9}+\frac{g}{18}-\frac{7f}{36}+\frac{13e}{72}-\frac{19d}{144}\right)\ln\left(x+2\right)-\frac{-\frac{d}{12}+\frac{e}{6}-\frac{f}{3}+\frac{2g}{3}-\frac{4h}{3}+\frac{8i}{3}}{x+2}+\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}\right.$
norman	$\frac{i\,x^{5}+\left(-\frac{4h}{3}+\frac{20i}{3}+\frac{2g}{3}-\frac{f}{3}+\frac{e}{6}-\frac{d}{12}\right)x+\left(-\frac{23i}{3}+\frac{4h}{3}-\frac{2g}{3}+\frac{f}{3}-\frac{e}{6}+\frac{d}{12}\right)x^{3}+\left(-\frac{d}{6}+\frac{e}{3}-\frac{2f}{3}+\frac{4g}{3}-\frac{8h}{3}+\frac{16i}{3}\right)x^{2}+\frac{8h}{3}-\frac{16i}{3}+\frac{d}{6}-\frac{e}{3}+\frac{2f}{3}-}{x^{4}-5x^{2}+4}$
risch	$-\frac{\ln(x-1)d}{18}-\frac{\ln(x-1)e}{18}-\frac{\ln(x+1)i}{6}+\frac{\ln(x+1)f}{6}+\frac{\ln(2-x)f}{12}+\frac{5\ln(-x-2)h}{9}-\frac{22\ln(-x-2)i}{9}+\frac{d}{12x+24}+\frac{\ln(2-}{4}$
parallelrisch	$\frac{-960i+48f-96g-352\ln(x+2)xi+96\ln(x-2)xi+12d+192h-24e+48\ln(x-2)xh-8\ln(x-1)xh+24\ln(x+1)xh+80\ln(x+2)xh}{}$

```
[In] int((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,metho
d=_RETURNVERBOSE)
```

```
[Out] i*x+(5/9*h-22/9*i+1/18*g-7/36*f+13/72*e-19/144*d)*ln(x+2)-(-1/12*d+1/6*e-1/
3*f+2/3*g-4/3*h+8/3*i)/(x+2)+(1/6*d-1/6*e+1/6*f-1/6*g+1/6*h-1/6*i)*ln(x+1)+
(-1/18*d-1/18*e-1/18*f-1/18*g-1/18*h-1/18*i)*ln(x-1)+(1/48*d+1/24*e+1/12*f+
1/6*g+1/3*h+2/3*i)*ln(x-2)
```

Fricas [A] (verification not implemented)

none

Time = 19.69 (sec) , antiderivative size = 200, normalized size of antiderivative = 1.64

$$\int \frac{(2-x-2x^2+x^3)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx$$
$$=\frac{144\,ix^2+288\,ix-((19\,d-26\,e+28\,f-8\,g-80\,h+352\,i)x+38\,d-52\,e+56\,f-16\,g-160\,h+704\,i}{}$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x
, algorithm="fricas")
```

```
[Out] 1/144*(144*i*x^2 + 288*i*x - ((19*d - 26*e + 28*f - 8*g - 80*h + 352*i)*x +
 38*d - 52*e + 56*f - 16*g - 160*h + 704*i)*log(x + 2) + 24*((d - e + f - g
 + h - i)*x + 2*d - 2*e + 2*f - 2*g + 2*h - 2*i)*log(x + 1) - 8*((d + e + f
```

```
 + g + h + i)*x + 2*d + 2*e + 2*f + 2*g + 2*h + 2*i)*log(x - 1) + 3*((d + 2
*e + 4*f + 8*g + 16*h + 32*i)*x + 2*d + 4*e + 8*f + 16*g + 32*h + 64*i)*log
(x - 2) + 12*d - 24*e + 48*f - 96*g + 192*h - 384*i)/(x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**3-2*x**2-x+2)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**
2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 108, normalized size of antiderivative = 0.89

$$\begin{aligned}
&\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx \\
&= ix - \frac{1}{144}(19\,d - 26\,e + 28\,f - 8\,g - 80\,h + 352\,i)\log(x+2) \\
&\quad + \frac{1}{6}(d - e + f - g + h - i)\log(x+1) - \frac{1}{18}(d + e + f + g + h + i)\log(x-1) \\
&\quad + \frac{1}{48}(d + 2\,e + 4\,f + 8\,g + 16\,h + 32\,i)\log(x-2) + \frac{d - 2\,e + 4\,f - 8\,g + 16\,h - 32\,i}{12\,(x+2)}
\end{aligned}$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x
, algorithm="maxima")
```

```
[Out] i*x - 1/144*(19*d - 26*e + 28*f - 8*g - 80*h + 352*i)*log(x + 2) + 1/6*(d -
 e + f - g + h - i)*log(x + 1) - 1/18*(d + e + f + g + h + i)*log(x - 1) +
1/48*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(x - 2) + 1/12*(d - 2*e + 4*f -
 8*g + 16*h - 32*i)/(x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 112, normalized size of antiderivative = 0.92

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$$
$$= ix - \frac{1}{144}(19\,d-26\,e+28\,f-8\,g-80\,h+352\,i)\log(|x+2|) + \frac{1}{6}(d-e+f-g+h-i)\log(|x+1|) - \frac{1}{18}(d+e+f+g+h+i)\log(|x-1|) + \frac{1}{48}(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log(|x-2|) + \frac{d-2\,e+4\,f-8\,g+16\,h-32\,i}{12\,(x+2)}$$

```
[In] integrate((x^3-2*x^2-x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x
, algorithm="giac")
```

```
[Out] i*x - 1/144*(19*d - 26*e + 28*f - 8*g - 80*h + 352*i)*log(abs(x + 2)) + 1/6
*(d - e + f - g + h - i)*log(abs(x + 1)) - 1/18*(d + e + f + g + h + i)*log
(abs(x - 1)) + 1/48*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(abs(x - 2)) + 1
/12*(d - 2*e + 4*f - 8*g + 16*h - 32*i)/(x + 2)
```

Mupad [B] (verification not implemented)

Time = 8.37 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.04

$$\int \frac{(2-x-2x^2+x^3)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$$
$$= i\,x + \frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2\,g}{3}+\frac{4\,h}{3}-\frac{8\,i}{3}}{x+2} + \ln(x+1)\left(\frac{d}{6}-\frac{e}{6}+\frac{f}{6}-\frac{g}{6}+\frac{h}{6}-\frac{i}{6}\right) + \ln(x-2)\left(\frac{d}{48}+\frac{e}{24}+\frac{f}{12}+\frac{g}{6}+\frac{h}{3}+\frac{2\,i}{3}\right) - \ln(x-1)\left(\frac{d}{18}+\frac{e}{18}+\frac{f}{18}+\frac{g}{18}+\frac{h}{18}+\frac{i}{18}\right) - \ln(x+2)\left(\frac{19\,d}{144}-\frac{13\,e}{72}+\frac{7\,f}{36}-\frac{g}{18}-\frac{5\,h}{9}+\frac{22\,i}{9}\right)$$

```
[In] int(-((x + 2*x^2 - x^3 - 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4
 - 5*x^2 + 4)^2,x)
```

```
[Out] i*x + (d/12 - e/6 + f/3 - (2*g)/3 + (4*h)/3 - (8*i)/3)/(x + 2) + log(x + 1)
*(d/6 - e/6 + f/6 - g/6 + h/6 - i/6) + log(x - 2)*(d/48 + e/24 + f/12 + g/6
 + h/3 + (2*i)/3) - log(x - 1)*(d/18 + e/18 + f/18 + g/18 + h/18 + i/18) -
log(x + 2)*((19*d)/144 - (13*e)/72 + (7*f)/36 - g/18 - (5*h)/9 + (22*i)/9)
```

3.91 $\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 21, antiderivative size = 56

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = -\frac{5+3x}{12(2+3x+x^2)}-\frac{1}{36}\log(1-x)+\frac{1}{144}\log(2-x)-\frac{7}{36}\log(1+x)+\frac{31}{144}\log(2+x)$$

```
[Out] 1/12*(-5-3*x)/(x^2+3*x+2)-1/36*ln(1-x)+1/144*ln(2-x)-7/36*ln(1+x)+31/144*ln
(2+x)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 56, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.238, Rules used = {1600, 988, 1086, 646, 31}

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = -\frac{3x+5}{12(x^2+3x+2)}-\frac{1}{36}\log(1-x)+\frac{1}{144}\log(2-x)-\frac{7}{36}\log(x+1)+\frac{31}{144}\log(x+2)$$

```
[In] Int[(2 - 3*x + x^2)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] -1/12*(5 + 3*x)/(2 + 3*x + x^2) - Log[1 - x]/36 + Log[2 - x]/144 - (7*Log[1
 + x])/36 + (31*Log[2 + x])/144
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 988

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_)*((d_.) + (e_.)*(x_) + (f_.)*(x
_)^2)^(q_), x_Symbol] :> Simp[(2*a*c^2*e - b^2*c*e + b^3*f + b*c*(c*d - 3*a
*f) + c*(2*c^2*d + b^2*f - c*(b*e + 2*a*f))*x)*(a + b*x + c*x^2)^(p + 1)*((
d + e*x + f*x^2)^(q + 1)/((b^2 - 4*a*c)*((c*d - a*f)^2 - (b*d - a*e)*(c*e -
 b*f))*(p + 1))), x] - Dist[1/((b^2 - 4*a*c)*((c*d - a*f)^2 - (b*d - a*e)*(
c*e - b*f))*(p + 1)), Int[(a + b*x + c*x^2)^(p + 1)*(d + e*x + f*x^2)^q*Sim
p[2*c*((c*d - a*f)^2 - (b*d - a*e)*(c*e - b*f))*(p + 1) - (2*c^2*d + b^2*f
- c*(b*e + 2*a*f))*(a*f*(p + 1) - c*d*(p + 2)) - e*(b^2*c*e - 2*a*c^2*e - b
^3*f - b*c*(c*d - 3*a*f))*(p + q + 2) + (2*f*(2*a*c^2*e - b^2*c*e + b^3*f +
 b*c*(c*d - 3*a*f))*(p + q + 2) - (2*c^2*d + b^2*f - c*(b*e + 2*a*f))*(b*f*
(p + 1) - c*e*(2*p + q + 4)))*x + c*f*(2*c^2*d + b^2*f - c*(b*e + 2*a*f))*(
2*p + 2*q + 5)*x^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, q}, x] && NeQ[b^
2 - 4*a*c, 0] && NeQ[e^2 - 4*d*f, 0] && LtQ[p, -1] && NeQ[(c*d - a*f)^2 - (
b*d - a*e)*(c*e - b*f), 0] &&  !( !IntegerQ[p] && ILtQ[q, -1]) &&  !IGtQ[q,
 0]
```

Rule 1086

```
Int[((A_.) + (B_.)*(x_) + (C_.)*(x_)^2)/(((a_) + (b_.)*(x_) + (c_.)*(x_)^2)
*((d_) + (e_.)*(x_) + (f_.)*(x_)^2)), x_Symbol] :> With[{q = c^2*d^2 - b*c*
d*e + a*c*e^2 + b^2*d*f - 2*a*c*d*f - a*b*e*f + a^2*f^2}, Dist[1/q, Int[(A*
c^2*d - a*c*C*d - A*b*c*e + a*B*c*e + A*b^2*f - a*b*B*f - a*A*c*f + a^2*C*f
 + c*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B*f)*x)/(a + b*x + c*x^2),
x], x] + Dist[1/q, Int[(c*C*d^2 - B*c*d*e + A*c*e^2 + b*B*d*f - A*c*d*f - a
*C*d*f - A*b*e*f + a*A*f^2 - f*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B
*f)*x)/(d + e*x + f*x^2), x], x] /; NeQ[q, 0]] /; FreeQ[{a, b, c, d, e, f,
A, B, C}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[e^2 - 4*d*f, 0]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{1}{(2-3x+x^2)(2+3x+x^2)^2}\,dx \\
&= -\frac{5+3x}{12(2+3x+x^2)} + \frac{1}{72}\int \frac{-18+48x-18x^2}{(2-3x+x^2)(2+3x+x^2)}\,dx \\
&= -\frac{5+3x}{12(2+3x+x^2)} + \frac{\int \frac{252-108x}{2-3x+x^2}\,dx}{5184} + \frac{\int \frac{-900+108x}{2+3x+x^2}\,dx}{5184} \\
&= -\frac{5+3x}{12(2+3x+x^2)} + \frac{1}{144}\int \frac{1}{-2+x}\,dx \\
&\quad -\frac{1}{36}\int \frac{1}{-1+x}\,dx - \frac{7}{36}\int \frac{1}{1+x}\,dx + \frac{31}{144}\int \frac{1}{2+x}\,dx \\
&= -\frac{5+3x}{12(2+3x+x^2)} - \frac{1}{36}\log(1-x) + \frac{1}{144}\log(2-x) - \frac{7}{36}\log(1+x) + \frac{31}{144}\log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 48, normalized size of antiderivative = 0.86

$$
\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(-\frac{12(5+3x)}{2+3x+x^2} - 4\log(1-x) + \log(2-x) - 28\log(1+x) + 31\log(2+x)\right)
$$

```
[In] Integrate[(2 - 3*x + x^2)/(4 - 5*x^2 + x^4)^2,x]
[Out] ((-12*(5 + 3*x))/(2 + 3*x + x^2) - 4*Log[1 - x] + Log[2 - x] - 28*Log[1 + x
] + 31*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 40, normalized size of antiderivative = 0.71

method	result
default	$-\frac{1}{12(x+2)}+\frac{31\ln(x+2)}{144}-\frac{1}{6(x+1)}-\frac{7\ln(x+1)}{36}-\frac{\ln(x-1)}{36}+\frac{\ln(x-2)}{144}$
risch	$\frac{-\frac{x}{4}-\frac{5}{12}}{x^2+3x+2}+\frac{\ln(x-2)}{144}-\frac{\ln(x-1)}{36}-\frac{7\ln(x+1)}{36}+\frac{31\ln(x+2)}{144}$
norman	$\frac{\frac{1}{3}x^2+\frac{3}{4}x-\frac{1}{4}x^3-\frac{5}{6}}{x^4-5x^2+4}+\frac{\ln(x-2)}{144}-\frac{\ln(x-1)}{36}-\frac{7\ln(x+1)}{36}+\frac{31\ln(x+2)}{144}$
parallelrisch	$\frac{\ln(x-2)x^2-4\ln(x-1)x^2-28\ln(x+1)x^2+31\ln(x+2)x^2-60+3\ln(x-2)x-12\ln(x-1)x-84\ln(x+1)x+93\ln(x+2)x+2\ln(x-2)-}{144x^2+432x+288}$

```
[In] int((x^2-3*x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -1/12/(x+2)+31/144*ln(x+2)-1/6/(x+1)-7/36*ln(x+1)-1/36*ln(x-1)+1/144*ln(x-2
)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 72, normalized size of antiderivative = 1.29

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{31\,(x^2+3\,x+2)\log{(x+2)}-28\,(x^2+3\,x+2)\log{(x+1)}-4\,(x^2+3\,x+2)\log{(x-1)}+(x^2+3\,x+2)}{144\,(x^2+3\,x+2)}$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] 1/144*(31*(x^2 + 3*x + 2)*log(x + 2) - 28*(x^2 + 3*x + 2)*log(x + 1) - 4*(x
^2 + 3*x + 2)*log(x - 1) + (x^2 + 3*x + 2)*log(x - 2) - 36*x - 60)/(x^2 + 3
*x + 2)
```

Sympy [A] (verification not implemented)

Time = 0.15 (sec) , antiderivative size = 46, normalized size of antiderivative = 0.82

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = \frac{-3x-5}{12x^2+36x+24}+\frac{\log{(x-2)}}{144}-\frac{\log{(x-1)}}{36} - \frac{7\log{(x+1)}}{36}+\frac{31\log{(x+2)}}{144}$$

```
[In] integrate((x**2-3*x+2)/(x**4-5*x**2+4)**2,x)
```

```
[Out] (-3*x - 5)/(12*x**2 + 36*x + 24) + log(x - 2)/144 - log(x - 1)/36 - 7*log(x
 + 1)/36 + 31*log(x + 2)/144
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 42, normalized size of antiderivative = 0.75

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = -\frac{3\,x+5}{12\,(x^2+3\,x+2)}+\frac{31}{144}\,\log{(x+2)} - \frac{7}{36}\,\log{(x+1)}-\frac{1}{36}\,\log{(x-1)}+\frac{1}{144}\,\log{(x-2)}$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] -1/12*(3*x + 5)/(x^2 + 3*x + 2) + 31/144*log(x + 2) - 7/36*log(x + 1) - 1/3
6*log(x - 1) + 1/144*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 46, normalized size of antiderivative = 0.82

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = -\frac{3\,x+5}{12\,(x+2)(x+1)} + \frac{31}{144}\,\log\left(|x+2|\right) - \frac{7}{36}\,\log\left(|x+1|\right) - \frac{1}{36}\,\log\left(|x-1|\right) + \frac{1}{144}\,\log\left(|x-2|\right)$$

```
[In] integrate((x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] -1/12*(3*x + 5)/((x + 2)*(x + 1)) + 31/144*log(abs(x + 2)) - 7/36*log(abs(x
 + 1)) - 1/36*log(abs(x - 1)) + 1/144*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 42, normalized size of antiderivative = 0.75

$$\int \frac{2-3x+x^2}{(4-5x^2+x^4)^2}\,dx = \frac{\ln\left(x-2\right)}{144} - \frac{7\,\ln\left(x+1\right)}{36} - \frac{\ln\left(x-1\right)}{36} + \frac{31\,\ln\left(x+2\right)}{144} - \frac{\frac{x}{4}+\frac{5}{12}}{x^2+3\,x+2}$$

```
[In] int((x^2 - 3*x + 2)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 2)/144 - (7*log(x + 1))/36 - log(x - 1)/36 + (31*log(x + 2))/144 -
(x/4 + 5/12)/(3*x + x^2 + 2)
```

3.92 $\int \frac{(d+ex)(2-3x+x^2)}{(4-5x^2+x^4)^2} \, dx$

Optimal result

Integrand size = 26, antiderivative size = 89

$$\int \frac{(d+ex)\left(2-3x+x^2\right)}{\left(4-5x^2+x^4\right)^2} \, dx = -\frac{5d-6e+(3d-4e)x}{12\left(2+3x+x^2\right)} - \frac{1}{36}(d+e)\log(1-x) + \frac{1}{144}(d+2e)\log(2-x) - \frac{1}{36}(7d-13e)\log(1+x) + \frac{1}{144}(31d-50e)\log(2+x)$$

```
[Out] 1/12*(-5*d+6*e-(3*d-4*e)*x)/(x^2+3*x+2)-1/36*(d+e)*ln(1-x)+1/144*(d+2*e)*ln
(2-x)-1/36*(7*d-13*e)*ln(1+x)+1/144*(31*d-50*e)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 89, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.192, Rules used = {1600, 1030, 1086, 646, 31}

$$\int \frac{(d+ex)\left(2-3x+x^2\right)}{\left(4-5x^2+x^4\right)^2} \, dx = -\frac{x(3d-4e)+5d-6e}{12\left(x^2+3x+2\right)} - \frac{1}{36}(d+e)\log(1-x) + \frac{1}{144}(d+2e)\log(2-x) - \frac{1}{36}(7d-13e)\log(x+1) + \frac{1}{144}(31d-50e)\log(x+2)$$

```
[In] Int[((d + e*x)*(2 - 3*x + x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] -1/12*(5*d - 6*e + (3*d - 4*e)*x)/(2 + 3*x + x^2) - ((d + e)*Log[1 - x])/36
 + ((d + 2*e)*Log[2 - x])/144 - ((7*d - 13*e)*Log[1 + x])/36 + ((31*d - 50*
e)*Log[2 + x])/144
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1030

```
Int[((g_.) + (h_.)*(x_))*((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_)*((d_) + (e
_.)*(x_) + (f_.)*(x_)^2)^(q_), x_Symbol] :> Simp[(a + b*x + c*x^2)^(p + 1)*
((d + e*x + f*x^2)^(q + 1)/((b^2 - 4*a*c)*((c*d - a*f)^2 - (b*d - a*e)*(c*e
 - b*f))*(p + 1)))*(g*c*(2*a*c*e - b*(c*d + a*f)) + (g*b - a*h)*(2*c^2*d +
b^2*f - c*(b*e + 2*a*f)) + c*(g*(2*c^2*d + b^2*f - c*(b*e + 2*a*f)) - h*(b*
c*d - 2*a*c*e + a*b*f))*x), x] + Dist[1/((b^2 - 4*a*c)*((c*d - a*f)^2 - (b*
d - a*e)*(c*e - b*f))*(p + 1)), Int[(a + b*x + c*x^2)^(p + 1)*(d + e*x + f*
x^2)^q*Simp[(b*h - 2*g*c)*((c*d - a*f)^2 - (b*d - a*e)*(c*e - b*f))*(p + 1)
 + (b^2*(g*f) - b*(h*c*d + g*c*e + a*h*f) + 2*(g*c*(c*d - a*f) - a*((-h)*c*
e)))*(a*f*(p + 1) - c*d*(p + 2)) - e*((g*c)*(2*a*c*e - b*(c*d + a*f)) + (g*
b - a*h)*(2*c^2*d + b^2*f - c*(b*e + 2*a*f)))*(p + q + 2) - (2*f*((g*c)*(2*
a*c*e - b*(c*d + a*f)) + (g*b - a*h)*(2*c^2*d + b^2*f - c*(b*e + 2*a*f)))*(
p + q + 2) - (b^2*g*f - b*(h*c*d + g*c*e + a*h*f) + 2*(g*c*(c*d - a*f) - a*
((-h)*c*e)))*(b*f*(p + 1) - c*e*(2*p + q + 4)))*x - c*f*(b^2*(g*f) - b*(h*c
*d + g*c*e + a*h*f) + 2*(g*c*(c*d - a*f) + a*h*c*e))*(2*p + 2*q + 5)*x^2, x
], x], x] /; FreeQ[{a, b, c, d, e, f, g, h, q}, x] && NeQ[b^2 - 4*a*c, 0] &
& NeQ[e^2 - 4*d*f, 0] && LtQ[p, -1] && NeQ[(c*d - a*f)^2 - (b*d - a*e)*(c*e
 - b*f), 0] &&  !( !IntegerQ[p] && ILtQ[q, -1])
```

Rule 1086

```
Int[((A_.) + (B_.)*(x_) + (C_.)*(x_)^2)/(((a_) + (b_.)*(x_) + (c_.)*(x_)^2)
*((d_) + (e_.)*(x_) + (f_.)*(x_)^2)), x_Symbol] :> With[{q = c^2*d^2 - b*c*
d*e + a*c*e^2 + b^2*d*f - 2*a*c*d*f - a*b*e*f + a^2*f^2}, Dist[1/q, Int[(A*
c^2*d - a*c*C*d - A*b*c*e + a*B*c*e + A*b^2*f - a*b*B*f - a*A*c*f + a^2*C*f
 + c*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B*f)*x)/(a + b*x + c*x^2),
x], x] + Dist[1/q, Int[(c*C*d^2 - B*c*d*e + A*c*e^2 + b*B*d*f - A*c*d*f - a
*C*d*f - A*b*e*f + a*A*f^2 - f*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B
*f)*x)/(d + e*x + f*x^2), x], x] /; NeQ[q, 0]] /; FreeQ[{a, b, c, d, e, f,
A, B, C}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[e^2 - 4*d*f, 0]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex}{(2-3x+x^2)\left(2+3x+x^2\right)^2}\,dx \\
&= -\frac{5d-6e+(3d-4e)x}{12\left(2+3x+x^2\right)}-\frac{1}{72}\int \frac{6(3d-10e)-24(2d-3e)x+6(3d-4e)x^2}{(2-3x+x^2)\left(2+3x+x^2\right)}\,dx \\
&= -\frac{5d-6e+(3d-4e)x}{12\left(2+3x+x^2\right)}-\frac{\int \frac{108(3d-10e)-288(2d-3e)+(-36(3d-10e)+72(3d-4e))x}{2-3x+x^2}\,dx}{5184} \\
&\quad -\frac{\int \frac{108(3d-10e)+288(2d-3e)-(-36(3d-10e)+72(3d-4e))x}{2+3x+x^2}\,dx}{5184} \\
&= -\frac{5d-6e+(3d-4e)x}{12\left(2+3x+x^2\right)}-\frac{1}{36}(7d-13e)\int \frac{1}{1+x}\,dx-\frac{1}{144}(-d-2e)\int \frac{1}{-2+x}\,dx \\
&\quad -\frac{1}{36}(d+e)\int \frac{1}{-1+x}\,dx-\frac{1}{144}(-31d+50e)\int \frac{1}{2+x}\,dx \\
&= -\frac{5d-6e+(3d-4e)x}{12\left(2+3x+x^2\right)}-\frac{1}{36}(d+e)\log(1-x)+\frac{1}{144}(d+2e)\log(2-x) \\
&\quad -\frac{1}{36}(7d-13e)\log(1+x)+\frac{1}{144}(31d-50e)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 80, normalized size of antiderivative = 0.90

$$\int \frac{(d+ex)\left(2-3x+x^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{1}{144}\left(\frac{12(-5d+6e-3dx+4ex)}{2+3x+x^2}-4(d+e)\log(1-x) + (d+2e)\log(2-x)+4(-7d+13e)\log(1+x) + (31d-50e)\log(2+x)\right)$$

```
[In] Integrate[((d + e*x)*(2 - 3*x + x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(-5*d + 6*e - 3*d*x + 4*e*x))/(2 + 3*x + x^2) - 4*(d + e)*Log[1 - x] +
 (d + 2*e)*Log[2 - x] + 4*(-7*d + 13*e)*Log[1 + x] + (31*d - 50*e)*Log[2 +
x])/144
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 78, normalized size of antiderivative = 0.88

method	result
default	$-\frac{\frac{d}{12}-\frac{e}{6}}{x+2}+\left(\frac{31d}{144}-\frac{25e}{72}\right)\ln\left(x+2\right)+\left(-\frac{7d}{36}+\frac{13e}{36}\right)\ln\left(x+1\right)-\frac{\frac{d}{6}-\frac{e}{6}}{x+1}+\left(-\frac{d}{36}-\frac{e}{36}\right)\ln\left(x-1\right)+\left(\frac{}{1}\right.$
risch	$\frac{\left(-\frac{d}{4}+\frac{e}{3}\right)x-\frac{5d}{12}+\frac{e}{2}}{x^2+3x+2}+\frac{\ln(2-x)d}{144}+\frac{\ln(2-x)e}{72}+\frac{31\ln(x+2)d}{144}-\frac{25\ln(x+2)e}{72}-\frac{\ln(x-1)d}{36}-\frac{\ln(x-1)e}{36}-\frac{7\ln(-x-1)d}{36}$
norman	$\frac{\left(-\frac{d}{4}+\frac{e}{3}\right)x^3+\left(\frac{3d}{4}-\frac{5e}{6}\right)x+\left(\frac{d}{3}-\frac{e}{2}\right)x^2-\frac{5d}{6}+e}{x^4-5x^2+4}+\left(-\frac{7d}{36}+\frac{13e}{36}\right)\ln\left(x+1\right)+\left(-\frac{d}{36}-\frac{e}{36}\right)\ln\left(x-1\right)+\left(\frac{d}{144}+\right.$
parallelrisch	$\frac{-60d+72e-36dx+2\ln(x-2)d+4\ln(x-2)e-8\ln(x-1)d-8\ln(x-1)e-150\ln(x+2)xe+6\ln(x-2)xe-12\ln(x-1)xd-12\ln(x-1)x}{}$

[In] int((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)

[Out] -(1/12*d-1/6*e)/(x+2)+(31/144*d-25/72*e)*ln(x+2)+(-7/36*d+13/36*e)*ln(x+1)-(1/6*d-1/6*e)/(x+1)+(-1/36*d-1/36*e)*ln(x-1)+(1/144*d+1/72*e)*ln(x-2)

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 153, normalized size of antiderivative = 1.72

$$\int \frac{(d+ex)\left(2-3x+x^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{12\,(3\,d-4\,e)x-((31\,d-50\,e)x^2+3\,(31\,d-50\,e)x+62\,d-100\,e)\log\left(x+2\right)+4\,((7\,d-13\,e)x^2+3\,(7}{}$$

[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="fricas")

[Out] -1/144*(12*(3*d - 4*e)*x - ((31*d - 50*e)*x^2 + 3*(31*d - 50*e)*x + 62*d - 100*e)*log(x + 2) + 4*((7*d - 13*e)*x^2 + 3*(7*d - 13*e)*x + 14*d - 26*e)*log(x + 1) + 4*((d + e)*x^2 + 3*(d + e)*x + 2*d + 2*e)*log(x - 1) - ((d + 2*e)*x^2 + 3*(d + 2*e)*x + 2*d + 4*e)*log(x - 2) + 60*d - 72*e)/(x^2 + 3*x + 2)

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1255 vs. 2(80) = 160.

Time = 6.98 (sec) , antiderivative size = 1255, normalized size of antiderivative = 14.10

$$\int \frac{(d+ex)(2-3x+x^2)}{(4-5x^2+x^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((e*x+d)*(x**2-3*x+2)/(x**4-5*x**2+4)**2,x)
[Out] -(d + e)*log(x + (-24383100*d**6 + 187408066*d**5*e + 10439775*d**5*(d + e)
 - 511591980*d**4*e**2 - 94132290*d**4*e*(d + e) + 667200*d**4*(d + e)**2 +
 469491120*d**3*e**3 + 333672552*d**3*e**2*(d + e) - 2703328*d**3*e*(d + e)
**2 - 198000*d**3*(d + e)**3 + 322778400*d**2*e**4 - 582497712*d**2*e**3*(d
 + e) + 1752768*d**2*e**2*(d + e)**2 + 1107552*d**2*e*(d + e)**3 - 86349385
6*d*e**5 + 500776560*d*e**4*(d + e) + 4226944*d*e**3*(d + e)**2 - 1880640*d
*e**2*(d + e)**3 + 429000000*e**6 - 169242912*e**5*(d + e) - 4538112*e**4*(
d + e)**2 + 964224*e**3*(d + e)**3)/(13474125*d**6 - 102860175*d**5*e + 274
190390*d**4*e**2 - 224142072*d**3*e**3 - 245084096*d**2*e**4 + 535797456*d*
e**5 - 256183200*e**6))/36 + (d + 2*e)*log(x + (-24383100*d**6 + 187408066*
d**5*e - 10439775*d**5*(d + 2*e)/4 - 511591980*d**4*e**2 + 47066145*d**4*e*
(d + 2*e)/2 + 41700*d**4*(d + 2*e)**2 + 469491120*d**3*e**3 - 83418138*d**3
*e**2*(d + 2*e) - 168958*d**3*e*(d + 2*e)**2 + 12375*d**3*(d + 2*e)**3/4 +
322778400*d**2*e**4 + 145624428*d**2*e**3*(d + 2*e) + 109548*d**2*e**2*(d +
 2*e)**2 - 34611*d**2*e*(d + 2*e)**3/2 - 863493856*d*e**5 - 125194140*d*e**
4*(d + 2*e) + 264184*d*e**3*(d + 2*e)**2 + 29385*d*e**2*(d + 2*e)**3 + 4290
00000*e**6 + 42310728*e**5*(d + 2*e) - 283632*e**4*(d + 2*e)**2 - 15066*e**
3*(d + 2*e)**3)/(13474125*d**6 - 102860175*d**5*e + 274190390*d**4*e**2 - 2
24142072*d**3*e**3 - 245084096*d**2*e**4 + 535797456*d*e**5 - 256183200*e**
6))/144 - (7*d - 13*e)*log(x + (-24383100*d**6 + 187408066*d**5*e + 1043977
5*d**5*(7*d - 13*e) - 511591980*d**4*e**2 - 94132290*d**4*e*(7*d - 13*e) +
667200*d**4*(7*d - 13*e)**2 + 469491120*d**3*e**3 + 333672552*d**3*e**2*(7*
d - 13*e) - 2703328*d**3*e*(7*d - 13*e)**2 - 198000*d**3*(7*d - 13*e)**3 +
322778400*d**2*e**4 - 582497712*d**2*e**3*(7*d - 13*e) + 1752768*d**2*e**2*
(7*d - 13*e)**2 + 1107552*d**2*e*(7*d - 13*e)**3 - 863493856*d*e**5 + 50077
6560*d*e**4*(7*d - 13*e) + 4226944*d*e**3*(7*d - 13*e)**2 - 1880640*d*e**2*
(7*d - 13*e)**3 + 429000000*e**6 - 169242912*e**5*(7*d - 13*e) - 4538112*e*
*4*(7*d - 13*e)**2 + 964224*e**3*(7*d - 13*e)**3)/(13474125*d**6 - 10286017
5*d**5*e + 274190390*d**4*e**2 - 224142072*d**3*e**3 - 245084096*d**2*e**4
+ 535797456*d*e**5 - 256183200*e**6))/36 + (31*d - 50*e)*log(x + (-24383100
*d**6 + 187408066*d**5*e - 10439775*d**5*(31*d - 50*e)/4 - 511591980*d**4*e
**2 + 47066145*d**4*e*(31*d - 50*e)/2 + 41700*d**4*(31*d - 50*e)**2 + 46949
1120*d**3*e**3 - 83418138*d**3*e**2*(31*d - 50*e) - 168958*d**3*e*(31*d - 5
0*e)**2 + 12375*d**3*(31*d - 50*e)**3/4 + 322778400*d**2*e**4 + 145624428*d
**2*e**3*(31*d - 50*e) + 109548*d**2*e**2*(31*d - 50*e)**2 - 34611*d**2*e*(
```

```
31*d - 50*e)**3/2 - 863493856*d*e**5 - 125194140*d*e**4*(31*d - 50*e) + 264
184*d*e**3*(31*d - 50*e)**2 + 29385*d*e**2*(31*d - 50*e)**3 + 429000000*e**
6 + 42310728*e**5*(31*d - 50*e) - 283632*e**4*(31*d - 50*e)**2 - 15066*e**3
*(31*d - 50*e)**3)/(13474125*d**6 - 102860175*d**5*e + 274190390*d**4*e**2
- 224142072*d**3*e**3 - 245084096*d**2*e**4 + 535797456*d*e**5 - 256183200*
e**6))/144 + (-5*d + 6*e + x*(-3*d + 4*e))/(12*x**2 + 36*x + 24)
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 75, normalized size of antiderivative = 0.84

$$\int \frac{(d+ex)(2-3x+x^2)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(31\,d-50\,e)\log\left(x+2\right) - \frac{1}{36}\,(7\,d-13\,e)\log\left(x+1\right) - \frac{1}{36}\,(d+e)\log\left(x-1\right) + \frac{1}{144}\,(d+2\,e)\log\left(x-2\right) - \frac{(3\,d-4\,e)x+5\,d-6\,e}{12\,(x^2+3\,x+2)}$$

```
[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/144*(31*d - 50*e)*log(x + 2) - 1/36*(7*d - 13*e)*log(x + 1) - 1/36*(d + e
)*log(x - 1) + 1/144*(d + 2*e)*log(x - 2) - 1/12*((3*d - 4*e)*x + 5*d - 6*e
)/(x^2 + 3*x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 79, normalized size of antiderivative = 0.89

$$\int \frac{(d+ex)(2-3x+x^2)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(31\,d-50\,e)\log\left(|x+2|\right) - \frac{1}{36}\,(7\,d-13\,e)\log\left(|x+1|\right) - \frac{1}{36}\,(d+e)\log\left(|x-1|\right) + \frac{1}{144}\,(d+2\,e)\log\left(|x-2|\right) - \frac{(3\,d-4\,e)x+5\,d-6\,e}{12\,(x+2)(x+1)}$$

```
[In] integrate((e*x+d)*(x^2-3*x+2)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/144*(31*d - 50*e)*log(abs(x + 2)) - 1/36*(7*d - 13*e)*log(abs(x + 1)) - 1
/36*(d + e)*log(abs(x - 1)) + 1/144*(d + 2*e)*log(abs(x - 2)) - 1/12*((3*d
- 4*e)*x + 5*d - 6*e)/((x + 2)*(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 79, normalized size of antiderivative = 0.89

$$\int \frac{(d+ex)(2-3x+x^2)}{(4-5x^2+x^4)^2}\,dx = \ln(x-2)\left(\frac{d}{144}+\frac{e}{72}\right)-\ln(x-1)\left(\frac{d}{36}+\frac{e}{36}\right) - \ln(x+1)\left(\frac{7\,d}{36}-\frac{13\,e}{36}\right)-\frac{\frac{5\,d}{12}-\frac{e}{2}+x\left(\frac{d}{4}-\frac{e}{3}\right)}{x^2+3\,x+2} + \ln(x+2)\left(\frac{31\,d}{144}-\frac{25\,e}{72}\right)$$

```
[In] int(((d + e*x)*(x^2 - 3*x + 2))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 2)*(d/144 + e/72) - log(x - 1)*(d/36 + e/36) - log(x + 1)*((7*d)/36
 - (13*e)/36) - ((5*d)/12 - e/2 + x*(d/4 - e/3))/(3*x + x^2 + 2) + log(x +
2)*((31*d)/144 - (25*e)/72)
```

3.93 $\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 105

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = -\frac{5d-6e+8f+(3d-4e+6f)x}{12(2+3x+x^2)} - \frac{1}{36}(d+e+f)\log(1-x) + \frac{1}{144}(d+2e+4f)\log(2-x) - \frac{1}{36}(7d-13e+19f)\log(1+x) + \frac{1}{144}(31d-50e+76f)\log(2+x)$$

```
[Out] 1/12*(-5*d+6*e-8*f-(3*d-4*e+6*f)*x)/(x^2+3*x+2)-1/36*(d+e+f)*ln(1-x)+1/144*
(d+2*e+4*f)*ln(2-x)-1/36*(7*d-13*e+19*f)*ln(1+x)+1/144*(31*d-50*e+76*f)*ln(
2+x)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 105, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.161, Rules used = {1600, 1074, 1086, 646, 31}

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = -\frac{x(3d-4e+6f)+5d-6e+8f}{12(x^2+3x+2)} - \frac{1}{36}\log(1-x)(d+e+f) + \frac{1}{144}\log(2-x)(d+2e+4f) - \frac{1}{36}\log(x+1)(7d-13e+19f) + \frac{1}{144}\log(x+2)(31d-50e+76f)$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] -1/12*(5*d - 6*e + 8*f + (3*d - 4*e + 6*f)*x)/(2 + 3*x + x^2) - ((d + e + f
)*Log[1 - x])/36 + ((d + 2*e + 4*f)*Log[2 - x])/144 - ((7*d - 13*e + 19*f)*
Log[1 + x])/36 + ((31*d - 50*e + 76*f)*Log[2 + x])/144
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 646

```
Int[((d_.) + (e_.)*(x_))/((a_) + (b_.)*(x_) + (c_.)*(x_)^2), x_Symbol] :> W
ith[{q = Rt[b^2 - 4*a*c, 2]}, Dist[(c*d - e*(b/2 - q/2))/q, Int[1/(b/2 - q/
2 + c*x), x], x] - Dist[(c*d - e*(b/2 + q/2))/q, Int[1/(b/2 + q/2 + c*x), x
], x]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a
*c, 0] && NiceSqrtQ[b^2 - 4*a*c]
```

Rule 1074

```
Int[((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_)*((A_.) + (B_.)*(x_) + (C_.)*(x_
)^2)*((d_) + (e_.)*(x_) + (f_.)*(x_)^2)^(q_), x_Symbol] :> Simp[(a + b*x +
c*x^2)^(p + 1)*((d + e*x + f*x^2)^(q + 1)/((b^2 - 4*a*c)*((c*d - a*f)^2 - (
b*d - a*e)*(c*e - b*f))*(p + 1)))*((A*c - a*C)*(2*a*c*e - b*(c*d + a*f)) +
(A*b - a*B)*(2*c^2*d + b^2*f - c*(b*e + 2*a*f)) + c*(A*(2*c^2*d + b^2*f - c
*(b*e + 2*a*f)) - B*(b*c*d - 2*a*c*e + a*b*f) + C*(b^2*d - a*b*e - 2*a*(c*d
 - a*f)))*x), x] + Dist[1/((b^2 - 4*a*c)*((c*d - a*f)^2 - (b*d - a*e)*(c*e
- b*f))*(p + 1)), Int[(a + b*x + c*x^2)^(p + 1)*(d + e*x + f*x^2)^q*Simp[(b
*B - 2*A*c - 2*a*C)*((c*d - a*f)^2 - (b*d - a*e)*(c*e - b*f))*(p + 1) + (b^
2*(C*d + A*f) - b*(B*c*d + A*c*e + a*C*e + a*B*f) + 2*(A*c*(c*d - a*f) - a*
(c*C*d - B*c*e - a*C*f)))*(a*f*(p + 1) - c*d*(p + 2)) - e*((A*c - a*C)*(2*a
*c*e - b*(c*d + a*f)) + (A*b - a*B)*(2*c^2*d + b^2*f - c*(b*e + 2*a*f)))*(p
 + q + 2) - (2*f*((A*c - a*C)*(2*a*c*e - b*(c*d + a*f)) + (A*b - a*B)*(2*c^
2*d + b^2*f - c*(b*e + 2*a*f))))*(p + q + 2) - (b^2*(C*d + A*f) - b*(B*c*d +
 A*c*e + a*C*e + a*B*f) + 2*(A*c*(c*d - a*f) - a*(c*C*d - B*c*e - a*C*f)))*
(b*f*(p + 1) - c*e*(2*p + q + 4)))*x - c*f*(b^2*(C*d + A*f) - b*(B*c*d + A*
c*e + a*C*e + a*B*f) + 2*(A*c*(c*d - a*f) - a*(c*C*d - B*c*e - a*C*f)))*(2*
p + 2*q + 5)*x^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, q}, x] &&
 NeQ[b^2 - 4*a*c, 0] && NeQ[e^2 - 4*d*f, 0] && LtQ[p, -1] && NeQ[(c*d - a*f
)^2 - (b*d - a*e)*(c*e - b*f), 0] &&  !( !IntegerQ[p] && ILtQ[q, -1]) &&  !
IGtQ[q, 0]
```

Rule 1086

```
Int[((A_.) + (B_.)*(x_) + (C_.)*(x_)^2)/(((a_) + (b_.)*(x_) + (c_.)*(x_)^2)
*((d_) + (e_.)*(x_) + (f_.)*(x_)^2)), x_Symbol] :> With[{q = c^2*d^2 - b*c*
```

```
d*e + a*c*e^2 + b^2*d*f - 2*a*c*d*f - a*b*e*f + a^2*f^2}, Dist[1/q, Int[(A*
c^2*d - a*c*C*d - A*b*c*e + a*B*c*e + A*b^2*f - a*b*B*f - a*A*c*f + a^2*C*f
 + c*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B*f)*x)/(a + b*x + c*x^2),
x], x] + Dist[1/q, Int[(c*C*d^2 - B*c*d*e + A*c*e^2 + b*B*d*f - A*c*d*f - a
*C*d*f - A*b*e*f + a*A*f^2 - f*(B*c*d - b*C*d - A*c*e + a*C*e + A*b*f - a*B
*f)*x)/(d + e*x + f*x^2), x], x] /; NeQ[q, 0]] /; FreeQ[{a, b, c, d, e, f,
A, B, C}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[e^2 - 4*d*f, 0]
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{(2-3x+x^2)\,(2+3x+x^2)^2}\,dx\\
&= -\frac{5d-6e+8f+(3d-4e+6f)x}{12\,(2+3x+x^2)}\\
&\quad -\frac{1}{72}\int \frac{6(3d-10e+12f)-24(2d-3e+5f)x+6(3d-4e+6f)x^2}{(2-3x+x^2)\,(2+3x+x^2)}\,dx\\
&= -\frac{5d-6e+8f+(3d-4e+6f)x}{12\,(2+3x+x^2)}\\
&\quad -\frac{\int \frac{-288(2d-3e+5f)+108(3d-10e+12f)+(72(3d-4e+6f)-36(3d-10e+12f))x}{2-3x+x^2}\,dx}{5184}\\
&\quad -\frac{\int \frac{288(2d-3e+5f)+108(3d-10e+12f)-(72(3d-4e+6f)-36(3d-10e+12f))x}{2+3x+x^2}\,dx}{5184}\\
&= -\frac{5d-6e+8f+(3d-4e+6f)x}{12\,(2+3x+x^2)}-\frac{1}{144}(-31d+50e-76f)\int \frac{1}{2+x}\,dx\\
&\quad -\frac{1}{144}(-d-2e-4f)\int \frac{1}{-2+x}\,dx\\
&\quad -\frac{1}{36}(d+e+f)\int \frac{1}{-1+x}\,dx-\frac{1}{36}(7d-13e+19f)\int \frac{1}{1+x}\,dx\\
&= -\frac{5d-6e+8f+(3d-4e+6f)x}{12\,(2+3x+x^2)}-\frac{1}{36}(d+e+f)\log(1-x)+\frac{1}{144}(d+2e\\
&\quad +4f)\log(2-x)-\frac{1}{36}(7d-13e+19f)\log(1+x)+\frac{1}{144}(31d-50e+76f)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 97, normalized size of antiderivative = 0.92

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(-\frac{12(-6e+8f-4ex+6fx+d(5+3x))}{2+3x+x^2} - 4(d+e+f)\log(1-x)+(d+2e+4f)\log(2-x) - 4(7d-13e+19f)\log(1+x) + (31d-50e+76f)\log(2+x)\right)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((-12*(-6*e + 8*f - 4*e*x + 6*f*x + d*(5 + 3*x)))/(2 + 3*x + x^2) - 4*(d +
e + f)*Log[1 - x] + (d + 2*e + 4*f)*Log[2 - x] - 4*(7*d - 13*e + 19*f)*Log[
1 + x] + (31*d - 50*e + 76*f)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 96, normalized size of antiderivative = 0.91

method	result
default	$-\frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}}{x+2}+\left(\frac{31d}{144}-\frac{25e}{72}+\frac{19f}{36}\right)\ln\left(x+2\right)+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}\right)\ln\left(x+1\right)-\frac{\frac{d}{6}-\frac{e}{6}+\frac{f}{6}}{x+1}+\left(-\frac{d}{36}-\right.$
norman	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}\right)x^3+\left(\frac{3d}{4}-\frac{5e}{6}+f\right)x+\left(\frac{d}{3}-\frac{e}{2}+\frac{5f}{6}\right)x^2-\frac{5d}{6}+e-\frac{4f}{3}}{x^4-5x^2+4}+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}\right)\ln\left(x+1\right)+\left(-\frac{d}{36}-\frac{e}{36}-\right.$
risch	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}\right)x-\frac{5d}{12}+\frac{e}{2}-\frac{2f}{3}}{x^2+3x+2}-\frac{\ln(x-1)d}{36}-\frac{\ln(x-1)e}{36}-\frac{\ln(x-1)f}{36}+\frac{31\ln(x+2)d}{144}-\frac{25\ln(x+2)e}{72}+\frac{19\ln(x+2)f}{36}-$
parallelrisch	$\underline{-96f-60d+72e-36dx+2\ln(x-2)d+4\ln(x-2)e-8\ln(x-1)d-8\ln(x-1)e+152\ln(x+2)f-152\ln(x+1)f-150\ln(x+2)xe+6\ln}$

```
[In] int((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] -(1/12*d-1/6*e+1/3*f)/(x+2)+(31/144*d-25/72*e+19/36*f)*ln(x+2)+(-7/36*d+13/
36*e-19/36*f)*ln(x+1)-(1/6*d-1/6*e+1/6*f)/(x+1)+(-1/36*d-1/36*e-1/36*f)*ln(
x-1)+(1/144*d+1/72*e+1/36*f)*ln(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 191 vs. $2(95) = 190$.

Time = 0.34 (sec) , antiderivative size = 191, normalized size of antiderivative = 1.82

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(3\,d-4\,e+6\,f)x-((31\,d-50\,e+76\,f)x^2+3\,(31\,d-50\,e+76\,f)x+62\,d-100\,e+152\,f)\log\left(x+\right.}{}$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/144*(12*(3*d - 4*e + 6*f)*x - ((31*d - 50*e + 76*f)*x^2 + 3*(31*d - 50*e
 + 76*f)*x + 62*d - 100*e + 152*f)*log(x + 2) + 4*((7*d - 13*e + 19*f)*x^2
+ 3*(7*d - 13*e + 19*f)*x + 14*d - 26*e + 38*f)*log(x + 1) + 4*((d + e + f)
*x^2 + 3*(d + e + f)*x + 2*d + 2*e + 2*f)*log(x - 1) - ((d + 2*e + 4*f)*x^2
 + 3*(d + 2*e + 4*f)*x + 2*d + 4*e + 8*f)*log(x - 2) + 60*d - 72*e + 96*f)/
(x^2 + 3*x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**2-3*x+2)*(f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 91, normalized size of antiderivative = 0.87

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = {} & \frac{1}{144}\,(31\,d-50\,e+76\,f)\log\left(x+2\right) \\ & - \frac{1}{36}\,(7\,d-13\,e+19\,f)\log\left(x+1\right) \\ & - \frac{1}{36}\,(d+e+f)\log\left(x-1\right) \\ & + \frac{1}{144}\,(d+2\,e+4\,f)\log\left(x-2\right) \\ & - \frac{(3\,d-4\,e+6\,f)x+5\,d-6\,e+8\,f}{12\,(x^2+3\,x+2)}\end{aligned}$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/144*(31*d - 50*e + 76*f)*log(x + 2) - 1/36*(7*d - 13*e + 19*f)*log(x + 1)
 - 1/36*(d + e + f)*log(x - 1) + 1/144*(d + 2*e + 4*f)*log(x - 2) - 1/12*((
3*d - 4*e + 6*f)*x + 5*d - 6*e + 8*f)/(x^2 + 3*x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 95, normalized size of antiderivative = 0.90

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(31\,d-50\,e+76\,f)\log\left(|x+2|\right) - \frac{1}{36}\,(7\,d-13\,e+19\,f)\log\left(|x+1|\right) - \frac{1}{36}\,(d+e+f)\log\left(|x-1|\right) + \frac{1}{144}\,(d+2\,e+4\,f)\log\left(|x-2|\right) - \frac{(3\,d-4\,e+6\,f)x+5\,d-6\,e+8\,f}{12\,(x+2)(x+1)}$$

```
[In] integrate((x^2-3*x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/144*(31*d - 50*e + 76*f)*log(abs(x + 2)) - 1/36*(7*d - 13*e + 19*f)*log(a
bs(x + 1)) - 1/36*(d + e + f)*log(abs(x - 1)) + 1/144*(d + 2*e + 4*f)*log(a
bs(x - 2)) - 1/12*((3*d - 4*e + 6*f)*x + 5*d - 6*e + 8*f)/((x + 2)*(x + 1))
```

Mupad [B] (verification not implemented)

Time = 0.08 (sec) , antiderivative size = 97, normalized size of antiderivative = 0.92

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = \ln(x-2)\,\left(\frac{d}{144}+\frac{e}{72}+\frac{f}{36}\right) - \ln(x+1)\,\left(\frac{7\,d}{36}-\frac{13\,e}{36}+\frac{19\,f}{36}\right) - \ln(x-1)\,\left(\frac{d}{36}+\frac{e}{36}+\frac{f}{36}\right) + \ln(x+2)\,\left(\frac{31\,d}{144}-\frac{25\,e}{72}+\frac{19\,f}{36}\right) - \frac{\frac{5\,d}{12}-\frac{e}{2}+\frac{2\,f}{3}+x\,\left(\frac{d}{4}-\frac{e}{3}+\frac{f}{2}\right)}{x^2+3\,x+2}$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 2)*(d/144 + e/72 + f/36) - log(x + 1)*((7*d)/36 - (13*e)/36 + (19*f
)/36) - log(x - 1)*(d/36 + e/36 + f/36) + log(x + 2)*((31*d)/144 - (25*e)/7
2 + (19*f)/36) - ((5*d)/12 - e/2 + (2*f)/3 + x*(d/4 - e/3 + f/2))/(3*x + x^
2 + 2)
```

3.94 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 117

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{d-e+f-g}{6(1+x)}-\frac{d-2e+4f-8g}{12(2+x)} -\frac{1}{36}(d+e+f+g)\log(1-x) +\frac{1}{144}(d+2e+4f+8g)\log(2-x) -\frac{1}{36}(7d-13e+19f-25g)\log(1+x) +\frac{1}{144}(31d-50e+76f-104g)\log(2+x)$$

```
[Out] 1/6*(-d+e-f+g)/(1+x)+1/12*(-d+2*e-4*f+8*g)/(2+x)-1/36*(d+e+f+g)*ln(1-x)+1/1
44*(d+2*e+4*f+8*g)*ln(2-x)-1/36*(7*d-13*e+19*f-25*g)*ln(1+x)+1/144*(31*d-50
*e+76*f-104*g)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 117, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.056, Rules used

= {1600, 6860}

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{d-2e+4f-8g}{12(x+2)} - \frac{d-e+f-g}{6(x+1)} - \frac{1}{36}\log(1-x)(d+e+f+g) + \frac{1}{144}\log(2-x)(d+2e+4f+8g) - \frac{1}{36}\log(x+1)(7d-13e+19f-25g) + \frac{1}{144}\log(x+2)(31d-50e+76f-104g)$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] -1/6*(d - e + f - g)/(1 + x) - (d - 2*e + 4*f - 8*g)/(12*(2 + x)) - ((d + e
 + f + g)*Log[1 - x])/36 + ((d + 2*e + 4*f + 8*g)*Log[2 - x])/144 - ((7*d -
 13*e + 19*f - 25*g)*Log[1 + x])/36 + ((31*d - 50*e + 76*f - 104*g)*Log[2 +
 x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6860

```
Int[(u_)/((a_.) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{v = RationalFunctionExpand[u/(a + b*x^n + c*x^(2*n)), x]}, Int[v, x] /; Su
mQ[v]] /; FreeQ[{a, b, c}, x] && EqQ[n2, 2*n] && IGtQ[n, 0]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2+gx^3}{(2-3x+x^2)(2+3x+x^2)^2}\,dx$$

$$= \int \left(\frac{d+2e+4f+8g}{144(-2+x)} + \frac{-d-e-f-g}{36(-1+x)} + \frac{d-e+f-g}{6(1+x)^2} + \frac{-7d+13e-19f+25g}{36(1+x)} + \frac{d-2e+4f-8g}{12(2+x)^2} + \frac{31d-50e+76f-104g}{144(2+x)}\right) dx$$

$$= -\frac{d-e+f-g}{6(1+x)} - \frac{d-2e+4f-8g}{12(2+x)} - \frac{1}{36}(d+e+f+g)\log(1-x) + \frac{1}{144}(d+2e+4f+8g)\log(2-x) - \frac{1}{36}(7d-13e+19f-25g)\log(1+x) + \frac{1}{144}(31d-50e+76f-104g)\log(2+x)$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 114, normalized size of antiderivative = 0.97

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\left(\frac{12(-5d+6e-8f+12g-3dx+4ex-6fx+10gx)}{2+3x+x^2} - 4(d+e+f+g)\log(1-x) + (d+2e+4f+8g)\log(2-x) + 4(-7d+13e-19f+25g)\log(1+x) + (31d-50e+76f-104g)\log(2+x)\right)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4)^2,x
]
```

```
[Out] ((12*(-5*d + 6*e - 8*f + 12*g - 3*d*x + 4*e*x - 6*f*x + 10*g*x))/(2 + 3*x +
 x^2) - 4*(d + e + f + g)*Log[1 - x] + (d + 2*e + 4*f + 8*g)*Log[2 - x] + 4
*(-7*d + 13*e - 19*f + 25*g)*Log[1 + x] + (31*d - 50*e + 76*f - 104*g)*Log[
2 + x])/144
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 114, normalized size of antiderivative = 0.97

method	result
default	$-\frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}}{x+2}+\left(\frac{31d}{144}-\frac{25e}{72}+\frac{19f}{36}-\frac{13g}{18}\right)\ln\left(x+2\right)+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}+\frac{25g}{36}\right)\ln\left(x+1\right)-$
norman	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}+\frac{5g}{6}\right)x^3+\left(\frac{3d}{4}-\frac{5e}{6}+f-\frac{4g}{3}\right)x+\left(\frac{d}{3}-\frac{e}{2}+\frac{5f}{6}-\frac{3g}{2}\right)x^2-\frac{5d}{6}+e+2g-\frac{4f}{3}}{x^4-5x^2+4}+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}+\frac{25g}{36}\right)\ln(x$
risch	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}+\frac{5g}{6}\right)x-\frac{5d}{12}+\frac{e}{2}-\frac{2f}{3}+g}{x^2+3x+2}-\frac{\ln(x-1)d}{36}-\frac{\ln(x-1)e}{36}-\frac{\ln(x-1)f}{36}-\frac{\ln(x-1)g}{36}-\frac{7\ln(-x-1)d}{36}+\frac{13\ln(-x-1}{36}$
parallelrisch	$-96f+144g+120gx-60d+72e-36dx+2\ln(x-2)d+4\ln(x-2)e-8\ln(x-1)d-8\ln(x-1)e+24\ln(x-2)xg-12\ln(x-1)xg+300\ln$

```
[In] int((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE
)
```

```
[Out] -(1/12*d-1/6*e+1/3*f-2/3*g)/(x+2)+(31/144*d-25/72*e+19/36*f-13/18*g)*ln(x+2
)+(-7/36*d+13/36*e-19/36*f+25/36*g)*ln(x+1)-(1/6*d-1/6*e+1/6*f-1/6*g)/(x+1)
+(-1/36*d-1/36*e-1/36*f-1/36*g)*ln(x-1)+(1/144*d+1/72*e+1/36*f+1/18*g)*ln(x
-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 229 vs. $2(105) = 210$.

Time = 0.72 (sec) , antiderivative size = 229, normalized size of antiderivative = 1.96

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(3\,d-4\,e+6\,f-10\,g)x-((31\,d-50\,e+76\,f-104\,g)x^2+3\,(31\,d-50\,e+76\,f-104\,g)x+62\,d-}{}$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fri
cas")
```

```
[Out] -1/144*(12*(3*d - 4*e + 6*f - 10*g)*x - ((31*d - 50*e + 76*f - 104*g)*x^2 +
 3*(31*d - 50*e + 76*f - 104*g)*x + 62*d - 100*e + 152*f - 208*g)*log(x + 2
) + 4*((7*d - 13*e + 19*f - 25*g)*x^2 + 3*(7*d - 13*e + 19*f - 25*g)*x + 14
*d - 26*e + 38*f - 50*g)*log(x + 1) + 4*((d + e + f + g)*x^2 + 3*(d + e + f
 + g)*x + 2*d + 2*e + 2*f + 2*g)*log(x - 1) - ((d + 2*e + 4*f + 8*g)*x^2 +
3*(d + 2*e + 4*f + 8*g)*x + 2*d + 4*e + 8*f + 16*g)*log(x - 2) + 60*d - 72*
e + 96*f - 144*g)/(x^2 + 3*x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**2-3*x+2)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 107, normalized size of antiderivative = 0.91

$$\begin{aligned}\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{144}\,(31\,d-50\,e+76\,f-104\,g)\log{(x+2)} - \frac{1}{36}\,(7\,d-13\,e+19\,f-25\,g)\log{(x+1)} \\ &\quad - \frac{1}{36}\,(d+e+f+g)\log{(x-1)} + \frac{1}{144}\,(d+2\,e+4\,f+8\,g)\log{(x-2)} \\ &\quad - \frac{(3\,d-4\,e+6\,f-10\,g)x+5\,d-6\,e+8\,f-12\,g}{12\,(x^2+3\,x+2)}\end{aligned}$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="max
ima")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g)*log(x + 2) - 1/36*(7*d - 13*e + 19*f - 2
5*g)*log(x + 1) - 1/36*(d + e + f + g)*log(x - 1) + 1/144*(d + 2*e + 4*f +
8*g)*log(x - 2) - 1/12*((3*d - 4*e + 6*f - 10*g)*x + 5*d - 6*e + 8*f - 12*g
)/(x^2 + 3*x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 111, normalized size of antiderivative = 0.95

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}(31\,d-50\,e+76\,f-104\,g)\log(|x+2|)-\frac{1}{36}(7\,d-13\,e+19\,f-25\,g)\log(|x+1|) - \frac{1}{36}(d+e+f+g)\log(|x-1|)+\frac{1}{144}(d+2\,e+4\,f+8\,g)\log(|x-2|) - \frac{(3\,d-4\,e+6\,f-10\,g)x+5\,d-6\,e+8\,f-12\,g}{12\,(x+2)(x+1)}$$

```
[In] integrate((x^2-3*x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="gia
c")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g)*log(abs(x + 2)) - 1/36*(7*d - 13*e + 19*
f - 25*g)*log(abs(x + 1)) - 1/36*(d + e + f + g)*log(abs(x - 1)) + 1/144*(d
 + 2*e + 4*f + 8*g)*log(abs(x - 2)) - 1/12*((3*d - 4*e + 6*f - 10*g)*x + 5*
d - 6*e + 8*f - 12*g)/((x + 2)*(x + 1))
```

Mupad [B] (verification not implemented)

Time = 7.98 (sec) , antiderivative size = 115, normalized size of antiderivative = 0.98

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \ln(x-2)\left(\frac{d}{144}+\frac{e}{72}+\frac{f}{36}+\frac{g}{18}\right) - \ln(x+1)\left(\frac{7\,d}{36}-\frac{13\,e}{36}+\frac{19\,f}{36}-\frac{25\,g}{36}\right) - \ln(x-1)\left(\frac{d}{36}+\frac{e}{36}+\frac{f}{36}+\frac{g}{36}\right) + \ln(x+2)\left(\frac{31\,d}{144}-\frac{25\,e}{72}+\frac{19\,f}{36}-\frac{13\,g}{18}\right) - \frac{\frac{5\,d}{12}-\frac{e}{2}+\frac{2\,f}{3}-g+x\left(\frac{d}{4}-\frac{e}{3}+\frac{f}{2}-\frac{5\,g}{6}\right)}{x^2+3\,x+2}$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 2)*(d/144 + e/72 + f/36 + g/18) - log(x + 1)*((7*d)/36 - (13*e)/36
+ (19*f)/36 - (25*g)/36) - log(x - 1)*(d/36 + e/36 + f/36 + g/36) + log(x +
 2)*((31*d)/144 - (25*e)/72 + (19*f)/36 - (13*g)/18) - ((5*d)/12 - e/2 + (2
*f)/3 - g + x*(d/4 - e/3 + f/2 - (5*g)/6))/(3*x + x^2 + 2)
```

3.95 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 41, antiderivative size = 131

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx &= -\frac{d-e+f-g+h}{6(1+x)}-\frac{d-2e+4f-8g+16h}{12(2+x)}-\frac{1}{36}(d+e+f+g+h)\log(1-x)\\ &\quad+\frac{1}{144}(d+2e+4f+8g+16h)\log(2-x)-\frac{1}{36}(7d-13e+19f-25g+31h)\log(1+x)\\ &\quad+\frac{1}{144}(31d-50e+76f-104g+112h)\log(2+x)\end{aligned}$$

```
[Out] 1/6*(-d+e-f+g-h)/(1+x)+1/12*(-d+2*e-4*f+8*g-16*h)/(2+x)-1/36*(d+e+f+g+h)*ln
(1-x)+1/144*(d+2*e+4*f+8*g+16*h)*ln(2-x)-1/36*(7*d-13*e+19*f-25*g+31*h)*ln(
1+x)+1/144*(31*d-50*e+76*f-104*g+112*h)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 131, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.049, Rules used = {1600, 6860}

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx &= -\frac{d-e+f-g+h}{6(x+1)}-\frac{d-2e+4f-8g+16h}{12(x+2)}-\frac{1}{36}\log(1-x)(d+e+f+g+h)\\ &\quad+\frac{1}{144}\log(2-x)(d+2e+4f+8g+16h)-\frac{1}{36}\log(x+1)(7d-13e+19f-25g+31h)\\ &\quad+\frac{1}{144}\log(x+2)(31d-50e+76f-104g+112h)\end{aligned}$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4)^2
,x]
```

```
[Out] -1/6*(d - e + f - g + h)/(1 + x) - (d - 2*e + 4*f - 8*g + 16*h)/(12*(2 + x)
) - ((d + e + f + g + h)*Log[1 - x])/36 + ((d + 2*e + 4*f + 8*g + 16*h)*Log
[2 - x])/144 - ((7*d - 13*e + 19*f - 25*g + 31*h)*Log[1 + x])/36 + ((31*d -
 50*e + 76*f - 104*g + 112*h)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6860

```
Int[(u_)/((a_.) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{v = RationalFunctionExpand[u/(a + b*x^n + c*x^(2*n)), x]}, Int[v, x] /; Su
mQ[v]] /; FreeQ[{a, b, c}, x] && EqQ[n2, 2*n] && IGtQ[n, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3+hx^4}{(2-3x+x^2)\,(2+3x+x^2)^2}\,dx \\
&= \int \left(\frac{d+2e+4f+8g+16h}{144(-2+x)}+\frac{-d-e-f-g-h}{36(-1+x)}+\frac{d-e+f-g+h}{6(1+x)^2}\right. \\
&\qquad +\frac{-7d+13e-19f+25g-31h}{36(1+x)}+\frac{d-2e+4f-8g+16h}{12(2+x)^2} \\
&\qquad \left.+\frac{31d-50e+76f-104g+112h}{144(2+x)}\right)dx \\
&= -\frac{d-e+f-g+h}{6(1+x)}-\frac{d-2e+4f-8g+16h}{12(2+x)}-\frac{1}{36}(d+e+f+g+h)\log(1-x) \\
&\quad +\frac{1}{144}(d+2e+4f+8g+16h)\log(2-x)-\frac{1}{36}(7d-13e+19f-25g+31h)\log(1+x) \\
&\quad +\frac{1}{144}(31d-50e+76f-104g+112h)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 136, normalized size of antiderivative = 1.04

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx$$
$$= \frac{1}{144}\left(-\frac{12(d(5+3x)+2(4f-6g+10h+3fx-5gx+9hx-e(3+2x)))}{2+3x+x^2} - 4(d+e+f+g+h)\log(1-x)+(d+2(e+2f+4g+8h))\log(2-x) - 4(7d-13e+19f-25g+31h)\log(1+x)+(31d-50e+76f-104g+112h)\log(2+x)\right)$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + 
x^4)^2,x]
```

```
[Out] ((-12*(d*(5 + 3*x) + 2*(4*f - 6*g + 10*h + 3*f*x - 5*g*x + 9*h*x - e*(3 + 2
*x))))/(2 + 3*x + x^2) - 4*(d + e + f + g + h)*Log[1 - x] + (d + 2*(e + 2*f
 + 4*g + 8*h))*Log[2 - x] - 4*(7*d - 13*e + 19*f - 25*g + 31*h)*Log[1 + x]
+ (31*d - 50*e + 76*f - 104*g + 112*h)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 132, normalized size of antiderivative = 1.01

method	result
default	$-\frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}+\frac{4h}{3}}{x+2}+\left(\frac{31d}{144}-\frac{25e}{72}+\frac{19f}{36}-\frac{13g}{18}+\frac{7h}{9}\right)\ln\left(x+2\right)+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}+\frac{25g}{36}-\frac{31h}{36}\right)$
norman	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}+\frac{5g}{6}-\frac{3h}{2}\right)x^3+\left(\frac{3d}{4}-\frac{5e}{6}+f-\frac{4g}{3}+2h\right)x+\left(\frac{d}{3}-\frac{e}{2}+\frac{5f}{6}-\frac{3g}{2}+\frac{17h}{6}\right)x^2-\frac{5d}{6}+e+2g-\frac{10h}{3}-\frac{4f}{3}}{x^4-5x^2+4}+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{1}{3}\right.$
risch	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}+\frac{5g}{6}-\frac{3h}{2}\right)x-\frac{5d}{12}+\frac{e}{2}-\frac{2f}{3}+g-\frac{5h}{3}}{x^2+3x+2}+\frac{31\ln(x+2)d}{144}-\frac{25\ln(x+2)e}{72}+\frac{19\ln(x+2)f}{36}-\frac{13\ln(x+2)g}{18}+\frac{7\ln(x+2}{9}$
parallelrisch	$\frac{-96f+144g+120gx-60d-240h+72e+48\ln(x-2)xh-12\ln(x-1)xh-372\ln(x+1)xh+336\ln(x+2)xh-36dx+2\ln(x-2)d+4\ln}{}$

```
[In] int((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNV
ERBOSE)
```

```
[Out] -(1/12*d-1/6*e+1/3*f-2/3*g+4/3*h)/(x+2)+(31/144*d-25/72*e+19/36*f-13/18*g+7
/9*h)*ln(x+2)+(-7/36*d+13/36*e-19/36*f+25/36*g-31/36*h)*ln(x+1)-(1/6*d-1/6*
e+1/6*f-1/6*g+1/6*h)/(x+1)+(-1/36*d-1/36*e-1/36*f-1/36*g-1/36*h)*ln(x-1)+(1
/144*d+1/72*e+1/36*f+1/18*g+1/9*h)*ln(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 267 vs. 2(119) = 238.

Time = 3.52 (sec) , antiderivative size = 267, normalized size of antiderivative = 2.04

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx =$$
$$-\frac{12\,(3\,d-4\,e+6\,f-10\,g+18\,h)x-((31\,d-50\,e+76\,f-104\,g+112\,h)x^2+3\,(31\,d-50\,e+76\,f-1}{}$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="fricas")
```

```
[Out] -1/144*(12*(3*d - 4*e + 6*f - 10*g + 18*h)*x - ((31*d - 50*e + 76*f - 104*g
 + 112*h)*x^2 + 3*(31*d - 50*e + 76*f - 104*g + 112*h)*x + 62*d - 100*e + 1
52*f - 208*g + 224*h)*log(x + 2) + 4*((7*d - 13*e + 19*f - 25*g + 31*h)*x^2
 + 3*(7*d - 13*e + 19*f - 25*g + 31*h)*x + 14*d - 26*e + 38*f - 50*g + 62*h
)*log(x + 1) + 4*((d + e + f + g + h)*x^2 + 3*(d + e + f + g + h)*x + 2*d +
 2*e + 2*f + 2*g + 2*h)*log(x - 1) - ((d + 2*e + 4*f + 8*g + 16*h)*x^2 + 3*
(d + 2*e + 4*f + 8*g + 16*h)*x + 2*d + 4*e + 8*f + 16*g + 32*h)*log(x - 2)
+ 60*d - 72*e + 96*f - 144*g + 240*h)/(x^2 + 3*x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**2-3*x+2)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 123, normalized size of antiderivative = 0.94

$$\begin{aligned}
&\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{1}{144}\,(31\,d-50\,e+76\,f-104\,g+112\,h)\log(x+2) \\
&\quad - \frac{1}{36}\,(7\,d-13\,e+19\,f-25\,g+31\,h)\log(x+1) \\
&\quad - \frac{1}{36}\,(d+e+f+g+h)\log(x-1) + \frac{1}{144}\,(d+2\,e+4\,f+8\,g+16\,h)\log(x-2) \\
&\quad - \frac{(3\,d-4\,e+6\,f-10\,g+18\,h)x+5\,d-6\,e+8\,f-12\,g+20\,h}{12\,(x^2+3\,x+2)}
\end{aligned}$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="maxima")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g + 112*h)*log(x + 2) - 1/36*(7*d - 13*e +
19*f - 25*g + 31*h)*log(x + 1) - 1/36*(d + e + f + g + h)*log(x - 1) + 1/14
4*(d + 2*e + 4*f + 8*g + 16*h)*log(x - 2) - 1/12*((3*d - 4*e + 6*f - 10*g +
 18*h)*x + 5*d - 6*e + 8*f - 12*g + 20*h)/(x^2 + 3*x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 127, normalized size of antiderivative = 0.97

$$
\begin{aligned}
&\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{1}{144}\,(31\,d-50\,e+76\,f-104\,g+112\,h)\log\left(|x+2|\right) \\
&\quad -\frac{1}{36}\,(7\,d-13\,e+19\,f-25\,g+31\,h)\log\left(|x+1|\right) \\
&\quad -\frac{1}{36}\,(d+e+f+g+h)\log\left(|x-1|\right)+\frac{1}{144}\,(d+2\,e+4\,f+8\,g+16\,h)\log\left(|x-2|\right) \\
&\quad -\frac{(3\,d-4\,e+6\,f-10\,g+18\,h)x+5\,d-6\,e+8\,f-12\,g+20\,h}{12\,(x+2)(x+1)}
\end{aligned}
$$

```
[In] integrate((x^2-3*x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="giac")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g + 112*h)*log(abs(x + 2)) - 1/36*(7*d - 13
*e + 19*f - 25*g + 31*h)*log(abs(x + 1)) - 1/36*(d + e + f + g + h)*log(abs
(x - 1)) + 1/144*(d + 2*e + 4*f + 8*g + 16*h)*log(abs(x - 2)) - 1/12*((3*d
- 4*e + 6*f - 10*g + 18*h)*x + 5*d - 6*e + 8*f - 12*g + 20*h)/((x + 2)*(x +
 1))
```

Mupad [B] (verification not implemented)

Time = 8.24 (sec) , antiderivative size = 133, normalized size of antiderivative = 1.02

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx$$
$$= \ln(x-2)\,\left(\frac{d}{144}+\frac{e}{72}+\frac{f}{36}+\frac{g}{18}+\frac{h}{9}\right)-\ln(x-1)\,\left(\frac{d}{36}+\frac{e}{36}+\frac{f}{36}+\frac{g}{36}+\frac{h}{36}\right)$$
$$-\ln(x+1)\,\left(\frac{7\,d}{36}-\frac{13\,e}{36}+\frac{19\,f}{36}-\frac{25\,g}{36}+\frac{31\,h}{36}\right)$$
$$-\frac{\frac{5\,d}{12}-\frac{e}{2}+\frac{2\,f}{3}-g+\frac{5\,h}{3}+x\,\left(\frac{d}{4}-\frac{e}{3}+\frac{f}{2}-\frac{5\,g}{6}+\frac{3\,h}{2}\right)}{x^2+3\,x+2}$$
$$+\ln(x+2)\,\left(\frac{31\,d}{144}-\frac{25\,e}{72}+\frac{19\,f}{36}-\frac{13\,g}{18}+\frac{7\,h}{9}\right)$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2 + 4)^2
,x)
```

```
[Out] log(x - 2)*(d/144 + e/72 + f/36 + g/18 + h/9) - log(x - 1)*(d/36 + e/36 + f
/36 + g/36 + h/36) - log(x + 1)*((7*d)/36 - (13*e)/36 + (19*f)/36 - (25*g)/
36 + (31*h)/36) - ((5*d)/12 - e/2 + (2*f)/3 - g + (5*h)/3 + x*(d/4 - e/3 +
f/2 - (5*g)/6 + (3*h)/2))/(3*x + x^2 + 2) + log(x + 2)*((31*d)/144 - (25*e)
/72 + (19*f)/36 - (13*g)/18 + (7*h)/9)
```

3.96 $\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 46, antiderivative size = 147

$$\begin{aligned}\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx &= -\frac{d-e+f-g+h-i}{6(1+x)}-\frac{d-2e+4f-8g+16h-32i}{12(2+x)} \\ &\quad -\frac{1}{36}(d+e+f+g+h+i)\log(1-x)+\frac{1}{144}(d+2e+4f+8g+16h+32i)\log(2-x) \\ &\quad -\frac{1}{36}(7d-13e+19f-25g+31h-37i)\log(1+x) \\ &\quad +\frac{1}{144}(31d-50e+76f-104g+112h-32i)\log(2+x)\end{aligned}$$

```
[Out] 1/6*(-d+e-f+g-h+i)/(1+x)+1/12*(-d+2*e-4*f+8*g-16*h+32*i)/(2+x)-1/36*(d+e+f+
g+h+i)*ln(1-x)+1/144*(d+2*e+4*f+8*g+16*h+32*i)*ln(2-x)-1/36*(7*d-13*e+19*f-
25*g+31*h-37*i)*ln(1+x)+1/144*(31*d-50*e+76*f-104*g+112*h-32*i)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 147, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.043, Rules used

$= \{1600, 6860\}$

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$$
$$= -\frac{d-2e+4f-8g+16h-32i}{12(x+2)} - \frac{d-e+f-g+h-i}{6(x+1)}$$
$$- \frac{1}{36}\log(1-x)(d+e+f+g+h+i) + \frac{1}{144}\log(2-x)(d+2e+4f+8g+16h+32i)$$
$$- \frac{1}{36}\log(x+1)(7d-13e+19f-25g+31h-37i)$$
$$+ \frac{1}{144}\log(x+2)(31d-50e+76f-104g+112h-32i)$$

```
[In] Int[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2
+ x^4)^2,x]
```

```
[Out] -1/6*(d - e + f - g + h - i)/(1 + x) - (d - 2*e + 4*f - 8*g + 16*h - 32*i)/
(12*(2 + x)) - ((d + e + f + g + h + i)*Log[1 - x])/36 + ((d + 2*e + 4*f +
8*g + 16*h + 32*i)*Log[2 - x])/144 - ((7*d - 13*e + 19*f - 25*g + 31*h - 37
*i)*Log[1 + x])/36 + ((31*d - 50*e + 76*f - 104*g + 112*h - 32*i)*Log[2 + x
])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6860

```
Int[(u_)/((a_.) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{v = RationalFunctionExpand[u/(a + b*x^n + c*x^(2*n)), x]}, Int[v, x] /; Su
mQ[v]] /; FreeQ[{a, b, c}, x] && EqQ[n2, 2*n] && IGtQ[n, 0]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(2-3x+x^2)\,(2+3x+x^2)^2}\,dx$$
$$= \int \left(\frac{d+2e+4f+8g+16h+32i}{144(-2+x)} + \frac{-d-e-f-g-h-i}{36(-1+x)} + \frac{d-e+f-g+h-i}{6(1+x)^2} + \frac{-7d+13e-19f+25g-31h+37i}{36(1+x)} + \frac{d-2e+4f-8g+16h-32i}{12(2+x)^2} + \frac{31d-50e+76f-104g+112h-32i}{144(2+x)}\right) dx$$

$$
\begin{aligned}
&= -\frac{d-e+f-g+h-i}{6(1+x)} - \frac{d-2e+4f-8g+16h-32i}{12(2+x)} \\
&\quad - \frac{1}{36}(d+e+f+g+h+i)\log(1-x) + \frac{1}{144}(d+2e+4f+8g+16h+32i)\log(2-x) \\
&\quad - \frac{1}{36}(7d-13e+19f-25g+31h-37i)\log(1+x) \\
&\quad + \frac{1}{144}(31d-50e+76f-104g+112h-32i)\log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 153, normalized size of antiderivative = 1.04

$$
\begin{aligned}
&\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx \\
&= \frac{1}{144}\left(\frac{12(-d(5+3x)+2(-4f+6g-10h+18i-3fx+5gx-9hx+17ix+e(3+2x)))}{2+3x+x^2}\right. \\
&\quad -4(d+e+f+g+h+i)\log(1-x)+(d+2e+4(f+2g+4h+8i))\log(2-x) \\
&\quad +4(-7d+13e-19f+25g-31h+37i)\log(1+x) \\
&\quad \left.+(31d-50e+76f-104g+112h-32i)\log(2+x)\right)
\end{aligned}
$$

```
[In] Integrate[((2 - 3*x + x^2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 -
5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(-(d*(5 + 3*x)) + 2*(-4*f + 6*g - 10*h + 18*i - 3*f*x + 5*g*x - 9*h*x
+ 17*i*x + e*(3 + 2*x))))/(2 + 3*x + x^2) - 4*(d + e + f + g + h + i)*Log[1
 - x] + (d + 2*e + 4*(f + 2*g + 4*h + 8*i))*Log[2 - x] + 4*(-7*d + 13*e - 1
9*f + 25*g - 31*h + 37*i)*Log[1 + x] + (31*d - 50*e + 76*f - 104*g + 112*h
- 32*i)*Log[2 + x])/144
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 150, normalized size of antiderivative = 1.02

method	result
default	$-\frac{\frac{d}{12}-\frac{e}{6}+\frac{f}{3}-\frac{2g}{3}+\frac{4h}{3}-\frac{8i}{3}}{x+2}+\left(\frac{31d}{144}-\frac{25e}{72}+\frac{19f}{36}-\frac{13g}{18}-\frac{2i}{9}+\frac{7h}{9}\right)\ln\left(x+2\right)+\left(-\frac{7d}{36}+\frac{13e}{36}-\frac{19f}{36}+\frac{25g}{36}\right.$
norman	$\frac{\left(-\frac{d}{4}+\frac{e}{3}-\frac{f}{2}+\frac{5g}{6}-\frac{3h}{2}+\frac{17i}{6}\right)x^3+\left(\frac{3d}{4}-\frac{5e}{6}+f-\frac{4g}{3}+2h-\frac{10i}{3}\right)x+\left(\frac{d}{3}-\frac{e}{2}+\frac{5f}{6}-\frac{3g}{2}+\frac{17h}{6}-\frac{11i}{2}\right)x^2+6i-\frac{5d}{6}+e+2g-\frac{10h}{3}-\frac{4f}{3}}{x^4-5x^2+4}+($
risch	$\frac{25\ln(-x-1)g}{36}-\frac{\ln(x-1)d}{36}-\frac{\ln(x-1)e}{36}+\frac{19\ln(x+2)f}{36}-\frac{31\ln(-x-1)h}{36}+\frac{\ln(2-x)f}{36}+\frac{\ln(2-x)d}{144}+\frac{\ln(2-x)e}{72}+3$
parallelrisch	$\underline{432i-96f+144g-96\ln(x+2)xi+96\ln(x-2)xi+120gx-60d-240h+72e+48\ln(x-2)xh-12\ln(x-1)xh-372\ln(x+1)xh+336\ln}$

```
[In] int((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_R
ETURNVERBOSE)
```

```
[Out] -(1/12*d-1/6*e+1/3*f-2/3*g+4/3*h-8/3*i)/(x+2)+(31/144*d-25/72*e+19/36*f-13/
18*g-2/9*i+7/9*h)*ln(x+2)+(-7/36*d+13/36*e-19/36*f+25/36*g-31/36*h+37/36*i)
*ln(x+1)-(1/6*d-1/6*e+1/6*f-1/6*g+1/6*h-1/6*i)/(x+1)+(-1/36*d-1/36*e-1/36*f
-1/36*g-1/36*h-1/36*i)*ln(x-1)+(1/144*d+1/72*e+1/36*f+1/18*g+1/9*h+2/9*i)*l
n(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 305 vs. 2(135) = 270.

Time = 21.46 (sec) , antiderivative size = 305, normalized size of antiderivative = 2.07

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx =$$
$$-\frac{12\,(3\,d-4\,e+6\,f-10\,g+18\,h-34\,i)x-((31\,d-50\,e+76\,f-104\,g+112\,h-32\,i)x^2+3\,(31\,d-50$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, al
gorithm="fricas")
```

```
[Out] -1/144*(12*(3*d - 4*e + 6*f - 10*g + 18*h - 34*i)*x - ((31*d - 50*e + 76*f
- 104*g + 112*h - 32*i)*x^2 + 3*(31*d - 50*e + 76*f - 104*g + 112*h - 32*i)
*x + 62*d - 100*e + 152*f - 208*g + 224*h - 64*i)*log(x + 2) + 4*((7*d - 13
*e + 19*f - 25*g + 31*h - 37*i)*x^2 + 3*(7*d - 13*e + 19*f - 25*g + 31*h -
37*i)*x + 14*d - 26*e + 38*f - 50*g + 62*h - 74*i)*log(x + 1) + 4*((d + e +
 f + g + h + i)*x^2 + 3*(d + e + f + g + h + i)*x + 2*d + 2*e + 2*f + 2*g +
 2*h + 2*i)*log(x - 1) - ((d + 2*e + 4*f + 8*g + 16*h + 32*i)*x^2 + 3*(d +
2*e + 4*f + 8*g + 16*h + 32*i)*x + 2*d + 4*e + 8*f + 16*g + 32*h + 64*i)*lo
g(x - 2) + 60*d - 72*e + 96*f - 144*g + 240*h - 432*i)/(x^2 + 3*x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((x**2-3*x+2)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)*
*2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 139, normalized size of antiderivative = 0.95

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}(31\,d-50\,e+76\,f-104\,g+112\,h-32\,i)\log(x+2) - \frac{1}{36}(7\,d-13\,e+19\,f-25\,g+31\,h-37\,i)\log(x+1) - \frac{1}{36}(d+e+f+g+h+i)\log(x-1) + \frac{1}{144}(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log(x-2) - \frac{(3\,d-4\,e+6\,f-10\,g+18\,h-34\,i)x+5\,d-6\,e+8\,f-12\,g+20\,h-36\,i}{12\,(x^2+3\,x+2)}$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, al
gorithm="maxima")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g + 112*h - 32*i)*log(x + 2) - 1/36*(7*d -
13*e + 19*f - 25*g + 31*h - 37*i)*log(x + 1) - 1/36*(d + e + f + g + h + i)
*log(x - 1) + 1/144*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(x - 2) - 1/12*(
(3*d - 4*e + 6*f - 10*g + 18*h - 34*i)*x + 5*d - 6*e + 8*f - 12*g + 20*h -
36*i)/(x^2 + 3*x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 143, normalized size of antiderivative = 0.97

$$\int \frac{(2-3x+x^2)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}(31\,d-50\,e+76\,f-104\,g+112\,h-32\,i)\log(|x+2|) - \frac{1}{36}(7\,d-13\,e+19\,f-25\,g+31\,h-37\,i)\log(|x+1|) - \frac{1}{36}(d+e+f+g+h+i)\log(|x-1|) + \frac{1}{144}(d+2\,e+4\,f+8\,g+16\,h+32\,i)\log(|x-2|) - \frac{(3\,d-4\,e+6\,f-10\,g+18\,h-34\,i)x+5\,d-6\,e+8\,f-12\,g+20\,h-36\,i}{12\,(x+2)(x+1)}$$

```
[In] integrate((x^2-3*x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, al
gorithm="giac")
```

```
[Out] 1/144*(31*d - 50*e + 76*f - 104*g + 112*h - 32*i)*log(abs(x + 2)) - 1/36*(7
*d - 13*e + 19*f - 25*g + 31*h - 37*i)*log(abs(x + 1)) - 1/36*(d + e + f +
g + h + i)*log(abs(x - 1)) + 1/144*(d + 2*e + 4*f + 8*g + 16*h + 32*i)*log(
abs(x - 2)) - 1/12*((3*d - 4*e + 6*f - 10*g + 18*h - 34*i)*x + 5*d - 6*e +
8*f - 12*g + 20*h - 36*i)/((x + 2)*(x + 1))
```

Mupad [B] (verification not implemented)

Time = 8.67 (sec) , antiderivative size = 151, normalized size of antiderivative = 1.03

$$\int \frac{(2-3x+x^2)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$$
$$=\ln(x-2)\,\left(\frac{d}{144}+\frac{e}{72}+\frac{f}{36}+\frac{g}{18}+\frac{h}{9}+\frac{2\,i}{9}\right)-\ln(x-1)\,\left(\frac{d}{36}+\frac{e}{36}+\frac{f}{36}+\frac{g}{36}+\frac{h}{36}+\frac{i}{36}\right)$$
$$-\ln(x+1)\,\left(\frac{7\,d}{36}-\frac{13\,e}{36}+\frac{19\,f}{36}-\frac{25\,g}{36}+\frac{31\,h}{36}-\frac{37\,i}{36}\right)$$
$$+\ln(x+2)\,\left(\frac{31\,d}{144}-\frac{25\,e}{72}+\frac{19\,f}{36}-\frac{13\,g}{18}+\frac{7\,h}{9}-\frac{2\,i}{9}\right)$$
$$-\frac{\frac{5\,d}{12}-\frac{e}{2}+\frac{2\,f}{3}-g+\frac{5\,h}{3}-3\,i+x\,\left(\frac{d}{4}-\frac{e}{3}+\frac{f}{2}-\frac{5\,g}{6}+\frac{3\,h}{2}-\frac{17\,i}{6}\right)}{x^2+3\,x+2}$$

```
[In] int(((x^2 - 3*x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4 - 5*x^
2 + 4)^2,x)
```

```
[Out] log(x - 2)*(d/144 + e/72 + f/36 + g/18 + h/9 + (2*i)/9) - log(x - 1)*(d/36
+ e/36 + f/36 + g/36 + h/36 + i/36) - log(x + 1)*((7*d)/36 - (13*e)/36 + (1
9*f)/36 - (25*g)/36 + (31*h)/36 - (37*i)/36) + log(x + 2)*((31*d)/144 - (25
*e)/72 + (19*f)/36 - (13*g)/18 + (7*h)/9 - (2*i)/9) - ((5*d)/12 - e/2 + (2*
f)/3 - g + (5*h)/3 - 3*i + x*(d/4 - e/3 + f/2 - (5*g)/6 + (3*h)/2 - (17*i)/
6))/(3*x + x^2 + 2)
```

3.97 $\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 16, antiderivative size = 68

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12(1-x)}+\frac{1}{36(2-x)}-\frac{1}{36(1+x)}+\frac{1}{18}\log(1-x) - \frac{35}{432}\log(2-x)+\frac{1}{54}\log(1+x)+\frac{1}{144}\log(2+x)$$

```
[Out] 1/12/(1-x)+1/36/(2-x)-1/36/(1+x)+1/18*ln(1-x)-35/432*ln(2-x)+1/54*ln(1+x)+1
/144*ln(2+x)
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 68, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.125$, Rules used = {1600, 2099}

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = \frac{1}{12(1-x)}+\frac{1}{36(2-x)}-\frac{1}{36(x+1)}+\frac{1}{18}\log(1-x) - \frac{35}{432}\log(2-x)+\frac{1}{54}\log(x+1)+\frac{1}{144}\log(x+2)$$

```
[In] Int[(2 + x)/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] 1/(12*(1 - x)) + 1/(36*(2 - x)) - 1/(36*(1 + x)) + Log[1 - x]/18 - (35*Log[
2 - x])/432 + Log[1 + x]/54 + Log[2 + x]/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
```

```
EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 2099

```
Int[(P_)^(p_)*(Q_)^(q_.), x_Symbol] :> With[{PP = Factor[P]}, Int[ExpandInt
egrand[PP^p*Q^q, x], x] /;  !SumQ[NonfreeFactors[PP, x]]] /; FreeQ[q, x] &&
 PolyQ[P, x] && PolyQ[Q, x] && IntegerQ[p] && NeQ[P, x]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{1}{(2+x)\left(2-x-2x^2+x^3\right)^2}\,dx \\
&= \int \left(\frac{1}{36(-2+x)^2}-\frac{35}{432(-2+x)}+\frac{1}{12(-1+x)^2}+\frac{1}{18(-1+x)}+\frac{1}{36(1+x)^2}\right. \\
&\qquad \left.+\frac{1}{54(1+x)}+\frac{1}{144(2+x)}\right)dx \\
&= \frac{1}{12(1-x)}+\frac{1}{36(2-x)}-\frac{1}{36(1+x)}+\frac{1}{18}\log(1-x) \\
&\quad -\frac{35}{432}\log(2-x)+\frac{1}{54}\log(1+x)+\frac{1}{144}\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.02 (sec) , antiderivative size = 60, normalized size of antiderivative = 0.88

$$\int \frac{2+x}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{1}{432}\left(\frac{12(5+6x-5x^2)}{2-x-2x^2+x^3}+24\log(1-x)-35\log(2-x)\right. \\
\left.+8\log(1+x)+3\log(2+x)\right)$$

```
[In] Integrate[(2 + x)/(4 - 5*x^2 + x^4)^2,x]
[Out] ((12*(5 + 6*x - 5*x^2))/(2 - x - 2*x^2 + x^3) + 24*Log[1 - x] - 35*Log[2 -
x] + 8*Log[1 + x] + 3*Log[2 + x])/432
```

Maple [A] (verified)

Time = 0.07 (sec) , antiderivative size = 47, normalized size of antiderivative = 0.69

method	result
default	$\frac{\ln(x+2)}{144}-\frac{1}{36(x+1)}+\frac{\ln(x+1)}{54}-\frac{1}{12(x-1)}+\frac{\ln(x-1)}{18}-\frac{1}{36(x-2)}-\frac{35\ln(x-2)}{432}$
risch	$\frac{-\frac{5}{36}x^2+\frac{1}{6}x+\frac{5}{36}}{x^3-2x^2-x+2}-\frac{35\ln(x-2)}{432}+\frac{\ln(x-1)}{18}+\frac{\ln(x+1)}{54}+\frac{\ln(x+2)}{144}$
norman	$\frac{-\frac{1}{9}x^2+\frac{17}{36}x-\frac{5}{36}x^3+\frac{5}{18}}{x^4-5x^2+4}-\frac{35\ln(x-2)}{432}+\frac{\ln(x-1)}{18}+\frac{\ln(x+1)}{54}+\frac{\ln(x+2)}{144}$
parallelrisch	$-\frac{35\ln(x-2)x^3-24\ln(x-1)x^3-8\ln(x+1)x^3-3\ln(x+2)x^3-60-70\ln(x-2)x^2+48\ln(x-1)x^2+16\ln(x+1)x^2+6\ln(x+2)x^2-3}{432(x^3-2x^2-}$

[In] int((x+2)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)

[Out] 1/144*ln(x+2)-1/36/(x+1)+1/54*ln(x+1)-1/12/(x-1)+1/18*ln(x-1)-1/36/(x-2)-35 /432*ln(x-2)

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 103 vs. $2(50) = 100$.

Time = 0.25 (sec) , antiderivative size = 103, normalized size of antiderivative = 1.51

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = -\frac{60\,x^2-3\,(x^3-2\,x^2-x+2)\log(x+2)-8\,(x^3-2\,x^2-x+2)\log(x+1)-24\,(x^3-2\,x^2-x+2)\log}{432\,(x^3-2\,x^2-x+2)}$$

[In] integrate((2+x)/(x^4-5*x^2+4)^2,x, algorithm="fricas")

[Out] -1/432*(60*x^2 - 3*(x^3 - 2*x^2 - x + 2)*log(x + 2) - 8*(x^3 - 2*x^2 - x + 2)*log(x + 1) - 24*(x^3 - 2*x^2 - x + 2)*log(x - 1) + 35*(x^3 - 2*x^2 - x + 2)*log(x - 2) - 72*x - 60)/(x^3 - 2*x^2 - x + 2)

Sympy [A] (verification not implemented)

Time = 0.16 (sec) , antiderivative size = 53, normalized size of antiderivative = 0.78

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = \frac{-5x^2+6x+5}{36x^3-72x^2-36x+72}-\frac{35\log(x-2)}{432}+\frac{\log(x-1)}{18}+\frac{\log(x+1)}{54}+\frac{\log(x+2)}{144}$$

[In] integrate((2+x)/(x**4-5*x**2+4)**2,x)

[Out] (-5*x**2 + 6*x + 5)/(36*x**3 - 72*x**2 - 36*x + 72) - 35*log(x - 2)/432 + l og(x - 1)/18 + log(x + 1)/54 + log(x + 2)/144

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 52, normalized size of antiderivative = 0.76

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = -\frac{5\,x^2-6\,x-5}{36\,(x^3-2\,x^2-x+2)}+\frac{1}{144}\,\log\left(x+2\right)+\frac{1}{54}\,\log\left(x+1\right)+\frac{1}{18}\,\log\left(x-1\right)-\frac{35}{432}\,\log\left(x-2\right)$$

```
[In] integrate((2+x)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] -1/36*(5*x^2 - 6*x - 5)/(x^3 - 2*x^2 - x + 2) + 1/144*log(x + 2) + 1/54*log
(x + 1) + 1/18*log(x - 1) - 35/432*log(x - 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 56, normalized size of antiderivative = 0.82

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = -\frac{5\,x^2-6\,x-5}{36\,(x+1)(x-1)(x-2)}+\frac{1}{144}\,\log\left(|x+2|\right)+\frac{1}{54}\,\log\left(|x+1|\right)+\frac{1}{18}\,\log\left(|x-1|\right)-\frac{35}{432}\,\log\left(|x-2|\right)$$

```
[In] integrate((2+x)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] -1/36*(5*x^2 - 6*x - 5)/((x + 1)*(x - 1)*(x - 2)) + 1/144*log(abs(x + 2)) +
 1/54*log(abs(x + 1)) + 1/18*log(abs(x - 1)) - 35/432*log(abs(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.03 (sec) , antiderivative size = 52, normalized size of antiderivative = 0.76

$$\int \frac{2+x}{(4-5x^2+x^4)^2}\,dx = \frac{\ln\left(x-1\right)}{18}+\frac{\ln\left(x+1\right)}{54}-\frac{35\,\ln\left(x-2\right)}{432}+\frac{\ln\left(x+2\right)}{144}-\frac{-\frac{5\,x^2}{36}+\frac{x}{6}+\frac{5}{36}}{-x^3+2\,x^2+x-2}$$

```
[In] int((x + 2)/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)/18 + log(x + 1)/54 - (35*log(x - 2))/432 + log(x + 2)/144 - (x/6
 - (5*x^2)/36 + 5/36)/(x + 2*x^2 - x^3 - 2)
```

3.98 $\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 21, antiderivative size = 105

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = \frac{d+e}{12(1-x)}+\frac{d+2e}{36(2-x)}-\frac{d-e}{36(1+x)} + \frac{1}{36}(2d+5e)\log(1-x)-\frac{1}{432}(35d+58e)\log(2-x) + \frac{1}{108}(2d+e)\log(1+x)+\frac{1}{144}(d-2e)\log(2+x)$$

```
[Out] 1/12*(d+e)/(1-x)+1/36*(d+2*e)/(2-x)+1/36*(-d+e)/(1+x)+1/36*(2*d+5*e)*ln(1-x
)-1/432*(35*d+58*e)*ln(2-x)+1/108*(2*d+e)*ln(1+x)+1/144*(d-2*e)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.13 (sec) , antiderivative size = 105, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.095, Rules used = {1600, 6874}

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = -\frac{d-e}{36(x+1)}+\frac{d+e}{12(1-x)}+\frac{d+2e}{36(2-x)} + \frac{1}{36}(2d+5e)\log(1-x)-\frac{1}{432}(35d+58e)\log(2-x) + \frac{1}{108}(2d+e)\log(x+1)+\frac{1}{144}(d-2e)\log(x+2)$$

```
[In] Int[((2 + x)*(d + e*x))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d + e)/(12*(1 - x)) + (d + 2*e)/(36*(2 - x)) - (d - e)/(36*(1 + x)) + ((2*
d + 5*e)*Log[1 - x])/36 - ((35*d + 58*e)*Log[2 - x])/432 + ((2*d + e)*Log[1
 + x])/108 + ((d - 2*e)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{d+ex}{(2+x)\left(2-x-2x^2+x^3\right)^2}\,dx \\
&= \int \left(\frac{d+2e}{36(-2+x)^2}+\frac{-35d-58e}{432(-2+x)}+\frac{d+e}{12(-1+x)^2}+\frac{2d+5e}{36(-1+x)}+\frac{d-e}{36(1+x)^2}\right. \\
&\qquad \left.+\frac{2d+e}{108(1+x)}+\frac{d-2e}{144(2+x)}\right)dx \\
&= \frac{d+e}{12(1-x)}+\frac{d+2e}{36(2-x)}-\frac{d-e}{36(1+x)}+\frac{1}{36}(2d+5e)\log(1-x) \\
&\quad -\frac{1}{432}(35d+58e)\log(2-x)+\frac{1}{108}(2d+e)\log(1+x)+\frac{1}{144}(d-2e)\log(2+x)
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 97, normalized size of antiderivative = 0.92

$$
\int \frac{(2+x)(d+ex)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{1}{432}\left(\frac{12\left(d\left(5+6x-5x^2\right)+2e\left(5-2x^2\right)\right)}{2-x-2x^2+x^3}+12(2d+5e)\log(1-x)\right.
$$
$$
\left.-(35d+58e)\log(2-x)+4(2d+e)\log(1+x)+3(d-2e)\log(2+x)\right)
$$

```
[In] Integrate[((2 + x)*(d + e*x))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(d*(5 + 6*x - 5*x^2) + 2*e*(5 - 2*x^2)))/(2 - x - 2*x^2 + x^3) + 12*(2
*d + 5*e)*Log[1 - x] - (35*d + 58*e)*Log[2 - x] + 4*(2*d + e)*Log[1 + x] +
3*(d - 2*e)*Log[2 + x])/432
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.88

method	result
default	$\left(\frac{d}{144}-\frac{e}{72}\right)\ln\left(x+2\right)-\frac{\frac{d}{36}-\frac{e}{36}}{x+1}+\left(\frac{d}{54}+\frac{e}{108}\right)\ln\left(x+1\right)-\frac{\frac{d}{12}+\frac{e}{12}}{x-1}+\left(\frac{d}{18}+\frac{5e}{36}\right)\ln\left(x-1\right)+\left(-\frac{35}{43}\right.$
norman	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}\right)x^3+\left(\frac{17d}{36}+\frac{5e}{18}\right)x+\left(-\frac{d}{9}-\frac{2e}{9}\right)x^2+\frac{5d}{18}+\frac{5e}{9}}{x^4-5x^2+4}+\left(-\frac{35d}{432}-\frac{29e}{216}\right)\ln\left(x-2\right)+\left(\frac{d}{18}+\frac{5e}{36}\right)\ln\left(x-1\right)+\left(\right.$
risch	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}\right)x^2+\frac{dx}{6}+\frac{5d}{36}+\frac{5e}{18}}{x^3-2x^2-x+2}+\frac{\ln(x+2)d}{144}-\frac{\ln(x+2)e}{72}+\frac{\ln(x-1)d}{18}+\frac{5\ln(x-1)e}{36}-\frac{35\ln(2-x)d}{432}-\frac{29\ln(2-x)e}{216}+\ln($
parallelrisch	$-\frac{60d\,x^2-60d-120e-72dx-60\ln(x-1)x^3e-8\ln(x+1)x^3d-4\ln(x+1)x^3e-3\ln(x+2)x^3d+6\ln(x+2)x^3e+70\ln(x-2)d+116\ln}{}$

```
[In] int((x+2)*(e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (1/144*d-1/72*e)*ln(x+2)-(1/36*d-1/36*e)/(x+1)+(1/54*d+1/108*e)*ln(x+1)-(1/
12*d+1/12*e)/(x-1)+(1/18*d+5/36*e)*ln(x-1)+(-35/432*d-29/216*e)*ln(x-2)-(1/
36*d+1/18*e)/(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 211 vs. 2(87) = 174.

Time = 0.28 (sec) , antiderivative size = 211, normalized size of antiderivative = 2.01

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(5\,d+4\,e)x^2-72\,dx-3\,((d-2\,e)x^3-2\,(d-2\,e)x^2-(d-2\,e)x+2\,d-4\,e)\log\left(x+2\right)-4\,((2\,d+}{}$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/432*(12*(5*d + 4*e)*x^2 - 72*d*x - 3*((d - 2*e)*x^3 - 2*(d - 2*e)*x^2 -
(d - 2*e)*x + 2*d - 4*e)*log(x + 2) - 4*((2*d + e)*x^3 - 2*(2*d + e)*x^2 -
(2*d + e)*x + 4*d + 2*e)*log(x + 1) - 12*((2*d + 5*e)*x^3 - 2*(2*d + 5*e)*x
^2 - (2*d + 5*e)*x + 4*d + 10*e)*log(x - 1) + ((35*d + 58*e)*x^3 - 2*(35*d
+ 58*e)*x^2 - (35*d + 58*e)*x + 70*d + 116*e)*log(x - 2) - 60*d - 120*e)/(x
^3 - 2*x^2 - x + 2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1034 vs. $2(82) = 164$.

Time = 5.46 (sec) , antiderivative size = 1034, normalized size of antiderivative = 9.85

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = \text{Too large to display}$$

```
[In] integrate((2+x)*(e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] (d - 2*e)*log(x + (8710660*d**5 + 91884504*d**4*e - 7579779*d**4*(d - 2*e)/
4 + 364910432*d**3*e**2 - 18128055*d**3*e*(d - 2*e) - 83772*d**3*(d - 2*e)*
*2 + 686697536*d**2*e**3 - 60296868*d**2*e**2*(d - 2*e) - 597816*d**2*e*(d
- 2*e)**2 + 65907*d**2*(d - 2*e)**3/4 + 614357568*d*e**4 - 85949220*d*e**3*
(d - 2*e) - 1500048*d*e**2*(d - 2*e)**2 + 105840*d*e*(d - 2*e)**3 + 2084704
00*e**5 - 45136356*e**4*(d - 2*e) - 1196064*e**3*(d - 2*e)**2 + 128277*e**2
*(d - 2*e)**3)/(3374210*d**5 + 38645295*d**4*e + 170558380*d**3*e**2 + 3620
61760*d**2*e**3 + 370298160*d*e**4 + 146466320*e**5))/144 + (2*d + e)*log(x
 + (8710660*d**5 + 91884504*d**4*e - 2526593*d**4*(2*d + e) + 364910432*d**
3*e**2 - 24170740*d**3*e*(2*d + e) - 148928*d**3*(2*d + e)**2 + 686697536*d
**2*e**3 - 80395824*d**2*e**2*(2*d + e) - 1062784*d**2*e*(2*d + e)**2 + 390
56*d**2*(2*d + e)**3 + 614357568*d*e**4 - 114598960*d*e**3*(2*d + e) - 2666
752*d*e**2*(2*d + e)**2 + 250880*d*e*(2*d + e)**3 + 208470400*e**5 - 601818
08*e**4*(2*d + e) - 2126336*e**3*(2*d + e)**2 + 304064*e**2*(2*d + e)**3)/(
3374210*d**5 + 38645295*d**4*e + 170558380*d**3*e**2 + 362061760*d**2*e**3
+ 370298160*d*e**4 + 146466320*e**5))/108 + (2*d + 5*e)*log(x + (8710660*d*
*5 + 91884504*d**4*e - 7579779*d**4*(2*d + 5*e) + 364910432*d**3*e**2 - 725
12220*d**3*e*(2*d + 5*e) - 1340352*d**3*(2*d + 5*e)**2 + 686697536*d**2*e**
3 - 241187472*d**2*e**2*(2*d + 5*e) - 9565056*d**2*e*(2*d + 5*e)**2 + 10545
12*d**2*(2*d + 5*e)**3 + 614357568*d*e**4 - 343796880*d*e**3*(2*d + 5*e) -
24000768*d*e**2*(2*d + 5*e)**2 + 6773760*d*e*(2*d + 5*e)**3 + 208470400*e**
5 - 180545424*e**4*(2*d + 5*e) - 19137024*e**3*(2*d + 5*e)**2 + 8209728*e**
2*(2*d + 5*e)**3)/(3374210*d**5 + 38645295*d**4*e + 170558380*d**3*e**2 + 3
62061760*d**2*e**3 + 370298160*d*e**4 + 146466320*e**5))/36 - (35*d + 58*e)
*log(x + (8710660*d**5 + 91884504*d**4*e + 2526593*d**4*(35*d + 58*e)/4 + 3
64910432*d**3*e**2 + 6042685*d**3*e*(35*d + 58*e) - 9308*d**3*(35*d + 58*e)
**2 + 686697536*d**2*e**3 + 20098956*d**2*e**2*(35*d + 58*e) - 66424*d**2*e
*(35*d + 58*e)**2 - 2441*d**2*(35*d + 58*e)**3/4 + 614357568*d*e**4 + 28649
740*d*e**3*(35*d + 58*e) - 166672*d*e**2*(35*d + 58*e)**2 - 3920*d*e*(35*d
+ 58*e)**3 + 208470400*e**5 + 15045452*e**4*(35*d + 58*e) - 132896*e**3*(35
*d + 58*e)**2 - 4751*e**2*(35*d + 58*e)**3)/(3374210*d**5 + 38645295*d**4*e
 + 170558380*d**3*e**2 + 362061760*d**2*e**3 + 370298160*d*e**4 + 146466320
*e**5))/432 + (6*d*x + 5*d + 10*e + x**2*(-5*d - 4*e))/(36*x**3 - 72*x**2 -
 36*x + 72)
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 88, normalized size of antiderivative = 0.84

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(d-2\,e)\log\left(x+2\right)+\frac{1}{108}\,(2\,d+e)\log\left(x+1\right) + \frac{1}{36}\,(2\,d+5\,e)\log\left(x-1\right)-\frac{1}{432}\,(35\,d+58\,e)\log\left(x-2\right) - \frac{(5\,d+4\,e)x^2-6\,dx-5\,d-10\,e}{36\,(x^3-2\,x^2-x+2)}$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/144*(d - 2*e)*log(x + 2) + 1/108*(2*d + e)*log(x + 1) + 1/36*(2*d + 5*e)*
log(x - 1) - 1/432*(35*d + 58*e)*log(x - 2) - 1/36*((5*d + 4*e)*x^2 - 6*d*x
 - 5*d - 10*e)/(x^3 - 2*x^2 - x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 92, normalized size of antiderivative = 0.88

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(d-2\,e)\log\left(|x+2|\right)+\frac{1}{108}\,(2\,d+e)\log\left(|x+1|\right) + \frac{1}{36}\,(2\,d+5\,e)\log\left(|x-1|\right)-\frac{1}{432}\,(35\,d+58\,e)\log\left(|x-2|\right) - \frac{(5\,d+4\,e)x^2-6\,dx-5\,d-10\,e}{36\,(x+1)(x-1)(x-2)}$$

```
[In] integrate((2+x)*(e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/144*(d - 2*e)*log(abs(x + 2)) + 1/108*(2*d + e)*log(abs(x + 1)) + 1/36*(2
*d + 5*e)*log(abs(x - 1)) - 1/432*(35*d + 58*e)*log(abs(x - 2)) - 1/36*((5*
d + 4*e)*x^2 - 6*d*x - 5*d - 10*e)/((x + 1)*(x - 1)*(x - 2))
```

Mupad [B] (verification not implemented)

Time = 0.06 (sec) , antiderivative size = 90, normalized size of antiderivative = 0.86

$$\int \frac{(2+x)(d+ex)}{(4-5x^2+x^4)^2}\,dx = \ln(x-1)\left(\frac{d}{18}+\frac{5\,e}{36}\right)-\frac{\left(-\frac{5\,d}{36}-\frac{e}{9}\right)x^2+\frac{d\,x}{6}+\frac{5\,d}{36}+\frac{5\,e}{18}}{-x^3+2\,x^2+x-2} + \ln(x+1)\left(\frac{d}{54}+\frac{e}{108}\right)+\ln(x+2)\left(\frac{d}{144}-\frac{e}{72}\right) - \ln(x-2)\left(\frac{35\,d}{432}+\frac{29\,e}{216}\right)$$

```
[In] int(((x + 2)*(d + e*x))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/18 + (5*e)/36) - ((5*d)/36 + (5*e)/18 - x^2*((5*d)/36 + e/9)
+ (d*x)/6)/(x + 2*x^2 - x^3 - 2) + log(x + 1)*(d/54 + e/108) + log(x + 2)*(
d/144 - e/72) - log(x - 2)*((35*d)/432 + (29*e)/216)
```

3.99 $\int \frac{(2+x)(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx$

Optimal result

Integrand size = 26, antiderivative size = 122

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{d+e+f}{12(1-x)}+\frac{d+2e+4f}{36(2-x)}-\frac{d-e+f}{36(1+x)} +\frac{1}{36}(2d+5e+8f)\log(1-x) -\frac{1}{432}(35d+58e+92f)\log(2-x) +\frac{1}{108}(2d+e-4f)\log(1+x)+\frac{1}{144}(d-2e+4f)\log(2+x)$$

```
[Out] 1/12*(d+e+f)/(1-x)+1/36*(d+2*e+4*f)/(2-x)+1/36*(-d+e-f)/(1+x)+1/36*(2*d+5*e
+8*f)*ln(1-x)-1/432*(35*d+58*e+92*f)*ln(2-x)+1/108*(2*d+e-4*f)*ln(1+x)+1/14
4*(d-2*e+4*f)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.15 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.077, Rules used = {1600, 6874}

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{d-e+f}{36(x+1)}+\frac{d+e+f}{12(1-x)}+\frac{d+2e+4f}{36(2-x)} +\frac{1}{36}\log(1-x)(2d+5e+8f) -\frac{1}{432}\log(2-x)(35d+58e+92f) +\frac{1}{108}\log(x+1)(2d+e-4f)+\frac{1}{144}\log(x+2)(d-2e+4f)$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d + e + f)/(12*(1 - x)) + (d + 2*e + 4*f)/(36*(2 - x)) - (d - e + f)/(36*(
1 + x)) + ((2*d + 5*e + 8*f)*Log[1 - x])/36 - ((35*d + 58*e + 92*f)*Log[2 -
 x])/432 + ((2*d + e - 4*f)*Log[1 + x])/108 + ((d - 2*e + 4*f)*Log[2 + x])/
144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2}{(2+x)\left(2-x-2x^2+x^3\right)^2}\,dx \\
&= \int \left(\frac{d+2e+4f}{36(-2+x)^2}+\frac{-35d-58e-92f}{432(-2+x)}+\frac{d+e+f}{12(-1+x)^2}+\frac{2d+5e+8f}{36(-1+x)}\right. \\
&\qquad \left.+\frac{d-e+f}{36(1+x)^2}+\frac{2d+e-4f}{108(1+x)}+\frac{d-2e+4f}{144(2+x)}\right)dx \\
&= \frac{d+e+f}{12(1-x)}+\frac{d+2e+4f}{36(2-x)}-\frac{d-e+f}{36(1+x)} \\
&\quad +\frac{1}{36}(2d+5e+8f)\log(1-x)-\frac{1}{432}(35d+58e+92f)\log(2-x) \\
&\quad +\frac{1}{108}(2d+e-4f)\log(1+x)+\frac{1}{144}(d-2e+4f)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.04 (sec) , antiderivative size = 121, normalized size of antiderivative = 0.99

$$\begin{aligned}
\int \frac{(2+x)\left(d+ex+fx^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{1}{432}\Bigg(&\frac{12(d(5+6x-5x^2)+e(10-4x^2)+2f(4+3x-4x^2))}{2-x-2x^2+x^3} \\
&+12(2d+5e+8f)\log(1-x)-(35d+58e+92f)\log(2-x) \\
&+4(2d+e-4f)\log(1+x)+3(d-2e+4f)\log(2+x)\Bigg)
\end{aligned}$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(d*(5 + 6*x - 5*x^2) + e*(10 - 4*x^2) + 2*f*(4 + 3*x - 4*x^2)))/(2 - x
 - 2*x^2 + x^3) + 12*(2*d + 5*e + 8*f)*Log[1 - x] - (35*d + 58*e + 92*f)*Lo
g[2 - x] + 4*(2*d + e - 4*f)*Log[1 + x] + 3*(d - 2*e + 4*f)*Log[2 + x])/432
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 113, normalized size of antiderivative = 0.93

method	result
default	$\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}\right)\ln\left(x+2\right)-\frac{\frac{d}{36}-\frac{e}{36}+\frac{f}{36}}{x+1}+\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}\right)\ln\left(x+1\right)-\frac{\frac{d}{12}+\frac{e}{12}+\frac{f}{12}}{x-1}+\left(\frac{d}{18}+\frac{5e}{36}+\right.$
norman	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}\right)x^3+\left(\frac{17d}{36}+\frac{5e}{18}+\frac{5f}{9}\right)x+\left(-\frac{d}{9}-\frac{2e}{9}-\frac{5f}{18}\right)x^2+\frac{5d}{18}+\frac{5e}{9}+\frac{4f}{9}}{x^4-5x^2+4}+\left(-\frac{35d}{432}-\frac{29e}{216}-\frac{23f}{108}\right)\ln\left(x-2\right)+\left(\frac{d}{18}\right.$
risch	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}\right)x^2+\left(\frac{d}{6}+\frac{f}{6}\right)x+\frac{5d}{36}+\frac{5e}{18}+\frac{2f}{9}}{x^3-2x^2-x+2}+\frac{\ln(x+2)d}{144}-\frac{\ln(x+2)e}{72}+\frac{\ln(x+2)f}{36}-\frac{35\ln(2-x)d}{432}-\frac{29\ln(2-x)e}{216}-$
parallelrisch	$-\frac{-96f+60d\,x^2-60d-120e+96f\,x^2-72dx-60\ln(x-1)x^3e-8\ln(x+1)x^3d-4\ln(x+1)x^3e-3\ln(x+2)x^3d+6\ln(x+2)x^3e+70\ln}{}$

```
[In] int((x+2)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (1/144*d-1/72*e+1/36*f)*ln(x+2)-(1/36*d-1/36*e+1/36*f)/(x+1)+(1/54*d+1/108*
e-1/27*f)*ln(x+1)-(1/12*d+1/12*e+1/12*f)/(x-1)+(1/18*d+5/36*e+2/9*f)*ln(x-1
)+(-35/432*d-29/216*e-23/108*f)*ln(x-2)-(1/36*d+1/18*e+1/9*f)/(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 267 vs. 2(104) = 208.

Time = 0.36 (sec) , antiderivative size = 267, normalized size of antiderivative = 2.19

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{12\,(5\,d+4\,e+8\,f)x^2-72\,(d+f)x-3\,((d-2\,e+4\,f)x^3-2\,(d-2\,e+4\,f)x^2-(d-2\,e+4\,f)x+2}{}$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
```

```
[Out] -1/432*(12*(5*d + 4*e + 8*f)*x^2 - 72*(d + f)*x - 3*((d - 2*e + 4*f)*x^3 -
2*(d - 2*e + 4*f)*x^2 - (d - 2*e + 4*f)*x + 2*d - 4*e + 8*f)*log(x + 2) - 4
*((2*d + e - 4*f)*x^3 - 2*(2*d + e - 4*f)*x^2 - (2*d + e - 4*f)*x + 4*d + 2
*e - 8*f)*log(x + 1) - 12*((2*d + 5*e + 8*f)*x^3 - 2*(2*d + 5*e + 8*f)*x^2
- (2*d + 5*e + 8*f)*x + 4*d + 10*e + 16*f)*log(x - 1) + ((35*d + 58*e + 92*
f)*x^3 - 2*(35*d + 58*e + 92*f)*x^2 - (35*d + 58*e + 92*f)*x + 70*d + 116*e
 + 184*f)*log(x - 2) - 60*d - 120*e - 96*f)/(x^3 - 2*x^2 - x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)\,(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((2+x)*(f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 108, normalized size of antiderivative = 0.89

$$\begin{aligned}\int \frac{(2+x)\,(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{144}\,(d-2\,e+4\,f)\log\left(x+2\right)+\frac{1}{108}\,(2\,d+e-4\,f)\log\left(x+1\right)\\ &\quad+\frac{1}{36}\,(2\,d+5\,e+8\,f)\log\left(x-1\right)\\ &\quad-\frac{1}{432}\,(35\,d+58\,e+92\,f)\log\left(x-2\right)\\ &\quad-\frac{(5\,d+4\,e+8\,f)x^2-6\,(d+f)x-5\,d-10\,e-8\,f}{36\,(x^3-2\,x^2-x+2)}\end{aligned}$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
[Out] 1/144*(d - 2*e + 4*f)*log(x + 2) + 1/108*(2*d + e - 4*f)*log(x + 1) + 1/36*
(2*d + 5*e + 8*f)*log(x - 1) - 1/432*(35*d + 58*e + 92*f)*log(x - 2) - 1/36
*((5*d + 4*e + 8*f)*x^2 - 6*(d + f)*x - 5*d - 10*e - 8*f)/(x^3 - 2*x^2 - x
+ 2)
```

Giac [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 112, normalized size of antiderivative = 0.92

$$\begin{aligned}\int \frac{(2+x)\,(d+ex+fx^2)}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{144}\,(d-2\,e+4\,f)\log\left(|x+2|\right)\\ &\quad+\frac{1}{108}\,(2\,d+e-4\,f)\log\left(|x+1|\right)\\ &\quad+\frac{1}{36}\,(2\,d+5\,e+8\,f)\log\left(|x-1|\right)\\ &\quad-\frac{1}{432}\,(35\,d+58\,e+92\,f)\log\left(|x-2|\right)\\ &\quad-\frac{(5\,d+4\,e+8\,f)x^2-6\,(d+f)x-5\,d-10\,e-8\,f}{36\,(x+1)(x-1)(x-2)}\end{aligned}$$

```
[In] integrate((2+x)*(f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/144*(d - 2*e + 4*f)*log(abs(x + 2)) + 1/108*(2*d + e - 4*f)*log(abs(x + 1
)) + 1/36*(2*d + 5*e + 8*f)*log(abs(x - 1)) - 1/432*(35*d + 58*e + 92*f)*lo
g(abs(x - 2)) - 1/36*((5*d + 4*e + 8*f)*x^2 - 6*(d + f)*x - 5*d - 10*e - 8*
f)/((x + 1)*(x - 1)*(x - 2))
```

Mupad [B] (verification not implemented)

Time = 7.90 (sec) , antiderivative size = 113, normalized size of antiderivative = 0.93

$$\int \frac{(2+x)\left(d+ex+fx^2\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \ln\left(x-1\right)\,\left(\frac{d}{18}+\frac{5\,e}{36}+\frac{2\,f}{9}\right)+\ln\left(x+1\right)\,\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}\right) + \ln\left(x+2\right)\,\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}\right) - \ln\left(x-2\right)\,\left(\frac{35\,d}{432}+\frac{29\,e}{216}+\frac{23\,f}{108}\right) - \frac{\left(-\frac{5\,d}{36}-\frac{e}{9}-\frac{2\,f}{9}\right)\,x^2+\left(\frac{d}{6}+\frac{f}{6}\right)\,x+\frac{5\,d}{36}+\frac{5\,e}{18}+\frac{2\,f}{9}}{-x^3+2\,x^2+x-2}$$

```
[In] int(((x + 2)*(d + e*x + f*x^2))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/18 + (5*e)/36 + (2*f)/9) + log(x + 1)*(d/54 + e/108 - f/27) +
 log(x + 2)*(d/144 - e/72 + f/36) - log(x - 2)*((35*d)/432 + (29*e)/216 + (
23*f)/108) - ((5*d)/36 + (5*e)/18 + (2*f)/9 + x*(d/6 + f/6) - x^2*((5*d)/36
 + e/9 + (2*f)/9))/(x + 2*x^2 - x^3 - 2)
```

3.100 $\int \frac{(2+x)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 31, antiderivative size = 141

$$\begin{aligned}\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{\left(4-5x^2+x^4\right)^2} dx &= \frac{d+e+f+g}{12(1-x)}+\frac{d+2e+4f+8g}{36(2-x)}-\frac{d-e+f-g}{36(1+x)} \\ &+\frac{1}{36}(2d+5e+8f+11g)\log(1-x) \\ &-\frac{1}{432}(35d+58e+92f+136g)\log(2-x) \\ &+\frac{1}{108}(2d+e-4f+7g)\log(1+x) \\ &+\frac{1}{144}(d-2e+4f-8g)\log(2+x)\end{aligned}$$

```
[Out] 1/12*(d+e+f+g)/(1-x)+1/36*(d+2*e+4*f+8*g)/(2-x)+1/36*(-d+e-f+g)/(1+x)+1/36*
(2*d+5*e+8*f+11*g)*ln(1-x)-1/432*(35*d+58*e+92*f+136*g)*ln(2-x)+1/108*(2*d+
e-4*f+7*g)*ln(1+x)+1/144*(d-2*e+4*f-8*g)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}} = 0.065$, Rules used

$= \{1600, 6874\}$

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{d-e+f-g}{36(x+1)} + \frac{d+e+f+g}{12(1-x)} + \frac{d+2e+4f+8g}{36(2-x)} + \frac{1}{36}\log(1-x)(2d+5e+8f+11g) - \frac{1}{432}\log(2-x)(35d+58e+92f+136g) + \frac{1}{108}\log(x+1)(2d+e-4f+7g) + \frac{1}{144}\log(x+2)(d-2e+4f-8g)$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d + e + f + g)/(12*(1 - x)) + (d + 2*e + 4*f + 8*g)/(36*(2 - x)) - (d - e
+ f - g)/(36*(1 + x)) + ((2*d + 5*e + 8*f + 11*g)*Log[1 - x])/36 - ((35*d +
 58*e + 92*f + 136*g)*Log[2 - x])/432 + ((2*d + e - 4*f + 7*g)*Log[1 + x])/
108 + ((d - 2*e + 4*f - 8*g)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{d+ex+fx^2+gx^3}{(2+x)\left(2-x-2x^2+x^3\right)^2}\,dx \\
&= \int \left(\frac{d+2e+4f+8g}{36(-2+x)^2} + \frac{-35d-58e-92f-136g}{432(-2+x)} + \frac{d+e+f+g}{12(-1+x)^2} + \frac{2d+5e+8f+11g}{36(-1+x)} + \frac{d-e+f-g}{36(1+x)^2} + \frac{2d+e-4f+7g}{108(1+x)} + \frac{d-2e+4f-8g}{144(2+x)}\right)dx \\
&= \frac{d+e+f+g}{12(1-x)} + \frac{d+2e+4f+8g}{36(2-x)} - \frac{d-e+f-g}{36(1+x)} + \frac{1}{36}(2d+5e+8f+11g)\log(1-x) - \frac{1}{432}(35d+58e+92f+136g)\log(2-x) \\
&\quad + \frac{1}{108}(2d+e-4f+7g)\log(1+x) + \frac{1}{144}(d-2e+4f-8g)\log(2+x)
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.05 (sec) , antiderivative size = 144, normalized size of antiderivative = 1.02

$$\int \frac{(2+x)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{432}\left(\frac{12(d(5+6x-5x^2)+2(g(8-5x^2)+f(4+3x-4x^2)+e(5-2x^2)))}{2-x-2x^2+x^3} + 12(2d+5e+8f+11g)\log(1-x)-(35d+58e+92f+136g)\log(2-x) + 4(2d+e-4f+7g)\log(1+x)+3(d-2e+4f-8g)\log(2+x)\right)$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] ((12*(d*(5 + 6*x - 5*x^2) + 2*(g*(8 - 5*x^2) + f*(4 + 3*x - 4*x^2) + e*(5 -
 2*x^2))))/(2 - x - 2*x^2 + x^3) + 12*(2*d + 5*e + 8*f + 11*g)*Log[1 - x] -
 (35*d + 58*e + 92*f + 136*g)*Log[2 - x] + 4*(2*d + e - 4*f + 7*g)*Log[1 +
x] + 3*(d - 2*e + 4*f - 8*g)*Log[2 + x])/432
```

Maple [A] (verified)

Time = 0.09 (sec) , antiderivative size = 134, normalized size of antiderivative = 0.95

method	result
default	$\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}\right)\ln\left(x+2\right)-\frac{\frac{d}{36}-\frac{e}{36}+\frac{f}{36}-\frac{g}{36}}{x+1}+\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7g}{108}\right)\ln\left(x+1\right)-\frac{\frac{d}{12}+\frac{e}{12}+\frac{f}{12}+}{x-1}$
norman	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}\right)x^3+\left(\frac{17d}{36}+\frac{5e}{18}+\frac{5f}{9}+\frac{4g}{9}\right)x+\left(-\frac{d}{9}-\frac{2e}{9}-\frac{5f}{18}-\frac{5g}{9}\right)x^2+\frac{5d}{18}+\frac{5e}{9}+\frac{4f}{9}+\frac{8g}{9}}{x^4-5x^2+4}+\left(-\frac{35d}{432}-\frac{29e}{216}-\frac{23f}{108}-\frac{17g}{54}\right)$
risch	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}\right)x^2+\left(\frac{d}{6}+\frac{f}{6}\right)x+\frac{5d}{36}+\frac{5e}{18}+\frac{2f}{9}+\frac{4g}{9}}{x^3-2x^2-x+2}+\frac{\ln(x+2)d}{144}-\frac{\ln(x+2)e}{72}+\frac{\ln(x+2)f}{36}-\frac{\ln(x+2)g}{18}+\frac{\ln(-x-1)d}{54}+$
parallelrisch	$-\frac{-96f-192g+60d\,x^2-60d-120e+96f\,x^2+120g\,x^2-72dx-60\ln(x-1)x^3e-8\ln(x+1)x^3d-4\ln(x+1)x^3e-3\ln(x+2)x^3d+6\ln(}{}$

```
[In] int((x+2)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE)
```

```
[Out] (1/144*d-1/72*e+1/36*f-1/18*g)*ln(x+2)-(1/36*d-1/36*e+1/36*f-1/36*g)/(x+1)+
(1/54*d+1/108*e-1/27*f+7/108*g)*ln(x+1)-(1/12*d+1/12*e+1/12*f+1/12*g)/(x-1)
+(1/18*d+5/36*e+2/9*f+11/36*g)*ln(x-1)+(-35/432*d-29/216*e-23/108*f-17/54*g
)*ln(x-2)-(1/36*d+1/18*e+1/9*f+2/9*g)/(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 321 vs. $2(123) = 246$.

Time = 0.78 (sec) , antiderivative size = 321, normalized size of antiderivative = 2.28

$$\int \frac{(2+x)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx =$$
$$-\frac{12\,(5\,d+4\,e+8\,f+10\,g)x^2-72\,(d+f)x-3\,((d-2\,e+4\,f-8\,g)x^3-2\,(d-2\,e+4\,f-8\,g)x^2-(d$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fricas")
[Out] -1/432*(12*(5*d + 4*e + 8*f + 10*g)*x^2 - 72*(d + f)*x - 3*((d - 2*e + 4*f
- 8*g)*x^3 - 2*(d - 2*e + 4*f - 8*g)*x^2 - (d - 2*e + 4*f - 8*g)*x + 2*d -
4*e + 8*f - 16*g)*log(x + 2) - 4*((2*d + e - 4*f + 7*g)*x^3 - 2*(2*d + e -
4*f + 7*g)*x^2 - (2*d + e - 4*f + 7*g)*x + 4*d + 2*e - 8*f + 14*g)*log(x +
1) - 12*((2*d + 5*e + 8*f + 11*g)*x^3 - 2*(2*d + 5*e + 8*f + 11*g)*x^2 - (2
*d + 5*e + 8*f + 11*g)*x + 4*d + 10*e + 16*f + 22*g)*log(x - 1) + ((35*d +
58*e + 92*f + 136*g)*x^3 - 2*(35*d + 58*e + 92*f + 136*g)*x^2 - (35*d + 58*
e + 92*f + 136*g)*x + 70*d + 116*e + 184*f + 272*g)*log(x - 2) - 60*d - 120
*e - 96*f - 192*g)/(x^3 - 2*x^2 - x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((2+x)*(g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 126, normalized size of antiderivative = 0.89

$$\begin{aligned}\int \frac{(2+x)(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx &= \frac{1}{144}\,(d-2\,e+4\,f-8\,g)\log(x+2)+\frac{1}{108}\,(2\,d+e-4\,f+7\,g)\log(x+1) \\ &+\frac{1}{36}\,(2\,d+5\,e+8\,f+11\,g)\log(x-1)-\frac{1}{432}\,(35\,d+58\,e+92\,f+136\,g)\log(x-2) \\ &-\frac{(5\,d+4\,e+8\,f+10\,g)x^2-6\,(d+f)x-5\,d-10\,e-8\,f-16\,g}{36\,(x^3-2\,x^2-x+2)}\end{aligned}$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="maxima")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g)*log(x + 2) + 1/108*(2*d + e - 4*f + 7*g)*log(x
+ 1) + 1/36*(2*d + 5*e + 8*f + 11*g)*log(x - 1) - 1/432*(35*d + 58*e + 92*f
 + 136*g)*log(x - 2) - 1/36*((5*d + 4*e + 8*f + 10*g)*x^2 - 6*(d + f)*x - 5
*d - 10*e - 8*f - 16*g)/(x^3 - 2*x^2 - x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 130, normalized size of antiderivative = 0.92

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}\,(d-2\,e+4\,f-8\,g)\log\left(|x+2|\right)+\frac{1}{108}\,(2\,d+e-4\,f+7\,g)\log\left(|x+1|\right) + \frac{1}{36}\,(2\,d+5\,e+8\,f+11\,g)\log\left(|x-1|\right)-\frac{1}{432}\,(35\,d+58\,e+92\,f+136\,g)\log\left(|x-2|\right) - \frac{(5\,d+4\,e+8\,f+10\,g)x^2-6\,(d+f)x-5\,d-10\,e-8\,f-16\,g}{36\,(x+1)(x-1)(x-2)}$$

```
[In] integrate((2+x)*(g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="giac")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g)*log(abs(x + 2)) + 1/108*(2*d + e - 4*f + 7*g)*l
og(abs(x + 1)) + 1/36*(2*d + 5*e + 8*f + 11*g)*log(abs(x - 1)) - 1/432*(35*
d + 58*e + 92*f + 136*g)*log(abs(x - 2)) - 1/36*((5*d + 4*e + 8*f + 10*g)*x
^2 - 6*(d + f)*x - 5*d - 10*e - 8*f - 16*g)/((x + 1)*(x - 1)*(x - 2))
```

Mupad [B] (verification not implemented)

Time = 8.21 (sec) , antiderivative size = 131, normalized size of antiderivative = 0.93

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3)}{(4-5x^2+x^4)^2}\,dx = \ln\left(x-1\right)\,\left(\frac{d}{18}+\frac{5\,e}{36}+\frac{2\,f}{9}+\frac{11\,g}{36}\right)+\ln\left(x+2\right)\,\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}\right) + \ln\left(x+1\right)\,\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7\,g}{108}\right)-\ln\left(x-2\right)\,\left(\frac{35\,d}{432}+\frac{29\,e}{216}+\frac{23\,f}{108}+\frac{17\,g}{54}\right) - \frac{\left(-\frac{5\,d}{36}-\frac{e}{9}-\frac{2\,f}{9}-\frac{5\,g}{18}\right)\,x^2+\left(\frac{d}{6}+\frac{f}{6}\right)\,x+\frac{5\,d}{36}+\frac{5\,e}{18}+\frac{2\,f}{9}+\frac{4\,g}{9}}{-x^3+2\,x^2+x-2}$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/18 + (5*e)/36 + (2*f)/9 + (11*g)/36) + log(x + 2)*(d/144 - e/
72 + f/36 - g/18) + log(x + 1)*(d/54 + e/108 - f/27 + (7*g)/108) - log(x -
```

```
2)*((35*d)/432 + (29*e)/216 + (23*f)/108 + (17*g)/54) - ((5*d)/36 + (5*e)/1
8 + (2*f)/9 + (4*g)/9 - x^2*((5*d)/36 + e/9 + (2*f)/9 + (5*g)/18) + x*(d/6
+ f/6))/(x + 2*x^2 - x^3 - 2)
```

3.101 $\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{\left(4-5x^2+x^4\right)^2}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 158

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{d+e+f+g+h}{12(1-x)}+\frac{d+2e+4f+8g+16h}{36(2-x)} - \frac{d-e+f-g+h}{36(1+x)} + \frac{1}{36}(2d+5e+8f+11g+14h)\log(1-x) - \frac{1}{432}(35d+58e+92f+136g+176h)\log(2-x) + \frac{1}{108}(2d+e-4f+7g-10h)\log(1+x) + \frac{1}{144}(d-2e+4f-8g+16h)\log(2+x)$$

```
[Out] 1/12*(d+e+f+g+h)/(1-x)+1/36*(d+2*e+4*f+8*g+16*h)/(2-x)+1/36*(-d+e-f+g-h)/(1
+x)+1/36*(2*d+5*e+8*f+11*g+14*h)*ln(1-x)-1/432*(35*d+58*e+92*f+136*g+176*h)
*ln(2-x)+1/108*(2*d+e-4*f+7*g-10*h)*ln(1+x)+1/144*(d-2*e+4*f-8*g+16*h)*ln(2
+x)
```

Rubi [A] (verified)

Time = 0.20 (sec) , antiderivative size = 158, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.056, Rules used = {1600, 6874}

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{d-e+f-g+h}{36(x+1)}+\frac{d+e+f+g+h}{12(1-x)}+\frac{d+2e+4f+8g+16h}{36(2-x)}+\frac{1}{36}\log(1-x)(2d+5e+8f+11g+14h)-\frac{1}{432}\log(2-x)(35d+58e+92f+136g+176h)+\frac{1}{108}\log(x+1)(2d+e-4f+7g-10h)+\frac{1}{144}\log(x+2)(d-2e+4f-8g+16h)$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4)^2,x]
```

```
[Out] (d + e + f + g + h)/(12*(1 - x)) + (d + 2*e + 4*f + 8*g + 16*h)/(36*(2 - x)
) - (d - e + f - g + h)/(36*(1 + x)) + ((2*d + 5*e + 8*f + 11*g + 14*h)*Log
[1 - x])/36 - ((35*d + 58*e + 92*f + 136*g + 176*h)*Log[2 - x])/432 + ((2*d
 + e - 4*f + 7*g - 10*h)*Log[1 + x])/108 + ((d - 2*e + 4*f - 8*g + 16*h)*Lo
g[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2+gx^3+hx^4}{(2+x)\left(2-x-2x^2+x^3\right)^2}\,dx$$

$$= \int \left(\frac{d+2e+4f+8g+16h}{36(-2+x)^2}+\frac{-35d-58e-92f-136g-176h}{432(-2+x)}+\frac{d+e+f+g+h}{12(-1+x)^2}+\frac{2d+5e+8f+11g+14h}{36(-1+x)}+\frac{d-e+f-g+h}{36(1+x)^2}+\frac{2d+e-4f+7g-10h}{108(1+x)}+\frac{d-2e+4f-8g+16h}{144(2+x)}\right)dx$$

$$= \frac{d+e+f+g+h}{12(1-x)} + \frac{d+2e+4f+8g+16h}{36(2-x)} - \frac{d-e+f-g+h}{36(1+x)} + \frac{1}{36}(2d+5e+8f+11g+14h)\log(1-x) - \frac{1}{432}(35d+58e+92f+136g+176h)\log(2-x) + \frac{1}{108}(2d+e-4f+7g-10h)\log(1+x) + \frac{1}{144}(d-2e+4f-8g+16h)\log(2+x)$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 169, normalized size of antiderivative = 1.07

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4\right)}{\left(4-5x^2+x^4\right)^2}\,dx = \frac{1}{432}\left(\frac{12(d(5+6x-5x^2)+2(8g+10h+3hx-5gx^2-10hx^2+f(4+3x-4x^2)+e(5-2x^2)))}{2-x-2x^2+x^3} + 12(2d+5e+8f+11g+14h)\log(1-x) - (35d+58e+92f+136g+176h)\log(2-x) + 4(2d+e-4f+7g-10h)\log(1+x) + 3(d-2e+4f-8g+16h)\log(2+x)\right)$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(4 - 5*x^2 + x^4)^2,x
]
```

```
[Out] ((12*(d*(5 + 6*x - 5*x^2) + 2*(8*g + 10*h + 3*h*x - 5*g*x^2 - 10*h*x^2 + f*
(4 + 3*x - 4*x^2) + e*(5 - 2*x^2))))/(2 - x - 2*x^2 + x^3) + 12*(2*d + 5*e
+ 8*f + 11*g + 14*h)*Log[1 - x] - (35*d + 58*e + 92*f + 136*g + 176*h)*Log[
2 - x] + 4*(2*d + e - 4*f + 7*g - 10*h)*Log[1 + x] + 3*(d - 2*e + 4*f - 8*g
 + 16*h)*Log[2 + x])/432
```

Maple [A] (verified)

Time = 0.11 (sec) , antiderivative size = 155, normalized size of antiderivative = 0.98

method	result
default	$\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}\right)\ln\left(x+2\right)-\frac{\frac{d}{36}-\frac{e}{36}+\frac{f}{36}-\frac{g}{36}+\frac{h}{36}}{x+1}+\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7g}{108}-\frac{5h}{54}\right)\ln\left(x+1\right)$
norman	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}-\frac{5h}{9}\right)x^3+\left(\frac{17d}{36}+\frac{5e}{18}+\frac{5f}{9}+\frac{4g}{9}+\frac{8h}{9}\right)x+\left(-\frac{d}{9}-\frac{2e}{9}-\frac{5f}{18}-\frac{5g}{9}-\frac{17h}{18}\right)x^2+\frac{5d}{18}+\frac{5e}{9}+\frac{10h}{9}+\frac{4f}{9}+\frac{8g}{9}}{x^4-5x^2+4}+\left(-\frac{35d}{432}-\right.$
risch	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}-\frac{5h}{9}\right)x^2+\left(\frac{h}{6}+\frac{f}{6}+\frac{d}{6}\right)x+\frac{5d}{36}+\frac{5e}{18}+\frac{2f}{9}+\frac{4g}{9}+\frac{5h}{9}}{x^3-2x^2-x+2}+\frac{\ln(x+2)d}{144}-\frac{\ln(x+2)e}{72}+\frac{\ln(x+2)f}{36}-\frac{\ln(x+2)g}{18}+$
parallelrisch	$-\frac{-96f-192g+60d\,x^2-60d-240h-120e+96f\,x^2-176\ln(x-2)xh+168\ln(x-1)xh-40\ln(x+1)xh+48\ln(x+2)xh+120g\,x^2-72}{}$

```
[In] int((x+2)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNVERBOSE
)
```

```
[Out] (1/144*d-1/72*e+1/36*f-1/18*g+1/9*h)*ln(x+2)-(1/36*d-1/36*e+1/36*f-1/36*g+1
/36*h)/(x+1)+(1/54*d+1/108*e-1/27*f+7/108*g-5/54*h)*ln(x+1)-(1/12*d+1/12*e+
1/12*f+1/12*g+1/12*h)/(x-1)+(1/18*d+5/36*e+2/9*f+11/36*g+7/18*h)*ln(x-1)+(-
35/432*d-29/216*e-23/108*f-17/54*g-11/27*h)*ln(x-2)-(1/36*d+1/18*e+1/9*f+2/
9*g+4/9*h)/(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 376 vs. $2(140) = 280$.

Time = 3.63 (sec) , antiderivative size = 376, normalized size of antiderivative = 2.38

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = -\frac{12\,(5\,d+4\,e+8\,f+10\,g+20\,h)x^2-72\,(d+f+h)x-3\,((d-2\,e+4\,f-8\,g+16\,h)x^3-2\,(d-2\,e-}{}$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="fri
cas")
```

```
[Out] -1/432*(12*(5*d + 4*e + 8*f + 10*g + 20*h)*x^2 - 72*(d + f + h)*x - 3*((d -
 2*e + 4*f - 8*g + 16*h)*x^3 - 2*(d - 2*e + 4*f - 8*g + 16*h)*x^2 - (d - 2*
e + 4*f - 8*g + 16*h)*x + 2*d - 4*e + 8*f - 16*g + 32*h)*log(x + 2) - 4*((2
*d + e - 4*f + 7*g - 10*h)*x^3 - 2*(2*d + e - 4*f + 7*g - 10*h)*x^2 - (2*d
+ e - 4*f + 7*g - 10*h)*x + 4*d + 2*e - 8*f + 14*g - 20*h)*log(x + 1) - 12*
((2*d + 5*e + 8*f + 11*g + 14*h)*x^3 - 2*(2*d + 5*e + 8*f + 11*g + 14*h)*x^
2 - (2*d + 5*e + 8*f + 11*g + 14*h)*x + 4*d + 10*e + 16*f + 22*g + 28*h)*lo
g(x - 1) + ((35*d + 58*e + 92*f + 136*g + 176*h)*x^3 - 2*(35*d + 58*e + 92*
f + 136*g + 176*h)*x^2 - (35*d + 58*e + 92*f + 136*g + 176*h)*x + 70*d + 11
6*e + 184*f + 272*g + 352*h)*log(x - 2) - 60*d - 120*e - 96*f - 192*g - 240
*h)/(x^3 - 2*x^2 - x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = \text{Timed out}$$

```
[In] integrate((2+x)*(h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 145, normalized size of antiderivative = 0.92

$$\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}(d-2e+4f-8g+16h)\log(x+2)+\frac{1}{108}(2d+e-4f+7g-10h)\log(x+1) + \frac{1}{36}(2d+5e+8f+11g+14h)\log(x-1) - \frac{1}{432}(35d+58e+92f+136g+176h)\log(x-2) - \frac{(5d+4e+8f+10g+20h)x^2-6(d+f+h)x-5d-10e-8f-16g-20h}{36(x^3-2x^2-x+2)}$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="max
ima")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g + 16*h)*log(x + 2) + 1/108*(2*d + e - 4*f + 7*g
- 10*h)*log(x + 1) + 1/36*(2*d + 5*e + 8*f + 11*g + 14*h)*log(x - 1) - 1/43
2*(35*d + 58*e + 92*f + 136*g + 176*h)*log(x - 2) - 1/36*((5*d + 4*e + 8*f
+ 10*g + 20*h)*x^2 - 6*(d + f + h)*x - 5*d - 10*e - 8*f - 16*g - 20*h)/(x^3
 - 2*x^2 - x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 149, normalized size of antiderivative = 0.94

$$\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx = \frac{1}{144}(d-2e+4f-8g+16h)\log(|x+2|) + \frac{1}{108}(2d+e-4f+7g-10h)\log(|x+1|) + \frac{1}{36}(2d+5e+8f+11g+14h)\log(|x-1|) - \frac{1}{432}(35d+58e+92f+136g+176h)\log(|x-2|) - \frac{(5d+4e+8f+10g+20h)x^2-6(d+f+h)x-5d-10e-8f-16g-20h}{36(x+1)(x-1)(x-2)}$$

```
[In] integrate((2+x)*(h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorithm="gia
c")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g + 16*h)*log(abs(x + 2)) + 1/108*(2*d + e - 4*f +
 7*g - 10*h)*log(abs(x + 1)) + 1/36*(2*d + 5*e + 8*f + 11*g + 14*h)*log(abs
(x - 1)) - 1/432*(35*d + 58*e + 92*f + 136*g + 176*h)*log(abs(x - 2)) - 1/3
6*((5*d + 4*e + 8*f + 10*g + 20*h)*x^2 - 6*(d + f + h)*x - 5*d - 10*e - 8*f
 - 16*g - 20*h)/((x + 1)*(x - 1)*(x - 2))
```

Mupad [B] (verification not implemented)

Time = 8.43 (sec) , antiderivative size = 152, normalized size of antiderivative = 0.96

$$
\begin{aligned}
&\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4)}{(4-5x^2+x^4)^2}\,dx \\
&= \ln(x-1)\,\left(\frac{d}{18}+\frac{5\,e}{36}+\frac{2\,f}{9}+\frac{11\,g}{36}+\frac{7\,h}{18}\right) \\
&\quad - \frac{\left(-\frac{5\,d}{36}-\frac{e}{9}-\frac{2\,f}{9}-\frac{5\,g}{18}-\frac{5\,h}{9}\right)\,x^2+\left(\frac{d}{6}+\frac{f}{6}+\frac{h}{6}\right)\,x+\frac{5\,d}{36}+\frac{5\,e}{18}+\frac{2\,f}{9}+\frac{4\,g}{9}+\frac{5\,h}{9}}{-x^3+2\,x^2+x-2} \\
&\quad + \ln(x+2)\,\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}\right)+\ln(x+1)\,\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7\,g}{108}-\frac{5\,h}{54}\right) \\
&\quad - \ln(x-2)\,\left(\frac{35\,d}{432}+\frac{29\,e}{216}+\frac{23\,f}{108}+\frac{17\,g}{54}+\frac{11\,h}{27}\right)
\end{aligned}
$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4))/(x^4 - 5*x^2 + 4)^2,x)
```

```
[Out] log(x - 1)*(d/18 + (5*e)/36 + (2*f)/9 + (11*g)/36 + (7*h)/18) - ((5*d)/36 +
 (5*e)/18 + (2*f)/9 + (4*g)/9 + (5*h)/9 - x^2*((5*d)/36 + e/9 + (2*f)/9 + (
5*g)/18 + (5*h)/9) + x*(d/6 + f/6 + h/6))/(x + 2*x^2 - x^3 - 2) + log(x + 2
)*(d/144 - e/72 + f/36 - g/18 + h/9) + log(x + 1)*(d/54 + e/108 - f/27 + (7
*g)/108 - (5*h)/54) - log(x - 2)*((35*d)/432 + (29*e)/216 + (23*f)/108 + (1
7*g)/54 + (11*h)/27)
```

3.102 $\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2} dx$

Optimal result

Integrand size = 41, antiderivative size = 177

$$\begin{aligned}
&\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2} dx \\
&= \frac{d+e+f+g+h+i}{12(1-x)} + \frac{d+2e+4f+8g+16h+32i}{36(2-x)} \\
&\quad - \frac{d-e+f-g+h-i}{36(1+x)} + \frac{1}{36}(2d+5e+8f+11g+14h+17i)\log(1-x) \\
&\quad - \frac{1}{432}(35d+58e+92f+136g+176h+160i)\log(2-x) \\
&\quad + \frac{1}{108}(2d+e-4f+7g-10h+13i)\log(1+x)+\frac{1}{144}(d-2e+4f-8g+16h-32i)\log(2+x)
\end{aligned}$$

```
[Out] 1/12*(d+e+f+g+h+i)/(1-x)+1/36*(d+2*e+4*f+8*g+16*h+32*i)/(2-x)+1/36*(-d+e-f+
g-h+i)/(1+x)+1/36*(2*d+5*e+8*f+11*g+14*h+17*i)*ln(1-x)-1/432*(35*d+58*e+92*
f+136*g+176*h+160*i)*ln(2-x)+1/108*(2*d+e-4*f+7*g-10*h+13*i)*ln(1+x)+1/144*
(d-2*e+4*f-8*g+16*h-32*i)*ln(2+x)
```

Rubi [A] (verified)

Time = 0.23 (sec) , antiderivative size = 177, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.049, Rules used

$= \{1600, 6874\}$

$$\int \frac{(2+x)(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx$$
$$= -\frac{d-e+f-g+h-i}{36(x+1)} + \frac{d+e+f+g+h+i}{12(1-x)} + \frac{d+2e+4f+8g+16h+32i}{36(2-x)}$$
$$+ \frac{1}{36}\log(1-x)(2d+5e+8f+11g+14h+17i)$$
$$- \frac{1}{432}\log(2-x)(35d+58e+92f+136g+176h+160i)$$
$$+ \frac{1}{108}\log(x+1)(2d+e-4f+7g-10h+13i) + \frac{1}{144}\log(x+2)(d-2e+4f-8g+16h-32i)$$

```
[In] Int[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2 + x^4)^2
,x]
```

```
[Out] (d + e + f + g + h + i)/(12*(1 - x)) + (d + 2*e + 4*f + 8*g + 16*h + 32*i)/
(36*(2 - x)) - (d - e + f - g + h - i)/(36*(1 + x)) + ((2*d + 5*e + 8*f + 1
1*g + 14*h + 17*i)*Log[1 - x])/36 - ((35*d + 58*e + 92*f + 136*g + 176*h +
160*i)*Log[2 - x])/432 + ((2*d + e - 4*f + 7*g - 10*h + 13*i)*Log[1 + x])/1
08 + ((d - 2*e + 4*f - 8*g + 16*h - 32*i)*Log[2 + x])/144
```

Rule 1600

```
Int[(u_.)*(Px_)^(p_.)*(Qx_)^(q_.), x_Symbol] :> Int[u*PolynomialQuotient[Px
, Qx, x]^p*Qx^(p + q), x] /; FreeQ[q, x] && PolyQ[Px, x] && PolyQ[Qx, x] &&
 EqQ[PolynomialRemainder[Px, Qx, x], 0] && IntegerQ[p] && LtQ[p*q, 0]
```

Rule 6874

```
Int[u_, x_Symbol] :> With[{v = ExpandIntegrand[u, x]}, Int[v, x] /; SumQ[v]
]
```

Rubi steps

$$\text{integral} = \int \frac{d+ex+fx^2+gx^3+hx^4+ix^5}{(2+x)(2-x-2x^2+x^3)^2}\,dx$$
$$= \int \left(\frac{d+2e+4f+8g+16h+32i}{36(-2+x)^2} + \frac{-35d-58e-92f-136g-176h-160i}{432(-2+x)} + \frac{d+e+f+g+h+i}{12(-1+x)^2} + \frac{2d+5e+8f+11g+14h+17i}{36(-1+x)} + \frac{d-e+f-g+h-i}{36(1+x)^2} + \frac{2d+e-4f+7g-10h+13i}{108(1+x)} + \frac{d-2e+4f-8g+16h-32i}{144(2+x)}\right)dx$$

$$= \frac{d+e+f+g+h+i}{12(1-x)} + \frac{d+2e+4f+8g+16h+32i}{36(2-x)} - \frac{d-e+f-g+h-i}{36(1+x)} + \frac{1}{36}(2d+5e+8f+11g+14h+17i)\log(1-x) - \frac{1}{432}(35d+58e+92f+136g+176h+160i)\log(2-x) + \frac{1}{108}(2d+e-4f+7g-10h+13i)\log(1+x) + \frac{1}{144}(d-2e+4f-8g+16h-32i)\log(2+x)$$

Mathematica [A] (verified)

Time = 0.07 (sec) , antiderivative size = 195, normalized size of antiderivative = 1.10

$$\int \frac{(2+x)\,(d+ex+fx^2+gx^3+hx^4+ix^5)}{(4-5x^2+x^4)^2}\,dx = \frac{5d+10e+8f+16g+20h+40i+6dx+6fx+6hx-5dx^2-4ex^2-8fx^2-10gx^2-20hx^2-34ix^2}{36\,(2-x-2x^2+x^3)} + \frac{1}{36}(2d+5e+8f+11g+14h+17i)\log(1-x) + \frac{1}{432}(-35d-58e-92f-136g-176h-160i)\log(2-x) + \frac{1}{108}(2d+e-4f+7g-10h+13i)\log(1+x) + \frac{1}{144}(d-2e+4f-8g+16h-32i)\log(2+x)$$

```
[In] Integrate[((2 + x)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(4 - 5*x^2 +
x^4)^2,x]
```

```
[Out] (5*d + 10*e + 8*f + 16*g + 20*h + 40*i + 6*d*x + 6*f*x + 6*h*x - 5*d*x^2 -
4*e*x^2 - 8*f*x^2 - 10*g*x^2 - 20*h*x^2 - 34*i*x^2)/(36*(2 - x - 2*x^2 + x^
3)) + ((2*d + 5*e + 8*f + 11*g + 14*h + 17*i)*Log[1 - x])/36 + ((-35*d - 58
*e - 92*f - 136*g - 176*h - 160*i)*Log[2 - x])/432 + ((2*d + e - 4*f + 7*g
- 10*h + 13*i)*Log[1 + x])/108 + ((d - 2*e + 4*f - 8*g + 16*h - 32*i)*Log[2
 + x])/144
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 176, normalized size of antiderivative = 0.99

method	result
default	$\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}-\frac{2i}{9}\right)\ln\left(x+2\right)-\frac{\frac{d}{36}-\frac{e}{36}+\frac{f}{36}-\frac{g}{36}+\frac{h}{36}-\frac{i}{36}}{x+1}+\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7g}{108}-\frac{5h}{54}\right.$
norman	$\frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}-\frac{5h}{9}-\frac{17i}{18}\right)x^3+\left(\frac{17d}{36}+\frac{5e}{18}+\frac{5f}{9}+\frac{4g}{9}+\frac{8h}{9}+\frac{10i}{9}\right)x+\left(-\frac{d}{9}-\frac{2e}{9}-\frac{5f}{18}-\frac{5g}{9}-\frac{17h}{18}-\frac{17i}{9}\right)x^2+\frac{5d}{18}+\frac{5e}{9}+\frac{10h}{9}+\frac{20i}{9}+\frac{4}{9}}{x^4-5x^2+4}$
risch	$\frac{7\ln(-x-1)g}{108}+\frac{\ln(x-1)d}{18}+\frac{5\ln(x-1)e}{36}+\frac{\ln(x+2)f}{36}-\frac{5\ln(-x-1)h}{54}-\frac{23\ln(2-x)f}{108}-\frac{35\ln(2-x)d}{432}-\frac{29\ln(2-x)e}{216}$
parallelrisch	Expression too large to display

```
[In] int((x+2)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x,method=_RETURNV
ERBOSE)
```

```
[Out] (1/144*d-1/72*e+1/36*f-1/18*g+1/9*h-2/9*i)*ln(x+2)-(1/36*d-1/36*e+1/36*f-1/
36*g+1/36*h-1/36*i)/(x+1)+(1/54*d+1/108*e-1/27*f+7/108*g-5/54*h+13/108*i)*l
n(x+1)-(1/12*d+1/12*e+1/12*f+1/12*g+1/12*h+1/12*i)/(x-1)+(1/18*d+5/36*e+2/9
*f+11/36*g+7/18*h+17/36*i)*ln(x-1)+(-35/432*d-29/216*e-23/108*f-17/54*g-11/
27*h-10/27*i)*ln(x-2)-(1/36*d+1/18*e+1/9*f+2/9*g+4/9*h+8/9*i)/(x-2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 430 vs. $2(159) = 318$.

Time = 20.84 (sec) , antiderivative size = 430, normalized size of antiderivative = 2.43

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx = -\frac{12\,(5\,d+4\,e+8\,f+10\,g+20\,h+34\,i)x^2-72\,(d+f+h)x-3\,((d-2\,e+4\,f-8\,g+16\,h-32\,i)x^3}{}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="fricas")
```

```
[Out] -1/432*(12*(5*d + 4*e + 8*f + 10*g + 20*h + 34*i)*x^2 - 72*(d + f + h)*x -
3*((d - 2*e + 4*f - 8*g + 16*h - 32*i)*x^3 - 2*(d - 2*e + 4*f - 8*g + 16*h
- 32*i)*x^2 - (d - 2*e + 4*f - 8*g + 16*h - 32*i)*x + 2*d - 4*e + 8*f - 16*
g + 32*h - 64*i)*log(x + 2) - 4*((2*d + e - 4*f + 7*g - 10*h + 13*i)*x^3 -
2*(2*d + e - 4*f + 7*g - 10*h + 13*i)*x^2 - (2*d + e - 4*f + 7*g - 10*h + 1
3*i)*x + 4*d + 2*e - 8*f + 14*g - 20*h + 26*i)*log(x + 1) - 12*((2*d + 5*e
+ 8*f + 11*g + 14*h + 17*i)*x^3 - 2*(2*d + 5*e + 8*f + 11*g + 14*h + 17*i)*
x^2 - (2*d + 5*e + 8*f + 11*g + 14*h + 17*i)*x + 4*d + 10*e + 16*f + 22*g +
 28*h + 34*i)*log(x - 1) + ((35*d + 58*e + 92*f + 136*g + 176*h + 160*i)*x^
3 - 2*(35*d + 58*e + 92*f + 136*g + 176*h + 160*i)*x^2 - (35*d + 58*e + 92*
f + 136*g + 176*h + 160*i)*x + 70*d + 116*e + 184*f + 272*g + 352*h + 320*i
)*log(x - 2) - 60*d - 120*e - 96*f - 192*g - 240*h - 480*i)/(x^3 - 2*x^2 -
x + 2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx=\text{Timed out}$$

```
[In] integrate((2+x)*(i*x**5+h*x**4+g*x**3+f*x**2+e*x+d)/(x**4-5*x**2+4)**2,x)
```

```
[Out] Timed out
```

Maxima [A] (verification not implemented)

none

Time = 0.20 (sec) , antiderivative size = 163, normalized size of antiderivative = 0.92

$$\begin{aligned}
&\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx \\
&=\frac{1}{144}\,(d-2\,e+4\,f-8\,g+16\,h-32\,i)\log\left(x+2\right) \\
&\quad+\frac{1}{108}\,(2\,d+e-4\,f+7\,g-10\,h+13\,i)\log\left(x+1\right) \\
&\quad+\frac{1}{36}\,(2\,d+5\,e+8\,f+11\,g+14\,h+17\,i)\log\left(x-1\right) \\
&\quad-\frac{1}{432}\,(35\,d+58\,e+92\,f+136\,g+176\,h+160\,i)\log\left(x-2\right) \\
&\quad-\frac{(5\,d+4\,e+8\,f+10\,g+20\,h+34\,i)x^2-6\,(d+f+h)x-5\,d-10\,e-8\,f-16\,g-20\,h-40\,i}{36\,(x^3-2\,x^2-x+2)}
\end{aligned}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="maxima")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(x + 2) + 1/108*(2*d + e - 4*f
 + 7*g - 10*h + 13*i)*log(x + 1) + 1/36*(2*d + 5*e + 8*f + 11*g + 14*h + 17
*i)*log(x - 1) - 1/432*(35*d + 58*e + 92*f + 136*g + 176*h + 160*i)*log(x -
 2) - 1/36*((5*d + 4*e + 8*f + 10*g + 20*h + 34*i)*x^2 - 6*(d + f + h)*x -
5*d - 10*e - 8*f - 16*g - 20*h - 40*i)/(x^3 - 2*x^2 - x + 2)
```

Giac [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 167, normalized size of antiderivative = 0.94

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx$$
$$= \frac{1}{144}\left(d-2e+4f-8g+16h-32i\right)\log\left(|x+2|\right) + \frac{1}{108}\left(2d+e-4f+7g-10h+13i\right)\log\left(|x+1|\right) + \frac{1}{36}\left(2d+5e+8f+11g+14h+17i\right)\log\left(|x-1|\right) - \frac{1}{432}\left(35d+58e+92f+136g+176h+160i\right)\log\left(|x-2|\right) - \frac{(5d+4e+8f+10g+20h+34i)x^2-6(d+f+h)x-5d-10e-8f-16g-20h-40i}{36(x+1)(x-1)(x-2)}$$

```
[In] integrate((2+x)*(i*x^5+h*x^4+g*x^3+f*x^2+e*x+d)/(x^4-5*x^2+4)^2,x, algorith
m="giac")
```

```
[Out] 1/144*(d - 2*e + 4*f - 8*g + 16*h - 32*i)*log(abs(x + 2)) + 1/108*(2*d + e
- 4*f + 7*g - 10*h + 13*i)*log(abs(x + 1)) + 1/36*(2*d + 5*e + 8*f + 11*g +
 14*h + 17*i)*log(abs(x - 1)) - 1/432*(35*d + 58*e + 92*f + 136*g + 176*h +
 160*i)*log(abs(x - 2)) - 1/36*((5*d + 4*e + 8*f + 10*g + 20*h + 34*i)*x^2
- 6*(d + f + h)*x - 5*d - 10*e - 8*f - 16*g - 20*h - 40*i)/((x + 1)*(x - 1)
*(x - 2))
```

Mupad [B] (verification not implemented)

Time = 8.69 (sec) , antiderivative size = 170, normalized size of antiderivative = 0.96

$$\int \frac{(2+x)\left(d+ex+fx^2+gx^3+hx^4+ix^5\right)}{\left(4-5x^2+x^4\right)^2}\,dx$$
$$= \ln(x-1)\left(\frac{d}{18}+\frac{5e}{36}+\frac{2f}{9}+\frac{11g}{36}+\frac{7h}{18}+\frac{17i}{36}\right) + \ln(x+2)\left(\frac{d}{144}-\frac{e}{72}+\frac{f}{36}-\frac{g}{18}+\frac{h}{9}-\frac{2i}{9}\right) + \ln(x+1)\left(\frac{d}{54}+\frac{e}{108}-\frac{f}{27}+\frac{7g}{108}-\frac{5h}{54}+\frac{13i}{108}\right) - \ln(x-2)\left(\frac{35d}{432}+\frac{29e}{216}+\frac{23f}{108}+\frac{17g}{54}+\frac{11h}{27}+\frac{10i}{27}\right) - \frac{\left(-\frac{5d}{36}-\frac{e}{9}-\frac{2f}{9}-\frac{5g}{18}-\frac{5h}{9}-\frac{17i}{18}\right)x^2+\left(\frac{d}{6}+\frac{f}{6}+\frac{h}{6}\right)x+\frac{5d}{36}+\frac{5e}{18}+\frac{2f}{9}+\frac{4g}{9}+\frac{5h}{9}+\frac{10i}{9}}{-x^3+2x^2+x-2}$$

```
[In] int(((x + 2)*(d + e*x + f*x^2 + g*x^3 + h*x^4 + i*x^5))/(x^4 - 5*x^2 + 4)^2
,x)
```

```
[Out] log(x - 1)*(d/18 + (5*e)/36 + (2*f)/9 + (11*g)/36 + (7*h)/18 + (17*i)/36) +
 log(x + 2)*(d/144 - e/72 + f/36 - g/18 + h/9 - (2*i)/9) + log(x + 1)*(d/54
 + e/108 - f/27 + (7*g)/108 - (5*h)/54 + (13*i)/108) - log(x - 2)*((35*d)/4
32 + (29*e)/216 + (23*f)/108 + (17*g)/54 + (11*h)/27 + (10*i)/27) - ((5*d)/
36 + (5*e)/18 + (2*f)/9 + (4*g)/9 + (5*h)/9 + (10*i)/9 - x^2*((5*d)/36 + e/
9 + (2*f)/9 + (5*g)/18 + (5*h)/9 + (17*i)/18) + x*(d/6 + f/6 + h/6))/(x + 2
*x^2 - x^3 - 2)
```

3.103 $\int (d+ex+fx^2+gx^3)(a+bx^2+cx^4)^{3/2}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 717

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)^{3/2}dx =$$

$$-\frac{(18b^3cd-144abc^2d-8b^4f+57ab^2cf-84a^2c^2f)\,x\sqrt{a+bx^2+cx^4}}{315c^{5/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$

$$-\frac{3(b^2-4ac)\,(2ce-bg)\,(b+2cx^2)\sqrt{a+bx^2+cx^4}}{256c^3}$$

$$+\frac{x(9b^2cd+90ac^2d-4b^3f+9abcf+3c(9bcd-4b^2f+14acf)\,x^2)\sqrt{a+bx^2+cx^4}}{315c^2}$$

$$+\frac{(2ce-bg)\,(b+2cx^2)\left(a+bx^2+cx^4\right)^{3/2}}{32c^2}+\frac{x(3(3cd+bf)+7cfx^2)\left(a+bx^2+cx^4\right)^{3/2}}{63c}$$

$$+\frac{g(a+bx^2+cx^4)^{5/2}}{10c}+\frac{3(b^2-4ac)^2\,(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{512c^{7/2}}$$

$$+\frac{\sqrt[4]{a}(18b^3cd-144abc^2d-8b^4f+57ab^2cf-84a^2c^2f)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}(2\ldots}{315c^{11/4}\sqrt{a+bx^2+cx^4}}$$

$$-\frac{\sqrt[4]{a}\left(18b^3cd-144abc^2d-8b^4f+57ab^2cf-84a^2c^2f+\sqrt{a}\sqrt{c}(9b^2cd-180ac^2d-4b^3f+24abcf)\right)\left(\sqrt{a}+\ldots\right.}{630c^{11/4}\sqrt{a+bx^2+cx^4}}$$

```
[Out] 1/32*(-b*g+2*c*e)*(2*c*x^2+b)*(c*x^4+b*x^2+a)^(3/2)/c^2+1/63*x*(7*c*f*x^2+3
*b*f+9*c*d)*(c*x^4+b*x^2+a)^(3/2)/c+1/10*g*(c*x^4+b*x^2+a)^(5/2)/c+3/512*(-
4*a*c+b^2)^2*(-b*g+2*c*e)*arctanh(1/2*(2*c*x^2+b)/c^(1/2)/(c*x^4+b*x^2+a)^(
1/2))/c^(7/2)-3/256*(-4*a*c+b^2)*(-b*g+2*c*e)*(2*c*x^2+b)*(c*x^4+b*x^2+a)^(
1/2)/c^3+1/315*x*(9*b^2*c*d+90*a*c^2*d-4*b^3*f+9*a*b*c*f+3*c*(14*a*c*f-4*b^
2*f+9*b*c*d)*x^2)*(c*x^4+b*x^2+a)^(1/2)/c^2-1/315*(-84*a^2*c^2*f+57*a*b^2*c
```

```
*f-144*a*b*c^2*d-8*b^4*f+18*b^3*c*d)*x*(c*x^4+b*x^2+a)^(1/2)/c^(5/2)/(a^(1/
2)+x^2*c^(1/2))+1/315*a^(1/4)*(-84*a^2*c^2*f+57*a*b^2*c*f-144*a*b*c^2*d-8*b
^4*f+18*b^3*c*d)*(cos(2*arctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^
(1/4)*x/a^(1/4)))*EllipticE(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/
2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1
/2))^2)^(1/2)/c^(11/4)/(c*x^4+b*x^2+a)^(1/2)-1/630*a^(1/4)*(cos(2*arctan(c^
(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)*x/a^(1/4)))*EllipticF(sin(2
*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c
^(1/2))*(18*b^3*c*d-144*a*b*c^2*d-8*b^4*f+57*a*b^2*c*f-84*a^2*c^2*f+(24*a*b
*c*f-180*a*c^2*d-4*b^3*f+9*b^2*c*d)*a^(1/2)*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1
/2)+x^2*c^(1/2))^2)^(1/2)/c^(11/4)/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.39 (sec) , antiderivative size = 717, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}} = 0.312$, Rules used = {1687, 1190, 1211, 1117, 1209, 1261, 654, 626, 635, 212}

$$\int (d+ex+fx^2+gx^3)\left(a+bx^2+cx^4\right)^{3/2}dx =$$

$$-\frac{\sqrt[4]{a}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\left(-84a^2c^2f+57ab^2cf+\sqrt{a}\sqrt{c}(24abcf-180ac^2d-4b^3f+9b^2cd)-144abc\right.}{630c^{11/4}\sqrt{a+bx^2+cx^4}}$$

$$+\frac{\sqrt[4]{a}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}(-84a^2c^2f+57ab^2cf-144abc^2d-8b^4f+18b^3cd)E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2\right.\right.}{315c^{11/4}\sqrt{a+bx^2+cx^4}}$$

$$-\frac{x\sqrt{a+bx^2+cx^4}(-84a^2c^2f+57ab^2cf-144abc^2d-8b^4f+18b^3cd)}{315c^{5/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$

$$+\frac{3(b^2-4ac)^2(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{512c^{7/2}}$$

$$-\frac{3(b^2-4ac)(b+2cx^2)\sqrt{a+bx^2+cx^4}(2ce-bg)}{256c^3}$$

$$+\frac{x\sqrt{a+bx^2+cx^4}(3cx^2(14acf-4b^2f+9bcd)+9abcf+90ac^2d-4b^3f+9b^2cd)}{315c^2}$$

$$+\frac{(b+2cx^2)(a+bx^2+cx^4)^{3/2}(2ce-bg)}{32c^2}$$

$$+\frac{x(a+bx^2+cx^4)^{3/2}(3(bf+3cd)+7cfx^2)}{63c}+\frac{g(a+bx^2+cx^4)^{5/2}}{10c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] -1/315*((18*b^3*c*d - 144*a*b*c^2*d - 8*b^4*f + 57*a*b^2*c*f - 84*a^2*c^2*f
)*x*Sqrt[a + b*x^2 + c*x^4])/(c^(5/2)*(Sqrt[a] + Sqrt[c]*x^2)) - (3*(b^2 -
```

```
4*a*c)*(2*c*e - b*g)*(b + 2*c*x^2)*Sqrt[a + b*x^2 + c*x^4])/(256*c^3) + (x*
(9*b^2*c*d + 90*a*c^2*d - 4*b^3*f + 9*a*b*c*f + 3*c*(9*b*c*d - 4*b^2*f + 14
*a*c*f)*x^2)*Sqrt[a + b*x^2 + c*x^4])/(315*c^2) + ((2*c*e - b*g)*(b + 2*c*x
^2)*(a + b*x^2 + c*x^4)^(3/2))/(32*c^2) + (x*(3*(3*c*d + b*f) + 7*c*f*x^2)*
(a + b*x^2 + c*x^4)^(3/2))/(63*c) + (g*(a + b*x^2 + c*x^4)^(5/2))/(10*c) +
(3*(b^2 - 4*a*c)^2*(2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/(2*Sqrt[c]*Sqrt[a +
b*x^2 + c*x^4])])/(512*c^(7/2)) + (a^(1/4)*(18*b^3*c*d - 144*a*b*c^2*d - 8*
b^4*f + 57*a*b^2*c*f - 84*a^2*c^2*f)*(Sqrt[a] + Sqrt[c]*x^2)*Sqrt[(a + b*x^
2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticE[2*ArcTan[(c^(1/4)*x)/a^(1/4
)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(315*c^(11/4)*Sqrt[a + b*x^2 + c*x^4]) -
(a^(1/4)*(18*b^3*c*d - 144*a*b*c^2*d - 8*b^4*f + 57*a*b^2*c*f - 84*a^2*c^2*
f + Sqrt[a]*Sqrt[c]*(9*b^2*c*d - 180*a*c^2*d - 4*b^3*f + 24*a*b*c*f))*(Sqrt
[a] + Sqrt[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*Elli
pticF[2*ArcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(630*c^(
11/4)*Sqrt[a + b*x^2 + c*x^4])
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 626

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^p/(2*c*(2*p + 1))), x] - Dist[p*((b^2 - 4*a*c)/(2*c*(2*
p + 1))), Int[(a + b*x + c*x^2)^(p - 1), x], x] /; FreeQ[{a, b, c}, x] && N
eQ[b^2 - 4*a*c, 0] && GtQ[p, 0] && IntegerQ[4*p]
```

Rule 635

```
Int[1/Sqrt[(a_) + (b_.)*(x_) + (c_.)*(x_)^2], x_Symbol] :> Dist[2, Subst[In
t[1/(4*c - x^2), x], x, (b + 2*c*x)/Sqrt[a + b*x + c*x^2]], x] /; FreeQ[{a,
 b, c}, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 654

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[e*((a + b*x + c*x^2)^(p + 1)/(2*c*(p + 1))), x] + Dist[(2*c*d - b
*e)/(2*c), Int[(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c, d, e, p}, x]
&& NeQ[2*c*d - b*e, 0] && NeQ[p, -1]
```

Rule 1117

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbol] :> With[{q = Rt[c
/a, 4]}, Simp[(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*x^2)^2)]/
(2*q*Sqrt[a + b*x^2 + c*x^4]))*EllipticF[2*ArcTan[q*x], 1/2 - b*(q^2/(4*c))
```

```
], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[c/a]
```

Rule 1190

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(2*b*e*p + c*d*(4*p + 3) + c*e*(4*p + 1)*x^2)*((a + b*x^2 + c
*x^4)^p/(c*(4*p + 1)*(4*p + 3))), x] + Dist[2*(p/(c*(4*p + 1)*(4*p + 3))),
Int[Simp[2*a*c*d*(4*p + 3) - a*b*e + (2*a*c*e*(4*p + 1) + b*c*d*(4*p + 3) -
 b^2*e*(2*p + 1))*x^2, x]*(a + b*x^2 + c*x^4)^(p - 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
GtQ[p, 0] && FractionQ[p] && IntegerQ[2*p]
```

Rule 1209

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 4]}, Simp[(-d)*x*(Sqrt[a + b*x^2 + c*x^4]/(a*(1 + q
^2*x^2))), x] + Simp[d*(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*
x^2)^2)]/(q*Sqrt[a + b*x^2 + c*x^4]))*EllipticE[2*ArcTan[q*x], 1/2 - b*(q^2
/(4*c))], x] /; EqQ[e + d*q^2, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[c/a]
```

Rule 1211

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 2]}, Dist[(e + d*q)/q, Int[1/Sqrt[a + b*x^2 + c*x^4
], x], x] - Dist[e/q, Int[(1 - q*x^2)/Sqrt[a + b*x^2 + c*x^4], x], x] /; Ne
Q[e + d*q, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[
c/a]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \left(d+fx^2\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx + \int x\left(e+gx^2\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx$$

$$= \frac{x\left(3(3cd+bf)+7cfx^2\right)\left(a+bx^2+cx^4\right)^{3/2}}{63c} + \frac{1}{2}\text{Subst}\left(\int (e+gx)\left(a+bx+cx^2\right)^{3/2}\,dx, x, x^2\right) + \frac{\int \left(a(18cd-bf)+\left(9bcd-4b^2f+14acf\right)x^2\right)\sqrt{a+bx^2+cx^4}\,dx}{21c}$$

$$= \frac{x\left(9b^2cd+90ac^2d-4b^3f+9abcf+3c\left(9bcd-4b^2f+14acf\right)x^2\right)\sqrt{a+bx^2+cx^4}}{315c^2} + \frac{x\left(3(3cd+bf)+7cfx^2\right)\left(a+bx^2+cx^4\right)^{3/2}}{63c} + \frac{g\left(a+bx^2+cx^4\right)^{5/2}}{10c} + \frac{\int \frac{-a\left(9b^2cd-180ac^2d-4b^3f+24abcf\right)+\left(-18b^3cd+144abc^2d+8b^4f-57ab^2cf+84a^2c^2f\right)x^2}{\sqrt{a+bx^2+cx^4}}\,dx}{315c^2} + \frac{(2ce-bg)\text{Subst}\left(\int \left(a+bx+cx^2\right)^{3/2}\,dx, x, x^2\right)}{4c}$$

$$= \frac{x\left(9b^2cd+90ac^2d-4b^3f+9abcf+3c\left(9bcd-4b^2f+14acf\right)x^2\right)\sqrt{a+bx^2+cx^4}}{315c^2} + \frac{(2ce-bg)\left(b+2cx^2\right)\left(a+bx^2+cx^4\right)^{3/2}}{32c^2} + \frac{x\left(3(3cd+bf)+7cfx^2\right)\left(a+bx^2+cx^4\right)^{3/2}}{63c} + \frac{g\left(a+bx^2+cx^4\right)^{5/2}}{10c} + \frac{\left(\sqrt{a}\left(18b^3cd-144abc^2d-8b^4f+57ab^2cf-84a^2c^2f\right)\right)\int \frac{1-\frac{\sqrt{c}x^2}{\sqrt{a}}}{\sqrt{a+bx^2+cx^4}}\,dx}{315c^{5/2}} - \frac{\left(\sqrt{a}\left(18b^3cd-144abc^2d-8b^4f+57ab^2cf-84a^2c^2f+\sqrt{a}\sqrt{c}\left(9b^2cd-180ac^2d-4b^3f+24ab\right.\right.\right.}{315c^{5/2}} - \frac{\left(3\left(b^2-4ac\right)(2ce-bg)\right)\text{Subst}\left(\int \sqrt{a+bx+cx^2}\,dx, x, x^2\right)}{64c^2}$$

$$= -\frac{(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\,x\sqrt{a+bx^2+cx^4}}{315c^{5/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$
$$-\frac{3(b^2-4ac)\,(2ce-bg)\,(b+2cx^2)\sqrt{a+bx^2+cx^4}}{256c^3}$$
$$+\frac{x(9b^2cd + 90ac^2d - 4b^3f + 9abcf + 3c(9bcd - 4b^2f + 14acf)\,x^2)\sqrt{a+bx^2+cx^4}}{315c^2}$$
$$+\frac{(2ce-bg)\,(b+2cx^2)\,(a+bx^2+cx^4)^{3/2}}{32c^2}$$
$$+\frac{x(3(3cd+bf)+7cfx^2)\,(a+bx^2+cx^4)^{3/2}}{63c}+\frac{g(a+bx^2+cx^4)^{5/2}}{10c}$$
$$+\frac{\sqrt[4]{a}(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}\cdots}{\sqrt[4]{\cdots}}\right.\right.}{315c^{11/4}\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt[4]{a}\left(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f + \sqrt{a}\sqrt{c}(9b^2cd - 180ac^2d - 4b^3f + 24abcf\cdots\right.}{630c^{11/4}\sqrt{a+bx^2+cx^4}\cdots}$$
$$+\frac{\left(3(b^2-4ac)^2\,(2ce-bg)\right)\text{Subst}\left(\int\frac{1}{\sqrt{a+bx+cx^2}}\,dx, x, x^2\right)}{512c^3}$$

$$= -\frac{(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\,x\sqrt{a+bx^2+cx^4}}{315c^{5/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$
$$-\frac{3(b^2-4ac)\,(2ce-bg)\,(b+2cx^2)\sqrt{a+bx^2+cx^4}}{256c^3}$$
$$+\frac{x(9b^2cd + 90ac^2d - 4b^3f + 9abcf + 3c(9bcd - 4b^2f + 14acf)\,x^2)\sqrt{a+bx^2+cx^4}}{315c^2}$$
$$+\frac{(2ce-bg)\,(b+2cx^2)\,(a+bx^2+cx^4)^{3/2}}{32c^2}$$
$$+\frac{x(3(3cd+bf)+7cfx^2)\,(a+bx^2+cx^4)^{3/2}}{63c}+\frac{g(a+bx^2+cx^4)^{5/2}}{10c}$$
$$+\frac{\sqrt[4]{a}(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}\cdots}{\sqrt[4]{\cdots}}\right.\right.}{315c^{11/4}\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt[4]{a}\left(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f + \sqrt{a}\sqrt{c}(9b^2cd - 180ac^2d - 4b^3f + 24abcf\cdots\right.}{630c^{11/4}\sqrt{a+bx^2+cx^4}\cdots}$$
$$+\frac{\left(3(b^2-4ac)^2\,(2ce-bg)\right)\text{Subst}\left(\int\frac{1}{4c-x^2}\,dx, x, \frac{b+2cx^2}{\sqrt{a+bx^2+cx^4}}\right)}{256c^3}$$

$$= -\frac{(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\,x\sqrt{a + bx^2 + cx^4}}{315c^{5/2}\left(\sqrt{a} + \sqrt{c}x^2\right)}$$
$$- \frac{3(b^2 - 4ac)\,(2ce - bg)\,(b + 2cx^2)\sqrt{a + bx^2 + cx^4}}{256c^3}$$
$$+ \frac{x(9b^2cd + 90ac^2d - 4b^3f + 9abcf + 3c(9bcd - 4b^2f + 14acf)\,x^2)\sqrt{a + bx^2 + cx^4}}{315c^2}$$
$$+ \frac{(2ce - bg)\,(b + 2cx^2)\,(a + bx^2 + cx^4)^{3/2}}{32c^2}$$
$$+ \frac{x(3(3cd + bf) + 7cfx^2)\,(a + bx^2 + cx^4)^{3/2}}{63c} + \frac{g(a + bx^2 + cx^4)^{5/2}}{10c}$$
$$+ \frac{3(b^2 - 4ac)^2\,(2ce - bg)\tanh^{-1}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{512c^{7/2}}$$
$$+ \frac{\sqrt[4]{a}(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f)\left(\sqrt{a} + \sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\right.\right.}{315c^{11/4}\sqrt{a + bx^2 + cx^4}}$$
$$- \frac{\sqrt[4]{a}\left(18b^3cd - 144abc^2d - 8b^4f + 57ab^2cf - 84a^2c^2f + \sqrt{a}\sqrt{c}(9b^2cd - 180ac^2d - 4b^3f + 24abc\right.}{630c^{11/4}\sqrt{a + bx^2 + cx^4}}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 11.69 (sec) , antiderivative size = 2588, normalized size of antiderivative = 3.61

$$\int \left(d + ex + fx^2 + gx^3\right)\left(a + bx^2 + cx^4\right)^{3/2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)*(a + b*x^2 + c*x^4)^(3/2),x]

[Out] (-2*Sqrt[c]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*(a + b*x^2 + c*x^4)*(-945*b^4*g
 + 2*b^3*c*(945*e + x*(512*f + 315*g*x)) - 12*b^2*c*(-525*a*g + c*x*(192*d
+ 105*e*x + 64*f*x^2 + 42*g*x^3)) - 8*b*c^2*(3*a*(525*e + 256*f*x + 147*g*x
^2) + 2*c*x^3*(1152*d + 945*e*x + 800*f*x^2 + 693*g*x^3)) - 16*c^2*(504*a^2
*g + 2*c^2*x^5*(360*d + 7*x*(45*e + 40*f*x + 36*g*x^2)) + a*c*x*(2160*d + 7
*x*(225*e + 16*x*(11*f + 9*g*x))))) + (2304*I)*Sqrt[2]*b^3*c^(3/2)*(b - Sqr
t[b^2 - 4*a*c])*d*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*
a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*(EllipticE[I*ArcSinh[Sqr
t[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[
b^2 - 4*a*c])] - EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])
]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])]) + (18432*I)*Sqrt[2]
*a*b*c^(5/2)*(-b + Sqrt[b^2 - 4*a*c])*d*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x
^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*(E
llipticE[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^
```

```
2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(
b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])
]) + (7296*I)*Sqrt[2]*a*b^2*c^(3/2)*(b - Sqrt[b^2 - 4*a*c])*f*Sqrt[(b + Sqr
t[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b -
Sqrt[b^2 - 4*a*c])]*(EllipticE[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a
*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - EllipticF[I*Ar
cSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(
b - Sqrt[b^2 - 4*a*c])]) + (1024*I)*Sqrt[2]*b^4*Sqrt[c]*(-b + Sqrt[b^2 - 4*
a*c])*f*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqr
t[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*(EllipticE[I*ArcSinh[Sqrt[2]*Sqrt[
c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*
c])] - EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b +
 Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])]) + (10752*I)*Sqrt[2]*a^2*c^(5/
2)*(-b + Sqrt[b^2 - 4*a*c])*f*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + S
qrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*(EllipticE[I
*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c]
)/(b - Sqrt[b^2 - 4*a*c])] - EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b
^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])]) + (2304
*I)*Sqrt[2]*a*b^2*c^(5/2)*d*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqr
t[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*EllipticF[I*Ar
cSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(
b - Sqrt[b^2 - 4*a*c])] - (46080*I)*Sqrt[2]*a^2*c^(7/2)*d*Sqrt[(b + Sqrt[b^
2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt
[b^2 - 4*a*c])]*EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]
*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - (1024*I)*Sqrt[2]*a*
b^3*c^(3/2)*f*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c]
)]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*EllipticF[I*ArcSinh[Sqrt[2]*
Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 -
 4*a*c])] + (6144*I)*Sqrt[2]*a^2*b*c^(5/2)*f*Sqrt[(b + Sqrt[b^2 - 4*a*c] +
2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c*x^2)/(b - Sqrt[b^2 - 4*a*c]
)]*EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqr
t[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - 1890*b^4*c*Sqrt[c/(b + Sqrt[b^2
- 4*a*c])]*e*Sqrt[a + b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b
*x^2 + c*x^4]] + 15120*a*b^2*c^2*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*e*Sqrt[a +
 b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4]] - 3024
0*a^2*c^3*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*e*Sqrt[a + b*x^2 + c*x^4]*Log[b +
 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4]] + 945*b^5*Sqrt[c/(b + Sqrt[b^
2 - 4*a*c])]*g*Sqrt[a + b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[a +
 b*x^2 + c*x^4]] - 7560*a*b^3*c*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*g*Sqrt[a +
b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4]] + 15120
*a^2*b*c^2*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*g*Sqrt[a + b*x^2 + c*x^4]*Log[b
+ 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4]])/(161280*c^(7/2)*Sqrt[c/(b +
 Sqrt[b^2 - 4*a*c])]*Sqrt[a + b*x^2 + c*x^4])
```

Maple [A] (verified)

Time = 5.79 (sec) , antiderivative size = 1209, normalized size of antiderivative = 1.69

method	result	size
risch	Expression too large to display	1209
elliptic	Expression too large to display	1376
default	Expression too large to display	1580

```
[In] int((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/80640/c^3*(8064*c^4*g*x^8+8960*c^4*f*x^7+11088*b*c^3*g*x^6+10080*c^4*e*x^
6+12800*b*c^3*f*x^5+11520*c^4*d*x^5+16128*a*c^3*g*x^4+504*b^2*c^2*g*x^4+151
20*b*c^3*e*x^4+19712*a*c^3*f*x^3+768*b^2*c^2*f*x^3+18432*b*c^3*d*x^3+3528*a
*b*c^2*g*x^2+25200*a*c^3*e*x^2-630*b^3*c*g*x^2+1260*b^2*c^2*e*x^2+6144*a*b*
c^2*f*x+34560*a*c^3*d*x-1024*b^3*c*f*x+2304*b^2*c^2*d*x+8064*a^2*c^2*g-6300
*a*b^2*c*g+12600*a*b*c^2*e+945*b^4*g-1890*b^3*c*e)*(c*x^4+b*x^2+a)^(1/2)-1/
80640/c^3*(-11520*a^2*c^3*d*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*
(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/
2)/(c*x^4+b*x^2+a)^(1/2)*EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a
)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))+1536*a^2*b*c^2*f*2^(
1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^
(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)*Ellipt
icF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+
b^2)^(1/2))/a/c)^(1/2))-256*a*b^3*c*f*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(
1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/
a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)*EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2
)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))+576*a*b^2*
c^2*d*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2)
)/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/
2)*EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b
+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))+1/2*(21504*a^2*c^3*f-14592*a*b^2*c^2*f+368
64*a*b*c^3*d+2048*b^4*c*f-4608*b^3*c^2*d)*a*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2)
)/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(
1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)/(b+(-4*a*c+b^2)^(1/2))*(EllipticF(
1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)
^(1/2))/a/c)^(1/2))-EllipticE(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/
2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2)))-1/2*(-15120*a^2*b*c^2*g+
30240*a^2*c^3*e+7560*a*b^3*c*g-15120*a*b^2*c^2*e-945*b^5*g+1890*b^4*c*e)*ln
((1/2*b+c*x^2)/c^(1/2)+(c*x^4+b*x^2+a)^(1/2))/c^(1/2))
```

Fricas [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 911, normalized size of antiderivative = 1.27

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx =$$

$$-\frac{512\sqrt{\frac{1}{2}}\left(\left(18\left(b^3c^2-8\,abc^3\right)d-\left(8\,b^4c-57\,ab^2c^2+84\,a^2c^3\right)f\right)x\sqrt{\frac{b^2-4\,ac}{c^2}}-\left(18\left(b^4c-8\,ab^2c^2\right)d-\left(8\,b^5-57\right.\right.\right.}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(3/2),x, algorithm="fricas")
```

```
[Out] -1/322560*(512*sqrt(1/2)*((18*(b^3*c^2 - 8*a*b*c^3)*d - (8*b^4*c - 57*a*b^2
*c^2 + 84*a^2*c^3)*f)*x*sqrt((b^2 - 4*a*c)/c^2) - (18*(b^4*c - 8*a*b^2*c^2)
*d - (8*b^5 - 57*a*b^3*c + 84*a^2*b*c^2)*f)*x)*sqrt(c)*sqrt((c*sqrt((b^2 -
4*a*c)/c^2) - b)/c)*elliptic_e(arcsin(sqrt(1/2)*sqrt((c*sqrt((b^2 - 4*a*c)/
c^2) - b)/c)/x), 1/2*(b*c*sqrt((b^2 - 4*a*c)/c^2) + b^2 - 2*a*c)/(a*c)) - 5
12*sqrt(1/2)*((9*(2*b^3*c^2 + 20*a*c^4 - (16*a*b + b^2)*c^3)*d - (8*b^4*c +
 12*(7*a^2 + 2*a*b)*c^3 - (57*a*b^2 + 4*b^3)*c^2)*f)*x*sqrt((b^2 - 4*a*c)/c
^2) - (9*(2*b^4*c - 20*a*b*c^3 - (16*a*b^2 - b^3)*c^2)*d - (8*b^5 + 12*(7*a
^2*b - 2*a*b^2)*c^2 - (57*a*b^3 - 4*b^4)*c)*f)*x)*sqrt(c)*sqrt((c*sqrt((b^2
 - 4*a*c)/c^2) - b)/c)*elliptic_f(arcsin(sqrt(1/2)*sqrt((c*sqrt((b^2 - 4*a*
c)/c^2) - b)/c)/x), 1/2*(b*c*sqrt((b^2 - 4*a*c)/c^2) + b^2 - 2*a*c)/(a*c))
+ 945*(2*(b^4*c - 8*a*b^2*c^2 + 16*a^2*c^3)*e - (b^5 - 8*a*b^3*c + 16*a^2*b
*c^2)*g)*sqrt(c)*x*log(8*c^2*x^4 + 8*b*c*x^2 + b^2 - 4*sqrt(c*x^4 + b*x^2 +
 a)*(2*c*x^2 + b)*sqrt(c) + 4*a*c) - 4*(8064*c^5*g*x^9 + 8960*c^5*f*x^8 + 1
008*(10*c^5*e + 11*b*c^4*g)*x^7 + 1280*(9*c^5*d + 10*b*c^4*f)*x^6 + 504*(30
*b*c^4*e + (b^2*c^3 + 32*a*c^4)*g)*x^5 + 256*(72*b*c^4*d + (3*b^2*c^3 + 77*
a*c^4)*f)*x^4 + 126*(10*(b^2*c^3 + 20*a*c^4)*e - (5*b^3*c^2 - 28*a*b*c^3)*g
)*x^3 + 256*(9*(b^2*c^3 + 15*a*c^4)*d - 4*(b^3*c^2 - 6*a*b*c^3)*f)*x^2 - 46
08*(b^3*c^2 - 8*a*b*c^3)*d + 256*(8*b^4*c - 57*a*b^2*c^2 + 84*a^2*c^3)*f -
63*(10*(3*b^3*c^2 - 20*a*b*c^3)*e - (15*b^4*c - 100*a*b^2*c^2 + 128*a^2*c^3
)*g)*x)*sqrt(c*x^4 + b*x^2 + a))/(c^4*x)
```

Sympy [F]

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx=\int \left(a+bx^2+cx^4\right)^{\frac{3}{2}}\left(d+ex+fx^2+gx^3\right)\,dx$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)*(c*x**4+b*x**2+a)**(3/2),x)
[Out] Integral((a + b*x**2 + c*x**4)**(3/2)*(d + e*x + f*x**2 + g*x**3), x)
```

Maxima [F]

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx=\int \left(cx^4+bx^2+a\right)^{\frac{3}{2}}\left(gx^3+fx^2+ex+d\right)dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxima")
[Out] integrate((c*x^4 + b*x^2 + a)^(3/2)*(g*x^3 + f*x^2 + e*x + d), x)
```

Giac [F]

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx=\int \left(cx^4+bx^2+a\right)^{\frac{3}{2}}\left(gx^3+fx^2+ex+d\right)dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac")
[Out] integrate((c*x^4 + b*x^2 + a)^(3/2)*(g*x^3 + f*x^2 + e*x + d), x)
```

Mupad [F(-1)]

Timed out.

$$\int \left(d+ex+fx^2+gx^3\right)\left(a+bx^2+cx^4\right)^{3/2}\,dx=\int \left(c\,x^4+b\,x^2+a\right)^{3/2}\left(g\,x^3+f\,x^2+e\,x+d\right)\,dx$$

```
[In] int((a + b*x^2 + c*x^4)^(3/2)*(d + e*x + f*x^2 + g*x^3),x)
[Out] int((a + b*x^2 + c*x^4)^(3/2)*(d + e*x + f*x^2 + g*x^3), x)
```

3.104 $\int (d+ex+fx^2+gx^3)\sqrt{a+bx^2+cx^4}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 505

$$\int (d+ex+fx^2+gx^3)\sqrt{a+bx^2+cx^4}\,dx=\frac{(5bcd-2b^2f+6acf)\,x\sqrt{a+bx^2+cx^4}}{15c^{3/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$
$$+\frac{(2ce-bg)\,(b+2cx^2)\sqrt{a+bx^2+cx^4}}{16c^2}+\frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c}$$
$$+\frac{g(a+bx^2+cx^4)^{3/2}}{6c}-\frac{(b^2-4ac)\,(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{32c^{5/2}}$$
$$-\frac{\sqrt[4]{a}(5bcd-2b^2f+6acf)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{15c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\sqrt[4]{a}\left(b+2\sqrt{a}\sqrt{c}\right)\left(5cd-2bf+3\sqrt{a}\sqrt{c}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\text{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2-\right.\right.}{30c^{7/4}\sqrt{a+bx^2+cx^4}}$$

```
[Out] 1/6*g*(c*x^4+b*x^2+a)^(3/2)/c-1/32*(-4*a*c+b^2)*(-b*g+2*c*e)*arctanh(1/2*(2
*c*x^2+b)/c^(1/2)/(c*x^4+b*x^2+a)^(1/2))/c^(5/2)+1/16*(-b*g+2*c*e)*(2*c*x^2
+b)*(c*x^4+b*x^2+a)^(1/2)/c^2+1/15*x*(3*c*f*x^2+b*f+5*c*d)*(c*x^4+b*x^2+a)^
(1/2)/c+1/15*(6*a*c*f-2*b^2*f+5*b*c*d)*x*(c*x^4+b*x^2+a)^(1/2)/c^(3/2)/(a^(
1/2)+x^2*c^(1/2))-1/15*a^(1/4)*(6*a*c*f-2*b^2*f+5*b*c*d)*(cos(2*arctan(c^(1
/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)*x/a^(1/4)))*EllipticE(sin(2*a
rctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c^(
1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2))^2)^(1/2)/c^(7/4)/(c*x^4+b*x^2+
a)^(1/2)+1/30*a^(1/4)*(cos(2*arctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arct
an(c^(1/4)*x/a^(1/4)))*EllipticF(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/
a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c^(1/2))*(b+2*a^(1/2)*c^(1/2))*(5*c*d-
2*b*f+3*f*a^(1/2)*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2))^2)^(1/2)/
c^(7/4)/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.18 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 10, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.312, Rules used = {1687, 1190, 1211, 1117, 1209, 1261, 654, 626, 635, 212}

$$\int \left(d+ex+fx^2+gx^3\right)\sqrt{a+bx^2+cx^4}\,dx = -\frac{\sqrt[4]{a}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}(6acf-2b^2f+5bcd)\,E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{15c^{7/4}\sqrt{a+bx^2+cx^4}} + \frac{\sqrt[4]{a}\left(2\sqrt{a}\sqrt{c}+b\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\left(3\sqrt{a}\sqrt{c}f-2bf+5cd\right)\text{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2\right.\right.}{30c^{7/4}\sqrt{a+bx^2+cx^4}} - \frac{(b^2-4ac)\,(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{32c^{5/2}} + \frac{x\sqrt{a+bx^2+cx^4}(6acf-2b^2f+5bcd)}{15c^{3/2}\left(\sqrt{a}+\sqrt{c}x^2\right)} + \frac{(b+2cx^2)\sqrt{a+bx^2+cx^4}(2ce-bg)}{16c^2} + \frac{x\sqrt{a+bx^2+cx^4}(bf+5cd+3cfx^2)}{15c} + \frac{g(a+bx^2+cx^4)^{3/2}}{6c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)*Sqrt[a + b*x^2 + c*x^4],x]
```

```
[Out] ((5*b*c*d - 2*b^2*f + 6*a*c*f)*x*Sqrt[a + b*x^2 + c*x^4])/(15*c^(3/2)*(Sqrt
[a] + Sqrt[c]*x^2)) + ((2*c*e - b*g)*(b + 2*c*x^2)*Sqrt[a + b*x^2 + c*x^4])
/(16*c^2) + (x*(5*c*d + b*f + 3*c*f*x^2)*Sqrt[a + b*x^2 + c*x^4])/(15*c) +
(g*(a + b*x^2 + c*x^4)^(3/2))/(6*c) - ((b^2 - 4*a*c)*(2*c*e - b*g)*ArcTanh[
(b + 2*c*x^2)/(2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4])])/(32*c^(5/2)) - (a^(1/4)
*(5*b*c*d - 2*b^2*f + 6*a*c*f)*(Sqrt[a] + Sqrt[c]*x^2)*Sqrt[(a + b*x^2 + c*
x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticE[2*ArcTan[(c^(1/4)*x)/a^(1/4)], (2
 - b/(Sqrt[a]*Sqrt[c]))/4])/(15*c^(7/4)*Sqrt[a + b*x^2 + c*x^4]) + (a^(1/4)
*(b + 2*Sqrt[a]*Sqrt[c])*(5*c*d - 2*b*f + 3*Sqrt[a]*Sqrt[c]*f)*(Sqrt[a] + S
qrt[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticF[2
*ArcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(30*c^(7/4)*Sqr
t[a + b*x^2 + c*x^4])
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 626

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol] :> Simp[(b + 2*c*x)
*((a + b*x + c*x^2)^p/(2*c*(2*p + 1))), x] - Dist[p*((b^2 - 4*a*c)/(2*c*(2*
```

```
p + 1))), Int[(a + b*x + c*x^2)^(p - 1), x], x] /; FreeQ[{a, b, c}, x] && N
eQ[b^2 - 4*a*c, 0] && GtQ[p, 0] && IntegerQ[4*p]
```

Rule 635

```
Int[1/Sqrt[(a_) + (b_.)*(x_) + (c_.)*(x_)^2], x_Symbol] :> Dist[2, Subst[In
t[1/(4*c - x^2), x], x, (b + 2*c*x)/Sqrt[a + b*x + c*x^2]], x] /; FreeQ[{a,
 b, c}, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 654

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[e*((a + b*x + c*x^2)^(p + 1)/(2*c*(p + 1))), x] + Dist[(2*c*d - b
*e)/(2*c), Int[(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c, d, e, p}, x]
&& NeQ[2*c*d - b*e, 0] && NeQ[p, -1]
```

Rule 1117

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbol] :> With[{q = Rt[c
/a, 4]}, Simp[(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*x^2)^2)]/
(2*q*Sqrt[a + b*x^2 + c*x^4]))*EllipticF[2*ArcTan[q*x], 1/2 - b*(q^2/(4*c))
], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[c/a]
```

Rule 1190

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(2*b*e*p + c*d*(4*p + 3) + c*e*(4*p + 1)*x^2)*((a + b*x^2 + c
*x^4)^p/(c*(4*p + 1)*(4*p + 3))), x] + Dist[2*(p/(c*(4*p + 1)*(4*p + 3))),
Int[Simp[2*a*c*d*(4*p + 3) - a*b*e + (2*a*c*e*(4*p + 1) + b*c*d*(4*p + 3) -
 b^2*e*(2*p + 1))*x^2, x]*(a + b*x^2 + c*x^4)^(p - 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
GtQ[p, 0] && FractionQ[p] && IntegerQ[2*p]
```

Rule 1209

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 4]}, Simp[(-d)*x*(Sqrt[a + b*x^2 + c*x^4]/(a*(1 + q
^2*x^2))), x] + Simp[d*(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*
x^2)^2)]/(q*Sqrt[a + b*x^2 + c*x^4]))*EllipticE[2*ArcTan[q*x], 1/2 - b*(q^2
/(4*c))], x] /; EqQ[e + d*q^2, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[c/a]
```

Rule 1211

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 2]}, Dist[(e + d*q)/q, Int[1/Sqrt[a + b*x^2 + c*x^4
```

```
], x], x] - Dist[e/q, Int[(1 - q*x^2)/Sqrt[a + b*x^2 + c*x^4], x], x] /; Ne
Q[e + d*q, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[
c/a]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(d+fx^2\right)\sqrt{a+bx^2+cx^4}\,dx+\int x\left(e+gx^2\right)\sqrt{a+bx^2+cx^4}\,dx \\
&= \frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c} \\
&\quad +\frac{1}{2}\text{Subst}\left(\int (e+gx)\sqrt{a+bx+cx^2}\,dx,x,x^2\right)+\frac{\int \frac{a(10cd-bf)+(5bcd-2b^2f+6acf)x^2}{\sqrt{a+bx^2+cx^4}}\,dx}{15c} \\
&= \frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c}+\frac{g(a+bx^2+cx^4)^{3/2}}{6c} \\
&\quad +\frac{\left(\sqrt{a}\left(b+2\sqrt{a}\sqrt{c}\right)\left(5cd-2bf+3\sqrt{a}\sqrt{c}f\right)\right)\int \frac{1}{\sqrt{a+bx^2+cx^4}}\,dx}{15c^{3/2}} \\
&\quad -\frac{\left(\sqrt{a}(5bcd-2b^2f+6acf)\right)\int \frac{1-\frac{\sqrt{c}x^2}{\sqrt{a}}}{\sqrt{a+bx^2+cx^4}}\,dx}{15c^{3/2}} \\
&\quad +\frac{(2ce-bg)\text{Subst}\left(\int \sqrt{a+bx+cx^2}\,dx,x,x^2\right)}{4c}
\end{aligned}$$

$$= \frac{(5bcd-2b^2f+6acf)\,x\sqrt{a+bx^2+cx^4}}{15c^{3/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}+\frac{(2ce-bg)\left(b+2cx^2\right)\sqrt{a+bx^2+cx^4}}{16c^2}$$
$$+\frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c}+\frac{g(a+bx^2+cx^4)^{3/2}}{6c}$$
$$-\frac{\sqrt[4]{a}(5bcd-2b^2f+6acf)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{15c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\sqrt[4]{a}\left(b+2\sqrt{a}\sqrt{c}\right)\left(5cd-2bf+3\sqrt{a}\sqrt{c}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\right.\right.}{30c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\left((b^2-4ac)(2ce-bg)\right)\text{Subst}\left(\int\frac{1}{\sqrt{a+bx+cx^2}}\,dx,x,x^2\right)}{32c^2}$$

$$= \frac{(5bcd-2b^2f+6acf)\,x\sqrt{a+bx^2+cx^4}}{15c^{3/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}+\frac{(2ce-bg)\left(b+2cx^2\right)\sqrt{a+bx^2+cx^4}}{16c^2}$$
$$+\frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c}+\frac{g(a+bx^2+cx^4)^{3/2}}{6c}$$
$$-\frac{\sqrt[4]{a}(5bcd-2b^2f+6acf)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{15c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\sqrt[4]{a}\left(b+2\sqrt{a}\sqrt{c}\right)\left(5cd-2bf+3\sqrt{a}\sqrt{c}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\right.\right.}{30c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\left((b^2-4ac)(2ce-bg)\right)\text{Subst}\left(\int\frac{1}{4c-x^2}\,dx,x,\frac{b+2cx^2}{\sqrt{a+bx^2+cx^4}}\right)}{16c^2}$$

$$= \frac{(5bcd-2b^2f+6acf)\,x\sqrt{a+bx^2+cx^4}}{15c^{3/2}\left(\sqrt{a}+\sqrt{c}x^2\right)}+\frac{(2ce-bg)\left(b+2cx^2\right)\sqrt{a+bx^2+cx^4}}{16c^2}$$
$$+\frac{x(5cd+bf+3cfx^2)\sqrt{a+bx^2+cx^4}}{15c}+\frac{g(a+bx^2+cx^4)^{3/2}}{6c}$$
$$-\frac{(b^2-4ac)(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{32c^{5/2}}$$
$$-\frac{\sqrt[4]{a}(5bcd-2b^2f+6acf)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{15c^{7/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\sqrt[4]{a}\left(b+2\sqrt{a}\sqrt{c}\right)\left(5cd-2bf+3\sqrt{a}\sqrt{c}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\right.\right.}{30c^{7/4}\sqrt{a+bx^2+cx^4}}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 12.54 (sec) , antiderivative size = 661, normalized size of antiderivative = 1.31

$$\int (d + ex + fx^2 + gx^3) \sqrt{a + bx^2 + cx^4} \, dx$$

$$= \frac{2\sqrt{c}(a + bx^2 + cx^4) (-15b^2g + 2bc(15e + x(8f + 5gx)) + 4c(10ag + cx(20d + x(15e + 2x(6f + 5gx)))))}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)*Sqrt[a + b*x^2 + c*x^4],x]
```

```
[Out] (2*Sqrt[c]*(a + b*x^2 + c*x^4)*(-15*b^2*g + 2*b*c*(15*e + x*(8*f + 5*g*x))
+ 4*c*(10*a*g + c*x*(20*d + x*(15*e + 2*x*(6*f + 5*g*x))))) + ((-8*I)*Sqrt[
2]*Sqrt[c]*(-b + Sqrt[b^2 - 4*a*c])*(-5*b*c*d + 2*b^2*f - 6*a*c*f)*Sqrt[(b
- Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*Sqrt[(b + Sqrt[b^2
- 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*EllipticE[I*ArcSinh[Sqrt[2]*Sq
rt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4
*a*c])] + (8*I)*Sqrt[2]*Sqrt[c]*(-2*b^3*f + b*c*(-5*Sqrt[b^2 - 4*a*c]*d + 8
*a*f) + b^2*(5*c*d + 2*Sqrt[b^2 - 4*a*c]*f) - 2*a*c*(10*c*d + 3*Sqrt[b^2 -
4*a*c]*f))*Sqrt[(b - Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*
Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*EllipticF[I
*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c]
)/(b - Sqrt[b^2 - 4*a*c])] - 15*(b^2 - 4*a*c)*Sqrt[c/(b + Sqrt[b^2 - 4*a*c]
)]*(-2*c*e + b*g)*Sqrt[a + b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[
a + b*x^2 + c*x^4]])/Sqrt[c/(b + Sqrt[b^2 - 4*a*c])])/(480*c^(5/2)*Sqrt[a +
 b*x^2 + c*x^4])
```

Maple [A] (verified)

Time = 5.34 (sec) , antiderivative size = 620, normalized size of antiderivative = 1.23

method	result
elliptic	$\frac{g\,x^{4}\sqrt{c\,x^{4}+b\,x^{2}+a}}{6}+\frac{f\,x^{3}\sqrt{c\,x^{4}+b\,x^{2}+a}}{5}+\frac{\left(\frac{bg}{6}+ec\right)x^{2}\sqrt{c\,x^{4}+b\,x^{2}+a}}{4c}+\frac{\left(\frac{bf}{5}+cd\right)x\sqrt{c\,x^{4}+b\,x^{2}+a}}{3c}+\frac{\left(\frac{ag}{3}+be-\frac{3\left(\frac{bg}{6}+ec\right)b}{4c}\right)\sqrt{}}{2c}$
risch	$\frac{\left(40g\,x^{4}c^{2}+48f\,x^{3}c^{2}+10bcg\,x^{2}+60c^{2}e\,x^{2}+16bfxc+80c^{2}dx+40acg-15b^{2}g+30ebc\right)\sqrt{c\,x^{4}+b\,x^{2}+a}}{240c^{2}}-\frac{-\frac{40a\,c^{2}d\sqrt{2}\,\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^{2}}\right)}{a}}}{}}{}$
default	$d\left(\frac{x\sqrt{c\,x^{4}+b\,x^{2}+a}}{3}+\frac{a\sqrt{2}\,\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^{2}}\right)x^{2}}{a}}\,\sqrt{4+\frac{2\left(b+\sqrt{-4ac+b^{2}}\right)x^{2}}{a}}\,F\left(\frac{x\sqrt{2}\,\sqrt{\frac{-b+\sqrt{-4ac+b^{2}}}{a}}}{2},\frac{\sqrt{-4+\frac{2b\left(b+\sqrt{-4ac+b^{2}}\right)}{ac}}}{2}\right)}{6\sqrt{\frac{-b+\sqrt{-4ac+b^{2}}}{a}}\,\sqrt{c\,x^{4}+b\,x^{2}+a}}\right.$

```
[In] int((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/6*g*x^4*(c*x^4+b*x^2+a)^(1/2)+1/5*f*x^3*(c*x^4+b*x^2+a)^(1/2)+1/4*(1/6*b*
g+e*c)/c*x^2*(c*x^4+b*x^2+a)^(1/2)+1/3*(1/5*b*f+c*d)/c*x*(c*x^4+b*x^2+a)^(1
/2)+1/2*(1/3*a*g+b*e-3/4*(1/6*b*g+e*c)/c*b)/c*(c*x^4+b*x^2+a)^(1/2)+1/4*(d*
a-1/3*(1/5*b*f+c*d)/c*a)*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b
+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/
(c*x^4+b*x^2+a)^(1/2)*EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(
1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))+1/2*(a*e-1/2*(1/6*b*g+e
*c)/c*a-1/2*(1/3*a*g+b*e-3/4*(1/6*b*g+e*c)/c*b)/c*b)*ln((2*c*x^2+b)/c^(1/2)
+2*(c*x^4+b*x^2+a)^(1/2))/c^(1/2)-1/2*(2/5*a*f+b*d-2/3*(1/5*b*f+c*d)/c*b)*a
*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x
^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)/(b
+(-4*a*c+b^2)^(1/2))*(EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(
1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))-EllipticE(1/2*x*2^(1/2)
*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^
(1/2)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 574, normalized size of antiderivative = 1.14

$$\int \left(d+ex+fx^2+gx^3\right)\sqrt{a+bx^2+cx^4}\,dx$$

$$= \frac{32\sqrt{\frac{1}{2}}\left(\left(5\,bc^2d-2\,(b^2c-3\,ac^2)f\right)x\sqrt{\frac{b^2-4\,ac}{c^2}}-\left(5\,b^2cd-2\,(b^3-3\,abc)f\right)x\right)\sqrt{c}\sqrt{\frac{c\sqrt{\frac{b^2-4\,ac}{c^2}}-b}{c}}E(\arcsin\left(\frac{\sqrt{}}{}\right.}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(1/2),x, algorithm="fricas")
```

```
[Out] 1/960*(32*sqrt(1/2)*((5*b*c^2*d - 2*(b^2*c - 3*a*c^2)*f)*x*sqrt((b^2 - 4*a*
c)/c^2) - (5*b^2*c*d - 2*(b^3 - 3*a*b*c)*f)*x)*sqrt(c)*sqrt((c*sqrt((b^2 -
4*a*c)/c^2) - b)/c)*elliptic_e(arcsin(sqrt(1/2)*sqrt((c*sqrt((b^2 - 4*a*c)/
c^2) - b)/c)/x), 1/2*(b*c*sqrt((b^2 - 4*a*c)/c^2) + b^2 - 2*a*c)/(a*c)) - 3
2*sqrt(1/2)*((5*(b*c^2 - 2*c^3)*d - (2*b^2*c - (6*a + b)*c^2)*f)*x*sqrt((b^
2 - 4*a*c)/c^2) - (5*(b^2*c + 2*b*c^2)*d - (2*b^3 - (6*a*b - b^2)*c)*f)*x)*
sqrt(c)*sqrt((c*sqrt((b^2 - 4*a*c)/c^2) - b)/c)*elliptic_f(arcsin(sqrt(1/2)
*sqrt((c*sqrt((b^2 - 4*a*c)/c^2) - b)/c)/x), 1/2*(b*c*sqrt((b^2 - 4*a*c)/c^
2) + b^2 - 2*a*c)/(a*c)) + 15*(2*(b^2*c - 4*a*c^2)*e - (b^3 - 4*a*b*c)*g)*s
qrt(c)*x*log(8*c^2*x^4 + 8*b*c*x^2 + b^2 - 4*sqrt(c*x^4 + b*x^2 + a)*(2*c*x
^2 + b)*sqrt(c) + 4*a*c) + 4*(40*c^3*g*x^5 + 48*c^3*f*x^4 + 80*b*c^2*d + 10
*(6*c^3*e + b*c^2*g)*x^3 + 16*(5*c^3*d + b*c^2*f)*x^2 - 32*(b^2*c - 3*a*c^2
)*f + 5*(6*b*c^2*e - (3*b^2*c - 8*a*c^2)*g)*x)*sqrt(c*x^4 + b*x^2 + a))/(c^
3*x)
```

Sympy [F]

$$\int \left(d+ex+fx^2+gx^3\right)\sqrt{a+bx^2+cx^4}\,dx = \int \sqrt{a+bx^2+cx^4}\left(d+ex+fx^2+gx^3\right)\,dx$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)*(c*x**4+b*x**2+a)**(1/2),x)
```

```
[Out] Integral(sqrt(a + b*x**2 + c*x**4)*(d + e*x + f*x**2 + g*x**3), x)
```

Maxima [F]

$$\int (d+ex+fx^2+gx^3)\sqrt{a+bx^2+cx^4}\,dx=\int \sqrt{cx^4+bx^2+a}(gx^3+fx^2+ex+d)\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(1/2),x, algorithm="maxima")
[Out] integrate(sqrt(c*x^4 + b*x^2 + a)*(g*x^3 + f*x^2 + e*x + d), x)
```

Giac [F]

$$\int (d+ex+fx^2+gx^3)\sqrt{a+bx^2+cx^4}\,dx=\int \sqrt{cx^4+bx^2+a}(gx^3+fx^2+ex+d)\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)*(c*x^4+b*x^2+a)^(1/2),x, algorithm="giac")
[Out] integrate(sqrt(c*x^4 + b*x^2 + a)*(g*x^3 + f*x^2 + e*x + d), x)
```

Mupad [F(-1)]

Timed out.

$$\int (d+ex+fx^2+gx^3)\sqrt{a+bx^2+cx^4}\,dx=\int \sqrt{c\,x^4+b\,x^2+a}\,\left(g\,x^3+f\,x^2+e\,x+d\right)\,dx$$

```
[In] int((a + b*x^2 + c*x^4)^(1/2)*(d + e*x + f*x^2 + g*x^3),x)
[Out] int((a + b*x^2 + c*x^4)^(1/2)*(d + e*x + f*x^2 + g*x^3), x)
```

3.105 $\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 359

$$\begin{aligned}
\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx
&= \frac{g\sqrt{a+bx^2+cx^4}}{2c} + \frac{fx\sqrt{a+bx^2+cx^4}}{\sqrt{c}\left(\sqrt{a}+\sqrt{c}x^2\right)} + \frac{(2ce-bg)\operatorname{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{4c^{3/2}} \\
&\quad - \frac{\sqrt[4]{a}f\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{c^{3/4}\sqrt{a+bx^2+cx^4}} \\
&\quad + \frac{\sqrt[4]{a}\left(\frac{\sqrt{c}d}{\sqrt{a}}+f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\operatorname{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2c^{3/4}\sqrt{a+bx^2+cx^4}}
\end{aligned}$$

```
[Out] 1/4*(-b*g+2*c*e)*arctanh(1/2*(2*c*x^2+b)/c^(1/2)/(c*x^4+b*x^2+a)^(1/2))/c^(
3/2)+1/2*g*(c*x^4+b*x^2+a)^(1/2)/c+f*x*(c*x^4+b*x^2+a)^(1/2)/c^(1/2)/(a^(1/
2)+x^2*c^(1/2))-a^(1/4)*f*(cos(2*arctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*
arctan(c^(1/4)*x/a^(1/4)))*EllipticE(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(
2-b/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)
+x^2*c^(1/2))^2)^(1/2)/c^(3/4)/(c*x^4+b*x^2+a)^(1/2)+1/2*a^(1/4)*(cos(2*arc
tan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)*x/a^(1/4)))*EllipticF
(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)
+x^2*c^(1/2))*(f+d*c^(1/2)/a^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2))^
2)^(1/2)/c^(3/4)/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 359, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.250, Rules used = {1687, 1211, 1117, 1209, 1261, 654, 635, 212}

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx$$
$$=\frac{\sqrt[4]{a}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\left(\frac{\sqrt{c}d}{\sqrt{a}}+f\right)\text{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt[4]{a}f\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{(2ce-bg)\text{arctanh}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{4c^{3/2}}+\frac{fx\sqrt{a+bx^2+cx^4}}{\sqrt{c}\left(\sqrt{a}+\sqrt{c}x^2\right)}+\frac{g\sqrt{a+bx^2+cx^4}}{2c}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/Sqrt[a + b*x^2 + c*x^4],x]
```

```
[Out] (g*Sqrt[a + b*x^2 + c*x^4])/(2*c) + (f*x*Sqrt[a + b*x^2 + c*x^4])/(Sqrt[c]*
(Sqrt[a] + Sqrt[c]*x^2)) + ((2*c*e - b*g)*ArcTanh[(b + 2*c*x^2)/(2*Sqrt[c]*
Sqrt[a + b*x^2 + c*x^4])])/(4*c^(3/2)) - (a^(1/4)*f*(Sqrt[a] + Sqrt[c]*x^2)
*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticE[2*ArcTan[(c^
(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(c^(3/4)*Sqrt[a + b*x^2 +
c*x^4]) + (a^(1/4)*((Sqrt[c]*d)/Sqrt[a] + f)*(Sqrt[a] + Sqrt[c]*x^2)*Sqrt[(
a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticF[2*ArcTan[(c^(1/4)*x
)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(2*c^(3/4)*Sqrt[a + b*x^2 + c*x^4
])
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 635

```
Int[1/Sqrt[(a_) + (b_.)*(x_) + (c_.)*(x_)^2], x_Symbol] :> Dist[2, Subst[In
t[1/(4*c - x^2), x], x, (b + 2*c*x)/Sqrt[a + b*x + c*x^2]], x] /; FreeQ[{a,
 b, c}, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 654

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[e*((a + b*x + c*x^2)^(p + 1)/(2*c*(p + 1))), x] + Dist[(2*c*d - b
```

```
*e)/(2*c), Int[(a + b*x + c*x^2)^p, x], x] /; FreeQ[{a, b, c, d, e, p}, x]
&& NeQ[2*c*d - b*e, 0] && NeQ[p, -1]
```

Rule 1117

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbol] :> With[{q = Rt[c
/a, 4]}, Simp[(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*x^2)^2)]/
(2*q*Sqrt[a + b*x^2 + c*x^4]))*EllipticF[2*ArcTan[q*x], 1/2 - b*(q^2/(4*c))
], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[c/a]
```

Rule 1209

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 4]}, Simp[(-d)*x*(Sqrt[a + b*x^2 + c*x^4]/(a*(1 + q
^2*x^2))), x] + Simp[d*(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*
x^2)^2)]/(q*Sqrt[a + b*x^2 + c*x^4]))*EllipticE[2*ArcTan[q*x], 1/2 - b*(q^2
/(4*c))], x] /; EqQ[e + d*q^2, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[c/a]
```

Rule 1211

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 2]}, Dist[(e + d*q)/q, Int[1/Sqrt[a + b*x^2 + c*x^4
], x], x] - Dist[e/q, Int[(1 - q*x^2)/Sqrt[a + b*x^2 + c*x^4], x], x] /; Ne
Q[e + d*q, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[
c/a]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2}{\sqrt{a+bx^2+cx^4}}\,dx + \int \frac{x(e+gx^2)}{\sqrt{a+bx^2+cx^4}}\,dx$$

$$=\frac{1}{2}\text{Subst}\left(\int\frac{e+gx}{\sqrt{a+bx+cx^2}}\,dx,x,x^2\right)-\frac{\left(\sqrt{a}f\right)\int\frac{1-\frac{\sqrt{c}x^2}{\sqrt{a}}}{\sqrt{a+bx^2+cx^4}}\,dx}{\sqrt{c}}$$
$$+\left(d+\frac{\sqrt{a}f}{\sqrt{c}}\right)\int\frac{1}{\sqrt{a+bx^2+cx^4}}\,dx$$

$$=\frac{g\sqrt{a+bx^2+cx^4}}{2c}+\frac{fx\sqrt{a+bx^2+cx^4}}{\sqrt{c}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$
$$-\frac{\sqrt[4]{a}f\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\left(\sqrt{c}d+\sqrt{a}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2\sqrt[4]{a}c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{(2ce-bg)\text{Subst}\left(\int\frac{1}{\sqrt{a+bx+cx^2}}\,dx,x,x^2\right)}{4c}$$

$$=\frac{g\sqrt{a+bx^2+cx^4}}{2c}+\frac{fx\sqrt{a+bx^2+cx^4}}{\sqrt{c}\left(\sqrt{a}+\sqrt{c}x^2\right)}$$
$$-\frac{\sqrt[4]{a}f\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\left(\sqrt{c}d+\sqrt{a}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2\sqrt[4]{a}c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{(2ce-bg)\text{Subst}\left(\int\frac{1}{4c-x^2}\,dx,x,\frac{b+2cx^2}{\sqrt{a+bx^2+cx^4}}\right)}{2c}$$

$$=\frac{g\sqrt{a+bx^2+cx^4}}{2c}+\frac{fx\sqrt{a+bx^2+cx^4}}{\sqrt{c}\left(\sqrt{a}+\sqrt{c}x^2\right)}+\frac{(2ce-bg)\tanh^{-1}\left(\frac{b+2cx^2}{2\sqrt{c}\sqrt{a+bx^2+cx^4}}\right)}{4c^{3/2}}$$
$$-\frac{\sqrt[4]{a}f\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{c^{3/4}\sqrt{a+bx^2+cx^4}}$$
$$+\frac{\left(\sqrt{c}d+\sqrt{a}f\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2\sqrt[4]{a}c^{3/4}\sqrt{a+bx^2+cx^4}}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 10.87 (sec) , antiderivative size = 525, normalized size of antiderivative = 1.46

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx$$
$$= \frac{i\sqrt{2}\sqrt{c}\left(-b+\sqrt{b^2-4ac}\right)f\sqrt{\frac{b-\sqrt{b^2-4ac}+2cx^2}{b-\sqrt{b^2-4ac}}}\sqrt{\frac{b+\sqrt{b^2-4ac}+2cx^2}{b+\sqrt{b^2-4ac}}}E\left(i\text{arcsinh}\left(\sqrt{2}\sqrt{\frac{c}{b+\sqrt{b^2-4ac}}}x\right)|\frac{b+\sqrt{b^2-4ac}}{b-\sqrt{b^2-4ac}}\right)\ldots}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/Sqrt[a + b*x^2 + c*x^4],x]
```

```
[Out] (I*Sqrt[2]*Sqrt[c]*(-b + Sqrt[b^2 - 4*a*c])*f*Sqrt[(b - Sqrt[b^2 - 4*a*c] +
 2*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(
b + Sqrt[b^2 - 4*a*c])]*EllipticE[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 -
4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - I*Sqrt[2]*S
qrt[c]*(2*c*d + (-b + Sqrt[b^2 - 4*a*c])*f)*Sqrt[(b - Sqrt[b^2 - 4*a*c] + 2
*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b
+ Sqrt[b^2 - 4*a*c])]*EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*
a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] + Sqrt[c/(b + S
qrt[b^2 - 4*a*c])]*(2*Sqrt[c]*g*(a + b*x^2 + c*x^4) + (-2*c*e + b*g)*Sqrt[a
 + b*x^2 + c*x^4]*Log[b + 2*c*x^2 - 2*Sqrt[c]*Sqrt[a + b*x^2 + c*x^4]]))/(4
*c^(3/2)*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[a + b*x^2 + c*x^4])
```

Maple [A] (verified)

Time = 4.08 (sec) , antiderivative size = 426, normalized size of antiderivative = 1.19

method	result
elliptic	$\frac{g\sqrt{c x^4+b x^2+a}}{2c}+\frac{d\sqrt{2}\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\sqrt{4+\frac{2\left(b+\sqrt{-4ac+b^2}\right)x^2}{a}}F\left(\frac{x\sqrt{2}\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}}{2},\frac{\sqrt{-4+\frac{2b\left(b+\sqrt{-4ac+b^2}\right)}{ac}}}{2}\right)}{4\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}\sqrt{c x^4+b x^2+a}}+\ldots$
risch	$\frac{g\sqrt{c x^4+b x^2+a}}{2c}-\frac{-\frac{cd\sqrt{2}\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\sqrt{4+\frac{2\left(b+\sqrt{-4ac+b^2}\right)x^2}{a}}F\left(\frac{x\sqrt{2}\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}}{2},\frac{\sqrt{-4+\frac{2b\left(b+\sqrt{-4ac+b^2}\right)}{ac}}}{2}\right)}{2\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}\sqrt{c x^4+b x^2+a}}+\ldots}{\ldots}$
default	$\frac{d\sqrt{2}\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\sqrt{4+\frac{2\left(b+\sqrt{-4ac+b^2}\right)x^2}{a}}F\left(\frac{x\sqrt{2}\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}}{2},\frac{\sqrt{-4+\frac{2b\left(b+\sqrt{-4ac+b^2}\right)}{ac}}}{2}\right)}{4\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}\sqrt{c x^4+b x^2+a}}+g\left(\frac{\sqrt{c x^4+b x^2+a}}{2c}\ldots\right.$

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(1/2),x,method=_RETURNVERBOSE)
```

```
[Out] 1/2*g*(c*x^4+b*x^2+a)^(1/2)/c+1/4*d*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/
2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*
x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)*EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^
(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))+1/2*(e-1/2*g
/c*b)*ln((2*c*x^2+b)/c^(1/2)+2*(c*x^4+b*x^2+a)^(1/2))/c^(1/2)-1/2*f*a*2^(1/
2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1
/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)/(b+(-4*a
*c+b^2)^(1/2))*(EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1
/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))-EllipticE(1/2*x*2^(1/2)*((-b+
(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))
)
```

Fricas [A] (verification not implemented)

none

Time = 0.19 (sec) , antiderivative size = 376, normalized size of antiderivative = 1.05

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx$$

$$= \frac{4\sqrt{\frac{1}{2}}\left(acfx\sqrt{\frac{b^2-4\,ac}{c^2}}-abfx\right)\sqrt{c}\sqrt{\frac{c\sqrt{\frac{b^2-4\,ac}{c^2}}-b}{c}}E(\arcsin\left(\frac{\sqrt{\frac{1}{2}}\sqrt{\frac{c\sqrt{\frac{b^2-4\,ac}{c^2}}-b}{c}}}{x}\right)\mid\frac{bc\sqrt{\frac{b^2-4\,ac}{c^2}}+b^2-2\,ac}{2\,ac})+4\sqrt{\frac{1}{2}}\left((c^2\right.}{}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(1/2),x, algorithm="fricas")
```

```
[Out] 1/8*(4*sqrt(1/2)*(a*c*f*x*sqrt((b^2 - 4*a*c)/c^2) - a*b*f*x)*sqrt(c)*sqrt((
c*sqrt((b^2 - 4*a*c)/c^2) - b)/c)*elliptic_e(arcsin(sqrt(1/2)*sqrt((c*sqrt(
(b^2 - 4*a*c)/c^2) - b)/c)/x), 1/2*(b*c*sqrt((b^2 - 4*a*c)/c^2) + b^2 - 2*a
*c)/(a*c)) + 4*sqrt(1/2)*((c^2*d - a*c*f)*x*sqrt((b^2 - 4*a*c)/c^2) + (b*c*
d + a*b*f)*x)*sqrt(c)*sqrt((c*sqrt((b^2 - 4*a*c)/c^2) - b)/c)*elliptic_f(ar
csin(sqrt(1/2)*sqrt((c*sqrt((b^2 - 4*a*c)/c^2) - b)/c)/x), 1/2*(b*c*sqrt((b
^2 - 4*a*c)/c^2) + b^2 - 2*a*c)/(a*c)) - (2*a*c*e - a*b*g)*sqrt(c)*x*log(8*
c^2*x^4 + 8*b*c*x^2 + b^2 - 4*sqrt(c*x^4 + b*x^2 + a)*(2*c*x^2 + b)*sqrt(c)
 + 4*a*c) + 4*sqrt(c*x^4 + b*x^2 + a)*(a*c*g*x + 2*a*c*f))/(a*c^2*x)
```

Sympy [F]

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx=\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**(1/2),x)
[Out] Integral((d + e*x + f*x**2 + g*x**3)/sqrt(a + b*x**2 + c*x**4), x)
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx=\int \frac{gx^3+fx^2+ex+d}{\sqrt{cx^4+bx^2+a}}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(1/2),x, algorithm="maxima")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/sqrt(c*x^4 + b*x^2 + a), x)
```

Giac [F]

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx=\int \frac{gx^3+fx^2+ex+d}{\sqrt{cx^4+bx^2+a}}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(1/2),x, algorithm="giac")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/sqrt(c*x^4 + b*x^2 + a), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{\sqrt{a+bx^2+cx^4}}\,dx=\int \frac{g\,x^3+f\,x^2+e\,x+d}{\sqrt{c\,x^4+b\,x^2+a}}\,dx$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(1/2),x)
[Out] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(1/2), x)
```

3.106 $\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 447

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{a(b^2-4ac)\sqrt{a+bx^2+cx^4}} - \frac{be-2ag+(2ce-bg)x^2}{(b^2-4ac)\sqrt{a+bx^2+cx^4}} - \frac{\sqrt{c}(bd-2af)x\sqrt{a+bx^2+cx^4}}{a(b^2-4ac)(\sqrt{a}+\sqrt{c}x^2)} + \frac{\sqrt[4]{c}(bd-2af)(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{a^{3/4}(b^2-4ac)\sqrt{a+bx^2+cx^4}} - \frac{(\sqrt{c}d-\sqrt{a}f)(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}\operatorname{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2a^{3/4}(b-2\sqrt{a}\sqrt{c})\sqrt[4]{c}\sqrt{a+bx^2+cx^4}}$$

```
[Out] x*(b^2*d-2*a*c*d-a*b*f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(
1/2)+(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(1/2)-(-2*a
*f+b*d)*x*c^(1/2)*(c*x^4+b*x^2+a)^(1/2)/a/(-4*a*c+b^2)/(a^(1/2)+x^2*c^(1/2)
)+c^(1/4)*(-2*a*f+b*d)*(cos(2*arctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arc
tan(c^(1/4)*x/a^(1/4)))*EllipticE(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b
/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2)+x^2*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^
2*c^(1/2))^2)^(1/2)/a^(3/4)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(1/2)-1/2*(cos(2*a
rctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)*x/a^(1/4)))*Ellipti
cF(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^(1/2))^(1/2))*(-f*a^
(1/2)+d*c^(1/2))*(a^(1/2)+x^2*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2
))^2)^(1/2)/a^(3/4)/c^(1/4)/(b-2*a^(1/2)*c^(1/2))/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 447, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.219$, Rules used = {1687, 1192, 1211, 1117, 1209, 1261, 650}

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{\sqrt[4]{c}(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}(bd-2af)E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{a^{3/4}(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$
$$-\frac{(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}(\sqrt{c}d-\sqrt{a}f)\operatorname{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2a^{3/4}\sqrt[4]{c}(b-2\sqrt{a}\sqrt{c})\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt{c}x\sqrt{a+bx^2+cx^4}(bd-2af)}{a(b^2-4ac)(\sqrt{a}+\sqrt{c}x^2)}$$
$$+\frac{x(cx^2(bd-2af)-abf-2acd+b^2d)}{a(b^2-4ac)\sqrt{a+bx^2+cx^4}}-\frac{-2ag+x^2(2ce-bg)+be}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (x*(b^2*d - 2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(a*(b^2 - 4*a*c)*Sqrt[a
 + b*x^2 + c*x^4]) - (b*e - 2*a*g + (2*c*e - b*g)*x^2)/((b^2 - 4*a*c)*Sqrt[
a + b*x^2 + c*x^4]) - (Sqrt[c]*(b*d - 2*a*f)*x*Sqrt[a + b*x^2 + c*x^4])/(a*
(b^2 - 4*a*c)*(Sqrt[a] + Sqrt[c]*x^2)) + (c^(1/4)*(b*d - 2*a*f)*(Sqrt[a] +
Sqrt[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticE[
2*ArcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(a^(3/4)*(b^2
- 4*a*c)*Sqrt[a + b*x^2 + c*x^4]) - ((Sqrt[c]*d - Sqrt[a]*f)*(Sqrt[a] + Sqr
t[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^2]*EllipticF[2*A
rcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(2*a^(3/4)*(b - 2
*Sqrt[a]*Sqrt[c])*c^(1/4)*Sqrt[a + b*x^2 + c*x^4])
```

Rule 650

```
Int[((d_.) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(3/2), x_Symbo
l] :> Simp[-2*((b*d - 2*a*e + (2*c*d - b*e)*x)/((b^2 - 4*a*c)*Sqrt[a + b*x
+ c*x^2])), x] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b
^2 - 4*a*c, 0]
```

Rule 1117

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbol] :> With[{q = Rt[c
/a, 4]}, Simp[(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*x^2)^2)]/
(2*q*Sqrt[a + b*x^2 + c*x^4]))*EllipticF[2*ArcTan[q*x], 1/2 - b*(q^2/(4*c))
], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[c/a]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1209

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 4]}, Simp[(-d)*x*(Sqrt[a + b*x^2 + c*x^4]/(a*(1 + q
^2*x^2))), x] + Simp[d*(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*
x^2)^2)]/(q*Sqrt[a + b*x^2 + c*x^4]))*EllipticE[2*ArcTan[q*x], 1/2 - b*(q^2
/(4*c))], x] /; EqQ[e + d*q^2, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[c/a]
```

Rule 1211

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 2]}, Dist[(e + d*q)/q, Int[1/Sqrt[a + b*x^2 + c*x^4
], x], x] - Dist[e/q, Int[(1 - q*x^2)/Sqrt[a + b*x^2 + c*x^4], x], x] /; Ne
Q[e + d*q, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[
c/a]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2}{(a+bx^2+cx^4)^{3/2}}\,dx + \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^{3/2}}\,dx$$

$$
\begin{aligned}
&= \frac{x(b^2d - 2acd - abf + c(bd - 2af)x^2)}{a\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} \\
&\quad + \frac{1}{2}\text{Subst}\left(\int \frac{e + gx}{(a + bx + cx^2)^{3/2}}\,dx, x, x^2\right) - \frac{\int \frac{a(2cd - bf) + c(bd - 2af)x^2}{\sqrt{a + bx^2 + cx^4}}\,dx}{a\left(b^2 - 4ac\right)} \\
&= \frac{x(b^2d - 2acd - abf + c(bd - 2af)x^2)}{a\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} \\
&\quad - \frac{be - 2ag + (2ce - bg)x^2}{\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} + \frac{\left(\sqrt{c}(bd - 2af)\right)\int \frac{1 - \frac{\sqrt{c}x^2}{\sqrt{a}}}{\sqrt{a + bx^2 + cx^4}}\,dx}{\sqrt{a}\left(b^2 - 4ac\right)} \\
&\quad - \frac{\left(\sqrt{c}(bd - 2af) + \sqrt{a}(2cd - bf)\right)\int \frac{1}{\sqrt{a + bx^2 + cx^4}}\,dx}{\sqrt{a}\left(b^2 - 4ac\right)} \\
&= \frac{x(b^2d - 2acd - abf + c(bd - 2af)x^2)}{a\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} \\
&\quad - \frac{be - 2ag + (2ce - bg)x^2}{\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} - \frac{\sqrt{c}(bd - 2af)x\sqrt{a + bx^2 + cx^4}}{a\left(b^2 - 4ac\right)\left(\sqrt{a} + \sqrt{c}x^2\right)} \\
&\quad + \frac{\sqrt[4]{c}(bd - 2af)\left(\sqrt{a} + \sqrt{c}x^2\right)\sqrt{\frac{a + bx^2 + cx^4}{\left(\sqrt{a} + \sqrt{c}x^2\right)^2}}\,E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right) | \frac{1}{4}\left(2 - \frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{a^{3/4}\left(b^2 - 4ac\right)\sqrt{a + bx^2 + cx^4}} \\
&\quad - \frac{\left(\sqrt{c}d - \sqrt{a}f\right)\left(\sqrt{a} + \sqrt{c}x^2\right)\sqrt{\frac{a + bx^2 + cx^4}{\left(\sqrt{a} + \sqrt{c}x^2\right)^2}}\,F\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right) | \frac{1}{4}\left(2 - \frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{2a^{3/4}\left(b - 2\sqrt{a}\sqrt{c}\right)\sqrt[4]{c}\sqrt{a + bx^2 + cx^4}}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 11.16 (sec) , antiderivative size = 513, normalized size of antiderivative = 1.15

$$
\int \frac{d + ex + fx^2 + gx^3}{(a + bx^2 + cx^4)^{3/2}}\,dx =
$$

$$
-\frac{4\sqrt{\frac{c}{b + \sqrt{b^2 - 4ac}}}(-2a^2g - bdx(b + cx^2) + 2acx(d + x(e + fx)) + ab(e + x(f - gx))) + i\left(-b + \sqrt{b^2 - 4ac}\right)(b
$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] -1/4*(4*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*(-2*a^2*g - b*d*x*(b + c*x^2) + 2*a
*c*x*(d + x*(e + f*x)) + a*b*(e + x*(f - g*x))) + I*(-b + Sqrt[b^2 - 4*a*c]
)*(b*d - 2*a*f)*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*
c])]*Sqrt[(2*b - 2*Sqrt[b^2 - 4*a*c] + 4*c*x^2)/(b - Sqrt[b^2 - 4*a*c])]*El
```

```
lipticE[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2
 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])] - I*(-(b^2*d) + 4*a*c*d + b*Sqrt[b^2 -
4*a*c]*d - 2*a*Sqrt[b^2 - 4*a*c]*f)*Sqrt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/
(b + Sqrt[b^2 - 4*a*c])]*Sqrt[(2*b - 2*Sqrt[b^2 - 4*a*c] + 4*c*x^2)/(b - Sq
rt[b^2 - 4*a*c])]*EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c]
)]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*S
qrt[c/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[a + b*x^2 + c*x^4])
```

Maple [A] (verified)

Time = 2.11 (sec) , antiderivative size = 565, normalized size of antiderivative = 1.26

method	result
elliptic	$-\frac{2c\left(-\frac{(2af-bd)x^3}{2a\left(4ac-b^2\right)}+\frac{(bg-2ec)x^2}{2c\left(4ac-b^2\right)}-\frac{\left(abf+2acd-b^2d\right)x}{2ac\left(4ac-b^2\right)}+\frac{2ag-be}{2\left(4ac-b^2\right)c}\right)}{\sqrt{\left(x^4+\frac{b\,x^2}{c}+\frac{a}{c}\right)c}}+\frac{\left(\frac{d}{a}-\frac{abf+2acd-b^2d}{a\left(4ac-b^2\right)}\right)\sqrt{2}\,\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\,\sqrt{4+\frac{2(b}{}}}{4\sqrt{\frac{-b+\sqrt{-4a}}{a}}}$
default	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] -2*c*(-1/2*(2*a*f-b*d)/a/(4*a*c-b^2)*x^3+1/2*(b*g-2*c*e)/c/(4*a*c-b^2)*x^2-
1/2*(a*b*f+2*a*c*d-b^2*d)/a/c/(4*a*c-b^2)*x+1/2*(2*a*g-b*e)/(4*a*c-b^2)/c)/
((x^4+1/c*b*x^2+a/c)*c)^(1/2)+1/4*(d/a-(a*b*f+2*a*c*d-b^2*d)/a/(4*a*c-b^2))
*2^(1/2)/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x
^2)^(1/2)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)*El
lipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*
a*c+b^2)^(1/2))/a/c)^(1/2))+1/2*c*(2*a*f-b*d)/(4*a*c-b^2)*2^(1/2)/((-b+(-4*
a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+
(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)/(b+(-4*a*c+b^2)^(1/2
))*(EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(
b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))-EllipticE(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)
^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2)))
```

Fricas [A] (verification not implemented)

none

Time = 0.11 (sec) , antiderivative size = 723, normalized size of antiderivative = 1.62

$$\int \frac{d+ex+fx^2+gx^3}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx =$$

$$-\frac{\sqrt{\frac{1}{2}}\left(ab^2cd-2\,a^2bcf+(b^2c^2d-2\,abc^2f)x^4+(b^3cd-2\,ab^2cf)x^2-(a^2bcd-2\,a^3cf+(abc^2d-2\,a^2c^2f)x^4\right.}{}$$

[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="fricas")

[Out] -1/2*(sqrt(1/2)*(a*b^2*c*d - 2*a^2*b*c*f + (b^2*c^2*d - 2*a*b*c^2*f)*x^4 + (b^3*c*d - 2*a*b^2*c*f)*x^2 - (a^2*b*c*d - 2*a^3*c*f + (a*b*c^2*d - 2*a^2*c^2*f)*x^4 + (a*b^2*c*d - 2*a^2*b*c*f)*x^2)*sqrt((b^2 - 4*a*c)/a^2))*sqrt(a)*sqrt((a*sqrt((b^2 - 4*a*c)/a^2) - b)/a)*elliptic_e(arcsin(sqrt(1/2)*x*sqrt((a*sqrt((b^2 - 4*a*c)/a^2) - b)/a)), 1/2*(a*b*sqrt((b^2 - 4*a*c)/a^2) + b^2 - 2*a*c)/(a*c)) - sqrt(1/2)*(((2*a*b + b^2)*c^2*d - (a*b^2*c + 2*a*b*c^2)*f)*x^4 + (2*a^2*b + a*b^2)*c*d + ((2*a*b^2 + b^3)*c*d - (a*b^3 + 2*a*b^2*c)*f)*x^2 - (a^2*b^2 + 2*a^2*b*c)*f + (((2*a^2 - a*b)*c^2*d - (a^2*b*c - 2*a^2*c^2)*f)*x^4 + (2*a^3 - a^2*b)*c*d + ((2*a^2*b - a*b^2)*c*d - (a^2*b^2 - 2*a^2*b*c)*f)*x^2 - (a^3*b - 2*a^3*c)*f)*sqrt((b^2 - 4*a*c)/a^2))*sqrt(a)*sqrt((a*sqrt((b^2 - 4*a*c)/a^2) - b)/a)*elliptic_f(arcsin(sqrt(1/2)*x*sqrt((a*sqrt((b^2 - 4*a*c)/a^2) - b)/a)), 1/2*(a*b*sqrt((b^2 - 4*a*c)/a^2) + b^2 - 2*a*c)/(a*c)) + 2*(a^2*b*c*e - 2*a^3*c*g - (a*b*c^2*d - 2*a^2*c^2*f)*x^3 + (2*a^2*c^2*e - a^2*b*c*g)*x^2 + (a^2*b*c*f - (a*b^2*c - 2*a^2*c^2)*d)*x)*sqrt(c*x^4 + b*x^2 + a))/(a^3*b^2*c - 4*a^4*c^2 + (a^2*b^2*c^2 - 4*a^3*c^3)*x^4 + (a^2*b^3*c - 4*a^3*b*c^2)*x^2)

Sympy [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{\frac{3}{2}}}\,dx$$

[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**(3/2),x)

[Out] Integral((d + e*x + f*x**2 + g*x**3)/(a + b*x**2 + c*x**4)**(3/2), x)

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^{\frac{3}{2}}}\,dx$$

[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxima")

[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a)^(3/2), x)

Giac [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a)^(3/2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{3/2}}\,dx = \int \frac{g\,x^3+f\,x^2+e\,x+d}{(c\,x^4+b\,x^2+a)^{3/2}}\,dx$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(3/2),x)
[Out] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(3/2), x)
```

3.107 $\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 680

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx = \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{3a(b^2-4ac)(a+bx^2+cx^4)^{3/2}}$$
$$-\frac{be-2ag+(2ce-bg)x^2}{3(b^2-4ac)(a+bx^2+cx^4)^{3/2}}+\frac{4(2ce-bg)(b+2cx^2)}{3(b^2-4ac)^2\sqrt{a+bx^2+cx^4}}$$
$$+\frac{x(2b^4d-17ab^2cd+20a^2c^2d+ab^3f+4a^2bcf+c(2b^3d-16abcd+ab^2f+12a^2cf)x^2)}{3a^2(b^2-4ac)^2\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt{c}(2b^3d-16abcd+ab^2f+12a^2cf)x\sqrt{a+bx^2+cx^4}}{3a^2(b^2-4ac)^2(\sqrt{a}+\sqrt{c}x^2)}$$
$$+\frac{\sqrt[4]{c}(2b^3d-16abcd+ab^2f+12a^2cf)(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{3a^{7/4}(b^2-4ac)^2\sqrt{a+bx^2+cx^4}}$$
$$-\frac{\sqrt[4]{c}(2b^2d-3\sqrt{a}b\sqrt{c}d-10acd+abf+6a^{3/2}\sqrt{c}f)(\sqrt{a}+\sqrt{c}x^2)\sqrt{\frac{a+bx^2+cx^4}{(\sqrt{a}+\sqrt{c}x^2)^2}}\,\text{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right.\right.}{6a^{7/4}(b-2\sqrt{a}\sqrt{c})(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[Out] 1/3*x*(b^2*d-2*a*c*d-a*b*f+c*(-2*a*f+b*d)*x^2)/a/(-4*a*c+b^2)/(c*x^4+b*x^2+
a)^(3/2)+1/3*(-b*e+2*a*g-(-b*g+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(3/
2)+4/3*(-b*g+2*c*e)*(2*c*x^2+b)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)^(1/2)+1/3*x*
(2*b^4*d-17*a*b^2*c*d+20*a^2*c^2*d+a*b^3*f+4*a^2*b*c*f+c*(12*a^2*c*f+a*b^2*
f-16*a*b*c*d+2*b^3*d)*x^2)/a^2/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)^(1/2)-1/3*(12
*a^2*c*f+a*b^2*f-16*a*b*c*d+2*b^3*d)*x*c^(1/2)*(c*x^4+b*x^2+a)^(1/2)/a^2/(-
4*a*c+b^2)^2/(a^(1/2)+x^2*c^(1/2))+1/3*c^(1/4)*(12*a^2*c*f+a*b^2*f-16*a*b*c
*d+2*b^3*d)*(cos(2*arctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)
*x/a^(1/4)))*EllipticE(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^
```

```
(1/2))^(1/2))*(a^(1/2)+x^2*c^(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2))^
2)^(1/2)/a^(7/4)/(-4*a*c+b^2)^2/(c*x^4+b*x^2+a)^(1/2)-1/6*c^(1/4)*(cos(2*ar
ctan(c^(1/4)*x/a^(1/4)))^2)^(1/2)/cos(2*arctan(c^(1/4)*x/a^(1/4)))*Elliptic
F(sin(2*arctan(c^(1/4)*x/a^(1/4))),1/2*(2-b/a^(1/2)/c^(1/2))^(1/2))*(a^(1/2
)+x^2*c^(1/2))*(2*b^2*d-10*a*c*d+a*b*f+6*a^(3/2)*f*c^(1/2)-3*b*d*a^(1/2)*c^
(1/2))*((c*x^4+b*x^2+a)/(a^(1/2)+x^2*c^(1/2))^2)^(1/2)/a^(7/4)/(-4*a*c+b^2)
/(b-2*a^(1/2)*c^(1/2))/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.32 (sec) , antiderivative size = 680, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.250$, Rules used = {1687, 1192, 1211, 1117, 1209, 1261, 652, 627}

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx =$$

$$-\frac{\sqrt[4]{c}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\left(6a^{3/2}\sqrt{c}f-3\sqrt{a}b\sqrt{c}d+abf-10acd+2b^2d\right)\text{EllipticF}\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right),\right.}{6a^{7/4}\left(b-2\sqrt{a}\sqrt{c}\right)\left(b^2-4ac\right)\sqrt{a+bx^2+cx^4}}$$

$$-\frac{\sqrt{c}x\sqrt{a+bx^2+cx^4}\left(12a^2cf+ab^2f-16abcd+2b^3d\right)}{3a^2\left(b^2-4ac\right)^2\left(\sqrt{a}+\sqrt{c}x^2\right)}$$

$$+\frac{x\left(cx^2\left(12a^2cf+ab^2f-16abcd+2b^3d\right)+4a^2bcf+20a^2c^2d+ab^3f-17ab^2cd+2b^4d\right)}{3a^2\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}}$$

$$+\frac{\sqrt[4]{c}\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}\left(12a^2cf+ab^2f-16abcd+2b^3d\right)E\left(2\arctan\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{c}}\right)\right)}{3a^{7/4}\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}}$$

$$+\frac{x\left(cx^2(bd-2af)-abf-2acd+b^2d\right)}{3a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}}$$

$$+\frac{4\left(b+2cx^2\right)(2ce-bg)}{3\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}}-\frac{-2ag+x^2(2ce-bg)+be}{3\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}}$$

```
[In] Int[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(5/2),x]
```

```
[Out] (x*(b^2*d - 2*a*c*d - a*b*f + c*(b*d - 2*a*f)*x^2))/(3*a*(b^2 - 4*a*c)*(a +
 b*x^2 + c*x^4)^(3/2)) - (b*e - 2*a*g + (2*c*e - b*g)*x^2)/(3*(b^2 - 4*a*c)
*(a + b*x^2 + c*x^4)^(3/2)) + (4*(2*c*e - b*g)*(b + 2*c*x^2))/(3*(b^2 - 4*a
*c)^2*Sqrt[a + b*x^2 + c*x^4]) + (x*(2*b^4*d - 17*a*b^2*c*d + 20*a^2*c^2*d
+ a*b^3*f + 4*a^2*b*c*f + c*(2*b^3*d - 16*a*b*c*d + a*b^2*f + 12*a^2*c*f)*x
^2))/(3*a^2*(b^2 - 4*a*c)^2*Sqrt[a + b*x^2 + c*x^4]) - (Sqrt[c]*(2*b^3*d -
16*a*b*c*d + a*b^2*f + 12*a^2*c*f)*x*Sqrt[a + b*x^2 + c*x^4])/(3*a^2*(b^2 -
 4*a*c)^2*(Sqrt[a] + Sqrt[c]*x^2)) + (c^(1/4)*(2*b^3*d - 16*a*b*c*d + a*b^2
*f + 12*a^2*c*f)*(Sqrt[a] + Sqrt[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a]
```

```
+ Sqrt[c]*x^2)^2]*EllipticE[2*ArcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*
Sqrt[c]))/4])/(3*a^(7/4)*(b^2 - 4*a*c)^2*Sqrt[a + b*x^2 + c*x^4]) - (c^(1/4
)*(2*b^2*d - 3*Sqrt[a]*b*Sqrt[c]*d - 10*a*c*d + a*b*f + 6*a^(3/2)*Sqrt[c]*f
)*(Sqrt[a] + Sqrt[c]*x^2)*Sqrt[(a + b*x^2 + c*x^4)/(Sqrt[a] + Sqrt[c]*x^2)^
2]*EllipticF[2*ArcTan[(c^(1/4)*x)/a^(1/4)], (2 - b/(Sqrt[a]*Sqrt[c]))/4])/(
6*a^(7/4)*(b - 2*Sqrt[a]*Sqrt[c])*(b^2 - 4*a*c)*Sqrt[a + b*x^2 + c*x^4])
```

Rule 627

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-3/2), x_Symbol] :> Simp[-2*((b +
2*c*x)/((b^2 - 4*a*c)*Sqrt[a + b*x + c*x^2])), x] /; FreeQ[{a, b, c}, x] &&
 NeQ[b^2 - 4*a*c, 0]
```

Rule 652

```
Int[((d_.) + (e_.)*(x_))*((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(p_), x_Symbol
] :> Simp[((b*d - 2*a*e + (2*c*d - b*e)*x)/((p + 1)*(b^2 - 4*a*c)))*(a + b*
x + c*x^2)^(p + 1), x] - Dist[(2*p + 3)*((2*c*d - b*e)/((p + 1)*(b^2 - 4*a*
c))), Int[(a + b*x + c*x^2)^(p + 1), x], x] /; FreeQ[{a, b, c, d, e}, x] &&
 NeQ[2*c*d - b*e, 0] && NeQ[b^2 - 4*a*c, 0] && LtQ[p, -1] && NeQ[p, -3/2]
```

Rule 1117

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbol] :> With[{q = Rt[c
/a, 4]}, Simp[(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*x^2)^2)]/
(2*q*Sqrt[a + b*x^2 + c*x^4]))*EllipticF[2*ArcTan[q*x], 1/2 - b*(q^2/(4*c))
], x]] /; FreeQ[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[c/a]
```

Rule 1192

```
Int[((d_) + (e_.)*(x_)^2)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symb
ol] :> Simp[x*(a*b*e - d*(b^2 - 2*a*c) - c*(b*d - 2*a*e)*x^2)*((a + b*x^2 +
 c*x^4)^(p + 1)/(2*a*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(2*a*(p + 1)*(b^2
 - 4*a*c)), Int[Simp[(2*p + 3)*d*b^2 - a*b*e - 2*a*c*d*(4*p + 5) + (4*p + 7
)*(d*b - 2*a*e)*c*x^2, x]*(a + b*x^2 + c*x^4)^(p + 1), x], x] /; FreeQ[{a,
b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] &&
LtQ[p, -1] && IntegerQ[2*p]
```

Rule 1209

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 4]}, Simp[(-d)*x*(Sqrt[a + b*x^2 + c*x^4]/(a*(1 + q
^2*x^2))), x] + Simp[d*(1 + q^2*x^2)*(Sqrt[(a + b*x^2 + c*x^4)/(a*(1 + q^2*
x^2)^2)]/(q*Sqrt[a + b*x^2 + c*x^4]))*EllipticE[2*ArcTan[q*x], 1/2 - b*(q^2
/(4*c))], x] /; EqQ[e + d*q^2, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2
- 4*a*c, 0] && PosQ[c/a]
```

Rule 1211

```
Int[((d_) + (e_.)*(x_)^2)/Sqrt[(a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4], x_Symbo
l] :> With[{q = Rt[c/a, 2]}, Dist[(e + d*q)/q, Int[1/Sqrt[a + b*x^2 + c*x^4
], x], x] - Dist[e/q, Int[(1 - q*x^2)/Sqrt[a + b*x^2 + c*x^4], x], x] /; Ne
Q[e + d*q, 0]] /; FreeQ[{a, b, c, d, e}, x] && NeQ[b^2 - 4*a*c, 0] && PosQ[
c/a]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\text{integral} = \int \frac{d+fx^2}{(a+bx^2+cx^4)^{5/2}}\,dx + \int \frac{x(e+gx^2)}{(a+bx^2+cx^4)^{5/2}}\,dx$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{3a(b^2-4ac)(a+bx^2+cx^4)^{3/2}} + \frac{1}{2}\text{Subst}\left(\int \frac{e+gx}{(a+bx+cx^2)^{5/2}}\,dx, x, x^2\right) - \frac{\int \frac{-2b^2d+10acd-abf-3c(bd-2af)x^2}{(a+bx^2+cx^4)^{3/2}}\,dx}{3a(b^2-4ac)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{3a(b^2-4ac)(a+bx^2+cx^4)^{3/2}} - \frac{be-2ag+(2ce-bg)x^2}{3(b^2-4ac)(a+bx^2+cx^4)^{3/2}} + \frac{x(2b^4d-17ab^2cd+20a^2c^2d+ab^3f+4a^2bcf+c(2b^3d-16abcd+ab^2f+12a^2cf)x^2)}{3a^2(b^2-4ac)^2\sqrt{a+bx^2+cx^4}} + \frac{\int \frac{-ac(b^2d-20acd+8abf)-c(2b^3d-16abcd+ab^2f+12a^2cf)x^2}{\sqrt{a+bx^2+cx^4}}\,dx}{3a^2(b^2-4ac)^2} - \frac{(2(2ce-bg))\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^{3/2}}\,dx, x, x^2\right)}{3(b^2-4ac)}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{3a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}} - \frac{be-2ag+(2ce-bg)x^2}{3\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}} + \frac{4(2ce-bg)\left(b+2cx^2\right)}{3\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}} + \frac{x(2b^4d-17ab^2cd+20a^2c^2d+ab^3f+4a^2bcf+c(2b^3d-16abcd+ab^2f+12a^2cf)\,x^2)}{3a^2\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}} + \frac{\left(\sqrt{c}(2b^3d-16abcd+ab^2f+12a^2cf)\right)\int\frac{1-\frac{\sqrt{c}x^2}{\sqrt{a}}}{\sqrt{a+bx^2+cx^4}}dx}{3a^{3/2}\left(b^2-4ac\right)^2} - \frac{\left(\sqrt{c}\left(2b^3d-16abcd+ab^2f+12a^2cf+\sqrt{a}\sqrt{c}(b^2d-20acd+8abf)\right)\right)\int\frac{1}{\sqrt{a+bx^2+cx^4}}dx}{3a^{3/2}\left(b^2-4ac\right)^2}$$

$$= \frac{x(b^2d-2acd-abf+c(bd-2af)x^2)}{3a\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}} - \frac{be-2ag+(2ce-bg)x^2}{3\left(b^2-4ac\right)\left(a+bx^2+cx^4\right)^{3/2}} + \frac{4(2ce-bg)\left(b+2cx^2\right)}{3\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}} + \frac{x(2b^4d-17ab^2cd+20a^2c^2d+ab^3f+4a^2bcf+c(2b^3d-16abcd+ab^2f+12a^2cf)\,x^2)}{3a^2\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}} - \frac{\sqrt{c}(2b^3d-16abcd+ab^2f+12a^2cf)\,x\sqrt{a+bx^2+cx^4}}{3a^2\left(b^2-4ac\right)^2\left(\sqrt{a}+\sqrt{c}x^2\right)} + \frac{\sqrt[4]{c}(2b^3d-16abcd+ab^2f+12a^2cf)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}E\left(2\tan^{-1}\left(\frac{\sqrt[4]{c}x}{\sqrt[4]{a}}\right)|\frac{1}{4}\left(2-\frac{b}{\sqrt{a}\sqrt{}}\right.\right.}{3a^{7/4}\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}} - \frac{\sqrt[4]{c}\left(2b^3d-16abcd+ab^2f+12a^2cf+\sqrt{a}\sqrt{c}(b^2d-20acd+8abf)\right)\left(\sqrt{a}+\sqrt{c}x^2\right)\sqrt{\frac{a+bx^2+cx^4}{\left(\sqrt{a}+\sqrt{c}x^2\right)^2}}}{6a^{7/4}\left(b^2-4ac\right)^2\sqrt{a+bx^2+cx^4}}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 11.99 (sec) , antiderivative size = 598, normalized size of antiderivative = 0.88

$$\int \frac{d+ex+fx^2+gx^3}{\left(a+bx^2+cx^4\right)^{5/2}}\,dx = \frac{-4a(b^2-4ac)\left(-2a^2g-bdx(b+cx^2)+2acx(d+x(e+fx))+ab(e+x(f-g\right.}{}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(5/2),x]
```

```
[Out] (-4*a*(b^2 - 4*a*c)*(-2*a^2*g - b*d*x*(b + c*x^2) + 2*a*c*x*(d + x*(e + f*x
)) + a*b*(e + x*(f - g*x))) + 4*(a + b*x^2 + c*x^4)*(2*b^3*d*x*(b + c*x^2)
+ a*b*x*(-17*b*c*d + b^2*f - 16*c^2*d*x^2 + b*c*f*x^2) + 4*a^2*(-(b^2*g) +
```

```
c^2*x*(5*d + x*(4*e + 3*f*x)) + b*c*(2*e + x*(f - 2*g*x)))) + (I*Sqrt[2]*Sq
rt[(b + Sqrt[b^2 - 4*a*c] + 2*c*x^2)/(b + Sqrt[b^2 - 4*a*c])]*Sqrt[1 + (2*c
*x^2)/(b - Sqrt[b^2 - 4*a*c])]*(a + b*x^2 + c*x^4)*(-((-b + Sqrt[b^2 - 4*a*
c])*(2*b^3*d - 16*a*b*c*d + a*b^2*f + 12*a^2*c*f)*EllipticE[I*ArcSinh[Sqrt[
2]*Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^
2 - 4*a*c])]) + (-2*b^4*d + b^3*(2*Sqrt[b^2 - 4*a*c]*d - a*f) + 4*a*b*c*(-4
*Sqrt[b^2 - 4*a*c]*d + a*f) + a*b^2*(18*c*d + Sqrt[b^2 - 4*a*c]*f) + 4*a^2*
c*(-10*c*d + 3*Sqrt[b^2 - 4*a*c]*f))*EllipticF[I*ArcSinh[Sqrt[2]*Sqrt[c/(b
+ Sqrt[b^2 - 4*a*c])]*x], (b + Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c])])
)/Sqrt[c/(b + Sqrt[b^2 - 4*a*c])]])/(12*a^2*(b^2 - 4*a*c)^2*(a + b*x^2 + c*x
^4)^(3/2))
```

Maple [A] (verified)

Time = 2.32 (sec) , antiderivative size = 829, normalized size of antiderivative = 1.22

method	result
elliptic	$\frac{\left(\frac{(2af-bd)x^3}{3ca\left(4ac-b^2\right)}-\frac{(bg-2ec)x^2}{3\left(4ac-b^2\right)c^2}+\frac{\left(abf+2acd-b^2d\right)x}{3a\left(4ac-b^2\right)c^2}-\frac{2ag-be}{3\left(4ac-b^2\right)c^2}\right)\sqrt{cx^4+bx^2+a}}{\left(x^4+\frac{bx^2}{c}+\frac{a}{c}\right)^2}-\frac{2c\left(-\frac{\left(12a^2cf+ab^2f-16abcd+2b^3d\right)x^3}{6a^2\left(4ac-b^2\right)^2}+\frac{4(bg-2ec)x}{3\left(4ac-b^2\right)}\right.}{\sqrt{(}}$
default	Expression too large to display

```
[In] int((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(5/2),x,method=_RETURNVERBOSE)
```

```
[Out] (1/3/c*(2*a*f-b*d)/a/(4*a*c-b^2)*x^3-1/3*(b*g-2*c*e)/(4*a*c-b^2)/c^2*x^2+1/
3*(a*b*f+2*a*c*d-b^2*d)/a/(4*a*c-b^2)/c^2*x-1/3*(2*a*g-b*e)/(4*a*c-b^2)/c^2
)*(c*x^4+b*x^2+a)^(1/2)/(x^4+1/c*b*x^2+a/c)^2-2*c*(-1/6*(12*a^2*c*f+a*b^2*f
-16*a*b*c*d+2*b^3*d)/a^2/(4*a*c-b^2)^2*x^3+4/3*(b*g-2*c*e)/(4*a*c-b^2)^2*x^
2-1/6*(4*a^2*b*c*f+20*a^2*c^2*d+a*b^3*f-17*a*b^2*c*d+2*b^4*d)/a^2/(4*a*c-b^
2)^2/c*x+2/3*b*(b*g-2*c*e)/(4*a*c-b^2)^2/c)/((x^4+1/c*b*x^2+a/c)*c)^(1/2)+1
/4*(-1/3/(4*a*c-b^2)*(a*b*f-10*a*c*d+2*b^2*d)/a^2-1/3*(4*a^2*b*c*f+20*a^2*c
^2*d+a*b^3*f-17*a*b^2*c*d+2*b^4*d)/a^2/(4*a*c-b^2)^2)*2^(1/2)/((-b+(-4*a*c+
b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)*(4+2*(b+(-4*
a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)*EllipticF(1/2*x*2^(1/2)*
((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(
1/2))+1/6*c*(12*a^2*c*f+a*b^2*f-16*a*b*c*d+2*b^3*d)/(4*a*c-b^2)^2/a*2^(1/2)
/((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2)*(4-2*(-b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2
)*(4+2*(b+(-4*a*c+b^2)^(1/2))/a*x^2)^(1/2)/(c*x^4+b*x^2+a)^(1/2)/(b+(-4*a*c
+b^2)^(1/2))*(EllipticF(1/2*x*2^(1/2)*((-b+(-4*a*c+b^2)^(1/2))/a)^(1/2),1/2
*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2))-EllipticE(1/2*x*2^(1/2)*((-b+(-
4*a*c+b^2)^(1/2))/a)^(1/2),1/2*(-4+2*b*(b+(-4*a*c+b^2)^(1/2))/a/c)^(1/2)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1948 vs. $2(652) = 1304$.

Time = 0.14 (sec) , antiderivative size = 1948, normalized size of antiderivative = 2.86

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(5/2),x, algorithm="fricas")
```

```
[Out] -1/6*(sqrt(1/2)*((2*(b^4*c^2 - 8*a*b^2*c^3)*d + (a*b^3*c^2 + 12*a^2*b*c^3)*
f)*x^8 + 2*(2*(b^5*c - 8*a*b^3*c^2)*d + (a*b^4*c + 12*a^2*b^2*c^2)*f)*x^6 +
 (2*(b^6 - 6*a*b^4*c - 16*a^2*b^2*c^2)*d + (a*b^5 + 14*a^2*b^3*c + 24*a^3*b
*c^2)*f)*x^4 + 2*(2*(a*b^5 - 8*a^2*b^3*c)*d + (a^2*b^4 + 12*a^3*b^2*c)*f)*x
^2 + 2*(a^2*b^4 - 8*a^3*b^2*c)*d + (a^3*b^3 + 12*a^4*b*c)*f - ((2*(a*b^3*c^
2 - 8*a^2*b*c^3)*d + (a^2*b^2*c^2 + 12*a^3*c^3)*f)*x^8 + 2*(2*(a*b^4*c - 8*
a^2*b^2*c^2)*d + (a^2*b^3*c + 12*a^3*b*c^2)*f)*x^6 + (2*(a*b^5 - 6*a^2*b^3*
c - 16*a^3*b*c^2)*d + (a^2*b^4 + 14*a^3*b^2*c + 24*a^4*c^2)*f)*x^4 + 2*(2*(
a^2*b^4 - 8*a^3*b^2*c)*d + (a^3*b^3 + 12*a^4*b*c)*f)*x^2 + 2*(a^3*b^3 - 8*a
^4*b*c)*d + (a^4*b^2 + 12*a^5*c)*f)*sqrt((b^2 - 4*a*c)/a^2))*sqrt(a)*sqrt((
a*sqrt((b^2 - 4*a*c)/a^2) - b)/a)*elliptic_e(arcsin(sqrt(1/2)*x*sqrt((a*sqr
t((b^2 - 4*a*c)/a^2) - b)/a)), 1/2*(a*b*sqrt((b^2 - 4*a*c)/a^2) + b^2 - 2*a
*c)/(a*c)) + sqrt(1/2)*(((4*(5*a^2*b + 4*a*b^2)*c^3 - (a*b^3 + 2*b^4)*c^2)*
d - (12*a^2*b*c^3 + (8*a^2*b^2 + a*b^3)*c^2)*f)*x^8 + 2*((4*(5*a^2*b^2 + 4*
a*b^3)*c^2 - (a*b^4 + 2*b^5)*c)*d - (12*a^2*b^2*c^2 + (8*a^2*b^3 + a*b^4)*c
)*f)*x^6 - ((a*b^5 + 2*b^6 - 8*(5*a^3*b + 4*a^2*b^2)*c^2 - 6*(3*a^2*b^3 + 2
*a*b^4)*c)*d + (8*a^2*b^4 + a*b^5 + 24*a^3*b*c^2 + 2*(8*a^3*b^2 + 7*a^2*b^3
)*c)*f)*x^4 - 2*((a^2*b^4 + 2*a*b^5 - 4*(5*a^3*b^2 + 4*a^2*b^3)*c)*d + (8*a
^3*b^3 + a^2*b^4 + 12*a^3*b^2*c)*f)*x^2 - (a^3*b^3 + 2*a^2*b^4 - 4*(5*a^4*b
 + 4*a^3*b^2)*c)*d - (8*a^4*b^2 + a^3*b^3 + 12*a^4*b*c)*f + (((4*(5*a^3 - 4
*a^2*b)*c^3 - (a^2*b^2 - 2*a*b^3)*c^2)*d + (12*a^3*c^3 - (8*a^3*b - a^2*b^2
)*c^2)*f)*x^8 + 2*((4*(5*a^3*b - 4*a^2*b^2)*c^2 - (a^2*b^3 - 2*a*b^4)*c)*d
+ (12*a^3*b*c^2 - (8*a^3*b^2 - a^2*b^3)*c)*f)*x^6 - ((a^2*b^4 - 2*a*b^5 - 8
*(5*a^4 - 4*a^3*b)*c^2 - 6*(3*a^3*b^2 - 2*a^2*b^3)*c)*d + (8*a^3*b^3 - a^2*
b^4 - 24*a^4*c^2 + 2*(8*a^4*b - 7*a^3*b^2)*c)*f)*x^4 - 2*((a^3*b^3 - 2*a^2*
b^4 - 4*(5*a^4*b - 4*a^3*b^2)*c)*d + (8*a^4*b^2 - a^3*b^3 - 12*a^4*b*c)*f)*
x^2 - (a^4*b^2 - 2*a^3*b^3 - 4*(5*a^5 - 4*a^4*b)*c)*d - (8*a^5*b - a^4*b^2
- 12*a^5*c)*f)*sqrt((b^2 - 4*a*c)/a^2))*sqrt(a)*sqrt((a*sqrt((b^2 - 4*a*c)/
a^2) - b)/a)*elliptic_f(arcsin(sqrt(1/2)*x*sqrt((a*sqrt((b^2 - 4*a*c)/a^2)
- b)/a)), 1/2*(a*b*sqrt((b^2 - 4*a*c)/a^2) + b^2 - 2*a*c)/(a*c)) - 2*((2*(a
*b^3*c^2 - 8*a^2*b*c^3)*d + (a^2*b^2*c^2 + 12*a^3*c^3)*f)*x^7 + 8*(2*a^3*c^
3*e - a^3*b*c^2*g)*x^6 + ((4*a*b^4*c - 33*a^2*b^2*c^2 + 20*a^3*c^3)*d + 2*(
a^2*b^3*c + 8*a^3*b*c^2)*f)*x^5 + 12*(2*a^3*b*c^2*e - a^3*b^2*c*g)*x^4 + (2
*(a*b^5 - 7*a^2*b^3*c)*d + (a^2*b^4 + 3*a^3*b^2*c + 20*a^4*c^2)*f)*x^3 + 3*
(2*(a^3*b^2*c + 4*a^4*c^2)*e - (a^3*b^3 + 4*a^4*b*c)*g)*x^2 - (a^3*b^3 - 12
```

```
*a^4*b*c)*e - 2*(a^4*b^2 + 4*a^5*c)*g + (8*a^4*b*c*f + (3*a^2*b^4 - 23*a^3*
b^2*c + 28*a^4*c^2)*d)*x)*sqrt(c*x^4 + b*x^2 + a))/(a^5*b^4 - 8*a^6*b^2*c +
 16*a^7*c^2 + (a^3*b^4*c^2 - 8*a^4*b^2*c^3 + 16*a^5*c^4)*x^8 + 2*(a^3*b^5*c
 - 8*a^4*b^3*c^2 + 16*a^5*b*c^3)*x^6 + (a^3*b^6 - 6*a^4*b^4*c + 32*a^6*c^3)
*x^4 + 2*(a^4*b^5 - 8*a^5*b^3*c + 16*a^6*b*c^2)*x^2)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(c*x**4+b*x**2+a)**(5/2),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^{\frac{5}{2}}}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(5/2),x, algorithm="maxima")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a)^(5/2), x)
```

Giac [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^4+bx^2+a)^{\frac{5}{2}}}\,dx$$

```
[In] integrate((g*x^3+f*x^2+e*x+d)/(c*x^4+b*x^2+a)^(5/2),x, algorithm="giac")
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^4 + b*x^2 + a)^(5/2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^2+cx^4)^{5/2}}\,dx=\int \frac{g\,x^3+f\,x^2+e\,x+d}{(c\,x^4+b\,x^2+a)^{5/2}}\,dx$$

```
[In] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(5/2),x)
[Out] int((d + e*x + f*x^2 + g*x^3)/(a + b*x^2 + c*x^4)^(5/2), x)
```

3.108 $\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx$

Optimal result

Integrand size = 28, antiderivative size = 19

$$\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=\frac{gx}{\sqrt{a+bx^2+cx^4}}$$

```
[Out] g*x/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.01 (sec) , antiderivative size = 19, normalized size of antiderivative = 1.00, number of steps used = 1, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.036, Rules used = {1602}

$$\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=\frac{gx}{\sqrt{a+bx^2+cx^4}}$$

```
[In] Int[(a*g - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (g*x)/Sqrt[a + b*x^2 + c*x^4]
```

Rule 1602

```
Int[(Pp_)*(Qq_)^(m_.), x_Symbol] :> With[{p = Expon[Pp, x], q = Expon[Qq, x
]}, Simp[Coeff[Pp, x, p]*x^(p - q + 1)*(Qq^(m + 1)/((p + m*q + 1)*Coeff[Qq,
 x, q])), x] /; NeQ[p + m*q + 1, 0] && EqQ[(p + m*q + 1)*Coeff[Qq, x, q]*Pp
, Coeff[Pp, x, p]*x^(p - q)*((p - q + 1)*Qq + (m + 1)*x*D[Qq, x])]] /; Free
Q[m, x] && PolyQ[Pp, x] && PolyQ[Qq, x] && NeQ[m, -1]
```

Rubi steps

$$\text{integral}=\frac{gx}{\sqrt{a+bx^2+cx^4}}$$

Mathematica [A] (verified)

Time = 10.09 (sec) , antiderivative size = 19, normalized size of antiderivative = 1.00

$$\int \frac{ag - cgx^4}{(a + bx^2 + cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{a + bx^2 + cx^4}}$$

```
[In] Integrate[(a*g - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
[Out] (g*x)/Sqrt[a + b*x^2 + c*x^4]
```

Maple [A] (verified)

Time = 1.10 (sec) , antiderivative size = 18, normalized size of antiderivative = 0.95

method	result	size
gosper	$\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$	18
default	$\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$	18
trager	$\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$	18
elliptic	$\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$	18
pseudoelliptic	$\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$	18

```
[In] int((-c*g*x^4+a*g)/(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
[Out] g*x/(c*x^4+b*x^2+a)^(1/2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 17, normalized size of antiderivative = 0.89

$$\int \frac{ag - cgx^4}{(a + bx^2 + cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{cx^4 + bx^2 + a}}$$

```
[In] integrate((-c*g*x^4+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="fricas")
[Out] g*x/sqrt(c*x^4 + b*x^2 + a)
```

Sympy [F]

$$\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -g\left(\int \left(-\frac{a}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx + \int \frac{cx^4}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\,dx\right)$$

```
[In] integrate((-c*g*x**4+a*g)/(c*x**4+b*x**2+a)**(3/2),x)
[Out] -g*(Integral(-a/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x
**4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x) + Integral(c*x**4/(a*sqrt(a +
b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**
2 + c*x**4)), x))
```

Maxima [A] (verification not implemented)

none

Time = 0.22 (sec) , antiderivative size = 17, normalized size of antiderivative = 0.89

$$\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{cx^4+bx^2+a}}$$

```
[In] integrate((-c*g*x^4+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxima")
[Out] g*x/sqrt(c*x^4 + b*x^2 + a)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 60 vs. $2(17) = 34$.

Time = 0.67 (sec) , antiderivative size = 60, normalized size of antiderivative = 3.16

$$\int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{(b^4g-8\,ab^2cg+16\,a^2c^2g)x}{\sqrt{cx^4+bx^2+a}(b^4-8\,ab^2c+16\,a^2c^2)}$$

```
[In] integrate((-c*g*x^4+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac")
[Out] (b^4*g - 8*a*b^2*c*g + 16*a^2*c^2*g)*x/(sqrt(c*x^4 + b*x^2 + a)*(b^4 - 8*a*
b^2*c + 16*a^2*c^2))
```

Mupad [B] (verification not implemented)

Time = 8.01 (sec) , antiderivative size = 17, normalized size of antiderivative = 0.89

$$\int \frac{ag - cgx^4}{(a + bx^2 + cx^4)^{3/2}}\, dx = \frac{g\,x}{\sqrt{c\,x^4 + b\,x^2 + a}}$$

```
[In] int((a*g - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x)
[Out] (g*x)/(a + b*x^2 + c*x^4)^(1/2)
```

3.109 $\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx$

Optimal result

Integrand size = 31, antiderivative size = 57

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{e(b+2cx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[Out] g*x/(c*x^4+b*x^2+a)^(1/2)-e*(2*c*x^2+b)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.04 (sec) , antiderivative size = 57, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.161, Rules used = {1687, 1602, 12, 1121, 627}

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{e(b+2cx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[In] Int[(a*g + e*x - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (g*x)/Sqrt[a + b*x^2 + c*x^4] - (e*(b + 2*c*x^2))/((b^2 - 4*a*c)*Sqrt[a + b
*x^2 + c*x^4])
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 627

```
Int[((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(-3/2), x_Symbol] :> Simp[-2*((b +
2*c*x)/((b^2 - 4*a*c)*Sqrt[a + b*x + c*x^2])), x] /; FreeQ[{a, b, c}, x] &&
```

```
NeQ[b^2 - 4*a*c, 0]
```

Rule 1121

```
Int[(x_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dist[1/2,
 Subst[Int[(a + b*x + c*x^2)^p, x], x, x^2], x] /; FreeQ[{a, b, c, p}, x]
```

Rule 1602

```
Int[(Pp_)*(Qq_)^(m_.), x_Symbol] :> With[{p = Expon[Pp, x], q = Expon[Qq, x
]}, Simp[Coeff[Pp, x, p]*x^(p - q + 1)*(Qq^(m + 1)/((p + m*q + 1)*Coeff[Qq,
 x, q])), x] /; NeQ[p + m*q + 1, 0] && EqQ[(p + m*q + 1)*Coeff[Qq, x, q]*Pp
, Coeff[Pp, x, p]*x^(p - q)*((p - q + 1)*Qq + (m + 1)*x*D[Qq, x])]] /; Free
Q[m, x] && PolyQ[Pp, x] && PolyQ[Qq, x] && NeQ[m, -1]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{ex}{(a+bx^2+cx^4)^{3/2}}\,dx + \int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + e\int \frac{x}{(a+bx^2+cx^4)^{3/2}}\,dx \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + \frac{1}{2}e\text{Subst}\left(\int \frac{1}{(a+bx+cx^2)^{3/2}}\,dx, x, x^2\right) \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{e(b+2cx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}
\end{aligned}$$

Mathematica [A] (verified)

Time = 10.13 (sec) , antiderivative size = 51, normalized size of antiderivative = 0.89

$$\int \frac{ag+ex-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx=\frac{-be+b^2gx-4acgx-2cex^2}{\left(b^2-4ac\right)\sqrt{a+bx^2+cx^4}}$$

```
[In] Integrate[(a*g + e*x - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (-(b*e) + b^2*g*x - 4*a*c*g*x - 2*c*e*x^2)/((b^2 - 4*a*c)*Sqrt[a + b*x^2 +
c*x^4])
```

Maple [A] (verified)

Time = 1.60 (sec) , antiderivative size = 52, normalized size of antiderivative = 0.91

method	result
gosper	$\frac{4acgx-b^2gx+2c\,x^2e+be}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}$
trager	$\frac{4acgx-b^2gx+2c\,x^2e+be}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}$
elliptic	$\frac{e(2c\,x^2+b)}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}+\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$
default	$ag\left(-\frac{2c\left(\frac{b\,x^3}{2a\left(4ac-b^2\right)}-\frac{\left(2ac-b^2\right)x}{2a\left(4ac-b^2\right)c}\right)}{\sqrt{\left(x^4+\frac{b\,x^2}{c}+\frac{a}{c}\right)c}}+\frac{\left(\frac{1}{a}-\frac{2ac-b^2}{a\left(4ac-b^2\right)}\right)\sqrt{2}\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\sqrt{4+\frac{2\left(b+\sqrt{-4ac+b^2}\right)x^2}{a}}\,F\left(\frac{x\sqrt{2}\sqrt{\frac{-b+\sqrt{}}{}}}{2}\right.}{4\sqrt{\frac{-b+\sqrt{-4ac+b^2}}{a}}\sqrt{c\,x^4+b\,x^2+a}}\right.$

```
[In] int((-c*g*x^4+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] (4*a*c*g*x-b^2*g*x+2*c*e*x^2+b*e)/(c*x^4+b*x^2+a)^(1/2)/(4*a*c-b^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 82, normalized size of antiderivative = 1.44

$$\int \frac{ag+ex-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx=-\frac{\sqrt{cx^4+bx^2+a}(2\,cex^2-(b^2-4\,ac)gx+be)}{(b^2c-4\,ac^2)x^4+ab^2-4\,a^2c+(b^3-4\,abc)x^2}$$

```
[In] integrate((-c*g*x^4+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="fricas")
```

```
[Out] -sqrt(c*x^4 + b*x^2 + a)*(2*c*e*x^2 - (b^2 - 4*a*c)*g*x + b*e)/((b^2*c - 4*
a*c^2)*x^4 + a*b^2 - 4*a^2*c + (b^3 - 4*a*b*c)*x^2)
```

Sympy [F]

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -\int \left(-\frac{ag}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx - \int \left(-\frac{ex}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx - \int \frac{cgx^4}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\,dx$$

```
[In] integrate((-c*g*x**4+a*g+e*x)/(c*x**4+b*x**2+a)**(3/2),x)
[Out] -Integral(-a*g/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x*
*4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x) - Integral(-e*x/(a*sqrt(a + b*x
**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**2 +
 c*x**4)), x) - Integral(c*g*x**4/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqr
t(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x)
```

Maxima [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 51, normalized size of antiderivative = 0.89

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -\frac{2\,cex^2+be-(b^2g-4\,acg)x}{\sqrt{cx^4+bx^2+a}(b^2-4\,ac)}$$

```
[In] integrate((-c*g*x^4+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxima")
[Out] -(2*c*e*x^2 + b*e - (b^2*g - 4*a*c*g)*x)/(sqrt(c*x^4 + b*x^2 + a)*(b^2 - 4*
a*c))
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 138 vs. 2(53) = 106.

Time = 0.66 (sec) , antiderivative size = 138, normalized size of antiderivative = 2.42

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -\frac{\left(\frac{2\,(b^2ce-4\,ac^2e)x}{b^4-8\,ab^2c+16\,a^2c^2}-\frac{b^4g-8\,ab^2cg+16\,a^2c^2g}{b^4-8\,ab^2c+16\,a^2c^2}\right)x+\frac{b^3e-4\,abce}{b^4-8\,ab^2c+16\,a^2c^2}}{\sqrt{cx^4+bx^2+a}}$$

```
[In] integrate((-c*g*x^4+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac")
[Out] -((2*(b^2*c*e - 4*a*c^2*e)*x/(b^4 - 8*a*b^2*c + 16*a^2*c^2) - (b^4*g - 8*a*
b^2*c*g + 16*a^2*c^2*g)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))*x + (b^3*e - 4*a*b*
c*e)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))/sqrt(c*x^4 + b*x^2 + a)
```

Mupad [B] (verification not implemented)

Time = 7.92 (sec) , antiderivative size = 51, normalized size of antiderivative = 0.89

$$\int \frac{ag+ex-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{-g\,b^2\,x+e\,b+2\,c\,e\,x^2+4\,a\,c\,g\,x}{(4\,a\,c-b^2)\,\sqrt{c\,x^4+b\,x^2+a}}$$

```
[In] int((a*g + e*x - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x)
```

```
[Out] (b*e + 2*c*e*x^2 - b^2*g*x + 4*a*c*g*x)/((4*a*c - b^2)*(a + b*x^2 + c*x^4)^
(1/2))
```

3.110 $\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx$

Optimal result

Integrand size = 33, antiderivative size = 57

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=\frac{gx}{\sqrt{a+bx^2+cx^4}}+\frac{f(2a+bx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[Out] g*x/(c*x^4+b*x^2+a)^(1/2)+f*(b*x^2+2*a)/(-4*a*c+b^2)/(c*x^4+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 57, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.152, Rules used = {1687, 1602, 12, 1128, 650}

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=\frac{f(2a+bx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}+\frac{gx}{\sqrt{a+bx^2+cx^4}}$$

```
[In] Int[(a*g + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (g*x)/Sqrt[a + b*x^2 + c*x^4] + (f*(2*a + b*x^2))/((b^2 - 4*a*c)*Sqrt[a + b
*x^2 + c*x^4])
```

Rule 12

```
Int[(a_)*(u_), x_Symbol] :> Dist[a, Int[u, x], x] /; FreeQ[a, x] &&  !Match
Q[u, (b_)*(v_) /; FreeQ[b, x]]
```

Rule 650

```
Int[((d_.) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(3/2), x_Symbo
l] :> Simp[-2*((b*d - 2*a*e + (2*c*d - b*e)*x)/((b^2 - 4*a*c)*Sqrt[a + b*x
```

```
+ c*x^2])), x] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b
^2 - 4*a*c, 0]
```

Rule 1128

```
Int[(x_)^(m_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_.), x_Symbol] :> Dis
t[1/2, Subst[Int[x^((m - 1)/2)*(a + b*x + c*x^2)^p, x], x, x^2], x] /; Free
Q[{a, b, c, p}, x] && IntegerQ[(m - 1)/2]
```

Rule 1602

```
Int[(Pp_)*(Qq_)^(m_.), x_Symbol] :> With[{p = Expon[Pp, x], q = Expon[Qq, x
]}, Simp[Coeff[Pp, x, p]*x^(p - q + 1)*(Qq^(m + 1)/((p + m*q + 1)*Coeff[Qq,
 x, q])), x] /; NeQ[p + m*q + 1, 0] && EqQ[(p + m*q + 1)*Coeff[Qq, x, q]*Pp
, Coeff[Pp, x, p]*x^(p - q)*((p - q + 1)*Qq + (m + 1)*x*D[Qq, x])]] /; Free
Q[m, x] && PolyQ[Pp, x] && PolyQ[Qq, x] && NeQ[m, -1]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{fx^3}{(a+bx^2+cx^4)^{3/2}}\,dx + \int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + f\int \frac{x^3}{(a+bx^2+cx^4)^{3/2}}\,dx \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + \frac{1}{2}f\,\text{Subst}\left(\int \frac{x}{(a+bx+cx^2)^{3/2}}\,dx, x, x^2\right) \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + \frac{f(2a+bx^2)}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}
\end{aligned}$$

Mathematica [A] (verified)

Time = 10.08 (sec) , antiderivative size = 48, normalized size of antiderivative = 0.84

$$\int \frac{ag+fx^3-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx=\frac{bx(bg+fx)+2a(f-2cgx)}{\left(b^2-4ac\right)\sqrt{a+bx^2+cx^4}}$$

```
[In] Integrate[(a*g + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
[Out] (b*x*(b*g + f*x) + 2*a*(f - 2*c*g*x))/((b^2 - 4*a*c)*Sqrt[a + b*x^2 + c*x^4
])
```

Maple [A] (verified)

Time = 1.60 (sec) , antiderivative size = 53, normalized size of antiderivative = 0.93

method	result
gosper	$\frac{4acgx-b^2gx-bf\,x^2-2af}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}$
trager	$\frac{4acgx-b^2gx-bf\,x^2-2af}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}$
elliptic	$-\frac{f\left(b\,x^2+2a\right)}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}+\frac{gx}{\sqrt{c\,x^4+b\,x^2+a}}$
default	$-\frac{f\left(b\,x^2+2a\right)}{\sqrt{c\,x^4+b\,x^2+a}\,(4ac-b^2)}+ag\left(-\frac{2c\left(\frac{b\,x^3}{2a\left(4ac-b^2\right)}-\frac{\left(2ac-b^2\right)x}{2a\left(4ac-b^2\right)c}\right)}{\sqrt{\left(x^4+\frac{b\,x^2}{c}+\frac{a}{c}\right)c}}+\frac{\left(\frac{1}{a}-\frac{2ac-b^2}{a\left(4ac-b^2\right)}\right)\sqrt{2}\sqrt{4-\frac{2\left(-b+\sqrt{-4ac+b^2}\right)x^2}{a}}\sqrt{4+\cdots}}{4\sqrt{\frac{-b+\cdots}{}}}\cdots\right.$

```
[In] int((-c*g*x^4+f*x^3+a*g)/(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
[Out] (4*a*c*g*x-b^2*g*x-b*f*x^2-2*a*f)/(c*x^4+b*x^2+a)^(1/2)/(4*a*c-b^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 80, normalized size of antiderivative = 1.40

$$\int \frac{ag+fx^3-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx=\frac{\sqrt{cx^4+bx^2+a}\left(bfx^2+(b^2-4\,ac)gx+2\,af\right)}{(b^2c-4\,ac^2)x^4+ab^2-4\,a^2c+(b^3-4\,abc)x^2}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="fricas")
[Out] sqrt(c*x^4 + b*x^2 + a)*(b*f*x^2 + (b^2 - 4*a*c)*g*x + 2*a*f)/((b^2*c - 4*a
*c^2)*x^4 + a*b^2 - 4*a^2*c + (b^3 - 4*a*b*c)*x^2)
```

Sympy [F]

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -\int \left(-\frac{ag}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx - \int \left(-\frac{fx^3}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx - \int \frac{cgx^4}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\,dx$$

```
[In] integrate((-c*g*x**4+f*x**3+a*g)/(c*x**4+b*x**2+a)**(3/2),x)
[Out] -Integral(-a*g/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x*
*4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x) - Integral(-f*x**3/(a*sqrt(a +
b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**
2 + c*x**4)), x) - Integral(c*g*x**4/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*
sqrt(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x)
```

Maxima [A] (verification not implemented)

none

Time = 0.23 (sec) , antiderivative size = 49, normalized size of antiderivative = 0.86

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{bfx^2+2\,af+(b^2g-4\,acg)x}{\sqrt{cx^4+bx^2+a}(b^2-4\,ac)}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxima")
[Out] (b*f*x^2 + 2*a*f + (b^2*g - 4*a*c*g)*x)/(sqrt(c*x^4 + b*x^2 + a)*(b^2 - 4*a
*c))
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 136 vs. 2(53) = 106.

Time = 0.62 (sec) , antiderivative size = 136, normalized size of antiderivative = 2.39

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{\left(\frac{(b^3f-4\,abcf)x}{b^4-8\,ab^2c+16\,a^2c^2}+\frac{b^4g-8\,ab^2cg+16\,a^2c^2g}{b^4-8\,ab^2c+16\,a^2c^2}\right)x+\frac{2\,(ab^2f-4\,a^2cf)}{b^4-8\,ab^2c+16\,a^2c^2}}{\sqrt{cx^4+bx^2+a}}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac")
[Out] (((b^3*f - 4*a*b*c*f)*x/(b^4 - 8*a*b^2*c + 16*a^2*c^2) + (b^4*g - 8*a*b^2*c
*g + 16*a^2*c^2*g)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))*x + 2*(a*b^2*f - 4*a^2*c
*f)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))/sqrt(c*x^4 + b*x^2 + a)
```

Mupad [B] (verification not implemented)

Time = 7.94 (sec) , antiderivative size = 51, normalized size of antiderivative = 0.89

$$\int \frac{ag+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = -\frac{g\,b^2\,x+f\,b\,x^2-4\,a\,c\,g\,x+2\,a\,f}{(4\,a\,c-b^2)\,\sqrt{c\,x^4+b\,x^2+a}}$$

```
[In] int((a*g + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x)
```

```
[Out] -(2*a*f + b*f*x^2 + b^2*g*x - 4*a*c*g*x)/((4*a*c - b^2)*(a + b*x^2 + c*x^4)
^(1/2))
```

3.111 $\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx$

Optimal result

Integrand size = 36, antiderivative size = 69

$$\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{be-2af+(2ce-bf)x^2}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[Out] g*x/(c*x^4+b*x^2+a)^(1/2)+(-b*e+2*a*f-(-b*f+2*c*e)*x^2)/(-4*a*c+b^2)/(c*x^4
+b*x^2+a)^(1/2)
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.111, Rules used = {1687, 1602, 1261, 650}

$$\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{-2af+x^2(2ce-bf)+be}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[In] Int[(a*g + e*x + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
```

```
[Out] (g*x)/Sqrt[a + b*x^2 + c*x^4] - (b*e - 2*a*f + (2*c*e - b*f)*x^2)/((b^2 - 4
*a*c)*Sqrt[a + b*x^2 + c*x^4])
```

Rule 650

```
Int[((d_.) + (e_.)*(x_))/((a_.) + (b_.)*(x_) + (c_.)*(x_)^2)^(3/2), x_Symbo
l] :> Simp[-2*((b*d - 2*a*e + (2*c*d - b*e)*x)/((b^2 - 4*a*c)*Sqrt[a + b*x
+ c*x^2])), x] /; FreeQ[{a, b, c, d, e}, x] && NeQ[2*c*d - b*e, 0] && NeQ[b
^2 - 4*a*c, 0]
```

Rule 1261

```
Int[(x_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(
p_.), x_Symbol] :> Dist[1/2, Subst[Int[(d + e*x)^q*(a + b*x + c*x^2)^p, x],
 x, x^2], x] /; FreeQ[{a, b, c, d, e, p, q}, x]
```

Rule 1602

```
Int[(Pp_)*(Qq_)^(m_.), x_Symbol] :> With[{p = Expon[Pp, x], q = Expon[Qq, x
]}, Simp[Coeff[Pp, x, p]*x^(p - q + 1)*(Qq^(m + 1)/((p + m*q + 1)*Coeff[Qq,
 x, q])), x] /; NeQ[p + m*q + 1, 0] && EqQ[(p + m*q + 1)*Coeff[Qq, x, q]*Pp
, Coeff[Pp, x, p]*x^(p - q)*((p - q + 1)*Qq + (m + 1)*x*D[Qq, x])]] /; Free
Q[m, x] && PolyQ[Pp, x] && PolyQ[Qq, x] && NeQ[m, -1]
```

Rule 1687

```
Int[(Pq_)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^(p_), x_Symbol] :> Module[{q
 = Expon[Pq, x], k}, Int[Sum[Coeff[Pq, x, 2*k]*x^(2*k), {k, 0, q/2}]*(a + b
*x^2 + c*x^4)^p, x] + Int[x*Sum[Coeff[Pq, x, 2*k + 1]*x^(2*k), {k, 0, (q -
1)/2}]*(a + b*x^2 + c*x^4)^p, x]] /; FreeQ[{a, b, c, p}, x] && PolyQ[Pq, x]
 &&  !PolyQ[Pq, x^2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{x(e+fx^2)}{(a+bx^2+cx^4)^{3/2}}\,dx + \int \frac{ag-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} + \frac{1}{2}\text{Subst}\left(\int \frac{e+fx}{(a+bx+cx^2)^{3/2}}\,dx, x, x^2\right) \\
&= \frac{gx}{\sqrt{a+bx^2+cx^4}} - \frac{be-2af+(2ce-bf)x^2}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}
\end{aligned}$$

Mathematica [A] (verified)

Time = 10.16 (sec) , antiderivative size = 61, normalized size of antiderivative = 0.88

$$\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx = \frac{-be+2af+b^2gx-4acgx-2cex^2+bfx^2}{(b^2-4ac)\sqrt{a+bx^2+cx^4}}$$

```
[In] Integrate[(a*g + e*x + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x]
[Out] (-(b*e) + 2*a*f + b^2*g*x - 4*a*c*g*x - 2*c*e*x^2 + b*f*x^2)/((b^2 - 4*a*c)
*Sqrt[a + b*x^2 + c*x^4])
```

Maple [A] (verified)

Time = 1.79 (sec) , antiderivative size = 63, normalized size of antiderivative = 0.91

method	result	size
gosper	$\frac{4acgx-b^{2}gx-bf\,x^{2}+2c\,x^{2}e-2af+be}{\sqrt{c\,x^{4}+b\,x^{2}+a}\,(4ac-b^{2})}$	63
trager	$\frac{4acgx-b^{2}gx-bf\,x^{2}+2c\,x^{2}e-2af+be}{\sqrt{c\,x^{4}+b\,x^{2}+a}\,(4ac-b^{2})}$	63
elliptic	$-\frac{bf\,x^{2}-2c\,x^{2}e+2af-be}{\sqrt{c\,x^{4}+b\,x^{2}+a}\,(4ac-b^{2})}+\frac{gx}{\sqrt{c\,x^{4}+b\,x^{2}+a}}$	69
default	Expression too large to display	1012

```
[In] int((-c*g*x^4+f*x^3+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x,method=_RETURNVERBOSE)
```

```
[Out] (4*a*c*g*x-b^2*g*x-b*f*x^2+2*c*e*x^2-2*a*f+b*e)/(c*x^4+b*x^2+a)^(1/2)/(4*a*
c-b^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.33

$$\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=\frac{\sqrt{cx^4+bx^2+a}((b^2-4\,ac)gx-(2\,ce-bf)x^2-be+2\,af)}{(b^2c-4\,ac^2)x^4+ab^2-4\,a^2c+(b^3-4\,abc)x^2}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="fric
as")
```

```
[Out] sqrt(c*x^4 + b*x^2 + a)*((b^2 - 4*a*c)*g*x - (2*c*e - b*f)*x^2 - b*e + 2*a*
f)/((b^2*c - 4*a*c^2)*x^4 + a*b^2 - 4*a^2*c + (b^3 - 4*a*b*c)*x^2)
```

Sympy [F]

$$\begin{aligned}\int \frac{ag+ex+fx^3-cgx^4}{(a+bx^2+cx^4)^{3/2}}\,dx=&\\ -\int\left(-\frac{ag}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx\\ -\int\left(-\frac{ex}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx\\ -\int\left(-\frac{fx^3}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\right)dx\\ -\int\frac{cgx^4}{a\sqrt{a+bx^2+cx^4}+bx^2\sqrt{a+bx^2+cx^4}+cx^4\sqrt{a+bx^2+cx^4}}\,dx\end{aligned}$$

```
[In] integrate((-c*g*x**4+f*x**3+a*g+e*x)/(c*x**4+b*x**2+a)**(3/2),x)
```

```
[Out] -Integral(-a*g/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x*
*4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x) - Integral(-e*x/(a*sqrt(a + b*x
**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**2 +
 c*x**4)), x) - Integral(-f*x**3/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt
(a + b*x**2 + c*x**4) + c*x**4*sqrt(a + b*x**2 + c*x**4)), x) - Integral(c*
g*x**4/(a*sqrt(a + b*x**2 + c*x**4) + b*x**2*sqrt(a + b*x**2 + c*x**4) + c*
x**4*sqrt(a + b*x**2 + c*x**4)), x)
```

Maxima [A] (verification not implemented)

none

Time = 0.23 (sec) , antiderivative size = 94, normalized size of antiderivative = 1.36

$$\int \frac{ag+ex+fx^3-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx=-\frac{\sqrt{cx^4+bx^2+a}\left((2\,ce-bf)x^2+be-2\,af-(b^2g-4\,acg)x\right)}{(b^2c-4\,ac^2)x^4+ab^2-4\,a^2c+(b^3-4\,abc)x^2}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="maxi
ma")
```

```
[Out] -sqrt(c*x^4 + b*x^2 + a)*((2*c*e - b*f)*x^2 + b*e - 2*a*f - (b^2*g - 4*a*c*
g)*x)/((b^2*c - 4*a*c^2)*x^4 + a*b^2 - 4*a^2*c + (b^3 - 4*a*b*c)*x^2)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 164 vs. 2(65) = 130.

Time = 0.66 (sec) , antiderivative size = 164, normalized size of antiderivative = 2.38

$$\int \frac{ag+ex+fx^3-cgx^4}{\left(a+bx^2+cx^4\right)^{3/2}}\,dx = -\frac{\left(\frac{(2\,b^2ce-8\,ac^2e-b^3f+4\,abcf)x}{b^4-8\,ab^2c+16\,a^2c^2}-\frac{b^4g-8\,ab^2cg+16\,a^2c^2g}{b^4-8\,ab^2c+16\,a^2c^2}\right)x+\frac{b^3e-4\,abce-2\,ab^2f+8\,a^2cf}{b^4-8\,ab^2c+16\,a^2c^2}}{\sqrt{cx^4+bx^2+a}}$$

```
[In] integrate((-c*g*x^4+f*x^3+a*g+e*x)/(c*x^4+b*x^2+a)^(3/2),x, algorithm="giac
")
```

```
[Out] -(((2*b^2*c*e - 8*a*c^2*e - b^3*f + 4*a*b*c*f)*x/(b^4 - 8*a*b^2*c + 16*a^2*
c^2) - (b^4*g - 8*a*b^2*c*g + 16*a^2*c^2*g)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))
*x + (b^3*e - 4*a*b*c*e - 2*a*b^2*f + 8*a^2*c*f)/(b^4 - 8*a*b^2*c + 16*a^2*
c^2))/sqrt(c*x^4 + b*x^2 + a)
```

Mupad [B] (verification not implemented)

Time = 8.00 (sec) , antiderivative size = 62, normalized size of antiderivative = 0.90

$$\int \frac{ag + ex + fx^3 - cgx^4}{(a + bx^2 + cx^4)^{3/2}} \, dx = -\frac{g\,b^2\,x + f\,b\,x^2 - e\,b - 2\,c\,e\,x^2 - 4\,a\,c\,g\,x + 2\,a\,f}{(4\,a\,c - b^2)\,\sqrt{c\,x^4 + b\,x^2 + a}}$$

```
[In] int((a*g + e*x + f*x^3 - c*g*x^4)/(a + b*x^2 + c*x^4)^(3/2),x)
[Out] -(2*a*f - b*e + b*f*x^2 - 2*c*e*x^2 + b^2*g*x - 4*a*c*g*x)/((4*a*c - b^2)*(
a + b*x^2 + c*x^4)^(1/2))
```

CHAPTER 4

APPENDIX

4.1 Listing of Grading functions

The following are the current version of the grading functions used for grading the quality of the antiderivative with reference to the optimal antiderivative included in the test suite.

There is a version for Maple and for Mathematica/Rubi. There is a version for grading Sympy and version for use with Sagemath.

The following are links to the current source code.

The following are the listings of source code of the grading functions.

Mathematica and Rubi grading function

```
(* Original version thanks to Albert Rich emailed on 03/21/2017 *)
(* ::Package:: *)

(* Nasser: April 7,2022. add second output which gives reason for the grade *)
(*                       Small rewrite of logic in main function to make it*)
(*                       match Maple's logic. No change in functionality otherwise*)

(* ::Subsection:: *)
(*GradeAntiderivative[result,optimal]*)


(* ::Text:: *)
(*If result and optimal are mathematical expressions, *)
```

```
(*          GradeAntiderivative[result,optimal] returns*)
(*  "F" if the result fails to integrate an expression that*)
(*      is integrable*)
(*  "C" if result involves higher level functions than necessary*)
(*  "B" if result is more than twice the size of the optimal*)
(*      antiderivative*)
(*  "A" if result can be considered optimal*)


GradeAntiderivative[result_,optimal_] := Module[{expnResult,expnOptimal,leafCountResult,leafCou
    expnResult  = ExpnType[result];
    expnOptimal = ExpnType[optimal];
    leafCountResult = LeafCount[result];
    leafCountOptimal = LeafCount[optimal];

    (*Print["expnResult=",expnResult," expnOptimal=",expnOptimal];*)
    If[expnResult<=expnOptimal,
        If[Not[FreeQ[result,Complex]], (*result contains complex*)
            If[Not[FreeQ[optimal,Complex]], (*optimal contains complex*)
                If[leafCountResult<=2*leafCountOptimal,
                    finalresult={"A",""}
                    ,(*ELSE*)
                    finalresult={"B","Both result and optimal contain complex but leaf count is
                  ]
            ,(*ELSE*)
                finalresult={"C","Result contains complex when optimal does not."}
            ]
        ,(*ELSE*)(*result does not contains complex*)
            If[leafCountResult<=2*leafCountOptimal,
               finalresult={"A",""}
            ,(*ELSE*)
               finalresult={"B","Leaf count is larger than twice the leaf count of optimal. $"<
              ]
        ]
    ,(*ELSE*) (*expnResult>expnOptimal*)
        If[FreeQ[result,Integrate] && FreeQ[result,Int],
            finalresult={"C","Result contains higher order function than in optimal. Order "<>T
            ,
            finalresult={"F","Contains unresolved integral."}
        ]
    ];

    finalresult
]

(* ::Text:: *)
(*The following summarizes the type number assigned an *)
```

```
(*expression based on the functions it involves*)
(*1 = rational function*)
(*2 = algebraic function*)
(*3 = elementary function*)
(*4 = special function*)
(*5 = hyperpergeometric function*)
(*6 = appell function*)
(*7 = rootsum function*)
(*8 = integrate function*)
(*9 = unknown function*)


ExpnType[expn_] :=
  If[AtomQ[expn],
    1,
  If[ListQ[expn],
    Max[Map[ExpnType,expn]],
  If[Head[expn]===Power,
    If[IntegerQ[expn[[2]]],
      ExpnType[expn[[1]]],
    If[Head[expn[[2]]]===Rational,
      If[IntegerQ[expn[[1]]] || Head[expn[[1]]]===Rational,
        1,
      Max[ExpnType[expn[[1]]],2]],
    Max[ExpnType[expn[[1]]],ExpnType[expn[[2]]],3]]],
  If[Head[expn]===Plus || Head[expn]===Times,
    Max[ExpnType[First[expn]],ExpnType[Rest[expn]]],
  If[ElementaryFunctionQ[Head[expn]],
    Max[3,ExpnType[expn[[1]]]],
  If[SpecialFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],4]],
  If[HypergeometricFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],5]],
  If[AppellFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],6]],
  If[Head[expn]===RootSum,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
  If[Head[expn]===Integrate || Head[expn]===Int,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],8]],
  9]]]]]]]]]]


ElementaryFunctionQ[func_] :=
  MemberQ[{
  Exp,Log,
  Sin,Cos,Tan,Cot,Sec,Csc,
  ArcSin,ArcCos,ArcTan,ArcCot,ArcSec,ArcCsc,
```

```
  Sinh,Cosh,Tanh,Coth,Sech,Csch,
  ArcSinh,ArcCosh,ArcTanh,ArcCoth,ArcSech,ArcCsch
},func]


SpecialFunctionQ[func_] :=
  MemberQ[{
  Erf, Erfc, Erfi,
  FresnelS, FresnelC,
  ExpIntegralE, ExpIntegralEi, LogIntegral,
  SinIntegral, CosIntegral, SinhIntegral, CoshIntegral,
  Gamma, LogGamma, PolyGamma,
  Zeta, PolyLog, ProductLog,
  EllipticF, EllipticE, EllipticPi
},func]


HypergeometricFunctionQ[func_] :=
  MemberQ[{Hypergeometric1F1,Hypergeometric2F1,HypergeometricPFQ},func]


AppellFunctionQ[func_] :=
  MemberQ[{AppellF1},func]
```

Maple grading function

```
# File: GradeAntiderivative.mpl
# Original version thanks to Albert Rich emailed on 03/21/2017

#Nasser 03/22/2017  Use Maple leaf count instead since buildin
#Nasser 03/23/2017  missing 'ln' for ElementaryFunctionQ added
#Nasser 03/24/2017  corrected the check for complex result
#Nasser 10/27/2017  check for leafsize and do not call ExpnType()
#                   if leaf size is "too large". Set at 500,000
#Nasser 12/22/2019  Added debug flag, added 'dilog' to special functions
#                   see problem 156, file Apostol_Problems
#Nasser 4/07/2022   add second output which gives reason for the grade

GradeAntiderivative := proc(result,optimal)
local leaf_count_result,
        leaf_count_optimal,
        ExpnType_result,
        ExpnType_optimal,
        debug:=false;

        leaf_count_result:=leafcount(result);
```

```
        #do NOT call ExpnType() if leaf size is too large. Recursion problem
        if leaf_count_result > 500000 then
            return "B","result has leaf size over 500,000. Avoiding possible recursion issues
        fi;

        leaf_count_optimal := leafcount(optimal);
        ExpnType_result  := ExpnType(result);
        ExpnType_optimal := ExpnType(optimal);

        if debug then
            print("ExpnType_result",ExpnType_result," ExpnType_optimal=",ExpnType_optimal);
        fi;

# If result and optimal are mathematical expressions,
#  GradeAntiderivative[result,optimal] returns
#   "F" if the result fails to integrate an expression that
#       is integrable
#   "C" if result involves higher level functions than necessary
#   "B" if result is more than twice the size of the optimal
#       antiderivative
#   "A" if result can be considered optimal

    #This check below actually is not needed, since I only
    #call this grading only for passed integrals. i.e. I check
    #for "F" before calling this. But no harm of keeping it here.
    #just in case.


    if not type(result,freeof('int')) then
        return "F","Result contains unresolved integral";
    fi;


    if ExpnType_result<=ExpnType_optimal then
        if debug then
             print("ExpnType_result<=ExpnType_optimal");
        fi;
        if is_contains_complex(result) then
            if is_contains_complex(optimal) then
                if debug then
                        print("both result and optimal complex");
                fi;
                if leaf_count_result<=2*leaf_count_optimal then
                    return "A"," ";
                else
                   return "B",cat("Both result and optimal contain complex but leaf count of
                                convert(leaf_count_result,string)," vs. $2 (",
```

```
                                   convert(leaf_count_optimal,string)," ) = ",convert(2*leaf_c
                   end if
               else #result contains complex but optimal is not
                   if debug then
                           print("result contains complex but optimal is not");
                   fi;
                   return "C","Result contains complex when optimal does not.";
               fi;
           else # result do not contain complex
                # this assumes optimal do not as well. No check is needed here.
               if debug then
                      print("result do not contain complex, this assumes optimal do not as well")
               fi;
               if leaf_count_result<=2*leaf_count_optimal then
                   if debug then
                       print("leaf_count_result<=2*leaf_count_optimal");
                   fi;
                   return "A"," ";
               else
                   if debug then
                       print("leaf_count_result>2*leaf_count_optimal");
                   fi;
                   return "B",cat("Leaf count of result is larger than twice the leaf count of op
                                   convert(leaf_count_result,string),"$ vs. $2(",
                                   convert(leaf_count_optimal,string),")=",convert(2*leaf_coun
               fi;
           fi;
       else #ExpnType(result) > ExpnType(optimal)
           if debug then
               print("ExpnType(result) > ExpnType(optimal)");
           fi;
           return "C",cat("Result contains higher order function than in optimal. Order ",
                       convert(ExpnType_result,string)," vs. order ",
                       convert(ExpnType_optimal,string),".");
       fi;

end proc:

#
# is_contains_complex(result)
# takes expressions and returns true if it contains "I" else false
#
#Nasser 032417
is_contains_complex:= proc(expression)
  return (has(expression,I));
end proc:
```

```
# The following summarizes the type number assigned an expression
# based on the functions it involves
# 1 = rational function
# 2 = algebraic function
# 3 = elementary function
# 4 = special function
# 5 = hyperpergeometric function
# 6 = appell function
# 7 = rootsum function
# 8 = integrate function
# 9 = unknown function

ExpnType := proc(expn)
  if type(expn,'atomic') then
    1
  elif type(expn,'list') then
    apply(max,map(ExpnType,expn))
  elif type(expn,'sqrt') then
    if type(op(1,expn),'rational') then
       1
    else
       max(2,ExpnType(op(1,expn)))
    end if
  elif type(expn,'`^`') then
    if type(op(2,expn),'integer') then
      ExpnType(op(1,expn))
    elif type(op(2,expn),'rational') then
      if type(op(1,expn),'rational') then
         1
      else
         max(2,ExpnType(op(1,expn)))
      end if
    else
         max(3,ExpnType(op(1,expn)),ExpnType(op(2,expn)))
    end if
  elif type(expn,'`+`') or type(expn,'`*`') then
    max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
  elif ElementaryFunctionQ(op(0,expn)) then
    max(3,ExpnType(op(1,expn)))
  elif SpecialFunctionQ(op(0,expn)) then
    max(4,apply(max,map(ExpnType,[op(expn)])))
  elif HypergeometricFunctionQ(op(0,expn)) then
    max(5,apply(max,map(ExpnType,[op(expn)])))
  elif AppellFunctionQ(op(0,expn)) then
    max(6,apply(max,map(ExpnType,[op(expn)])))
  elif op(0,expn)='int' then
    max(8,apply(max,map(ExpnType,[op(expn)]))) else
```

```
  9
  end if
end proc:


ElementaryFunctionQ := proc(func)
  member(func,[
        exp,log,ln,
        sin,cos,tan,cot,sec,csc,
        arcsin,arccos,arctan,arccot,arcsec,arccsc,
        sinh,cosh,tanh,coth,sech,csch,
        arcsinh,arccosh,arctanh,arccoth,arcsech,arccsch])
end proc:

SpecialFunctionQ := proc(func)
  member(func,[
        erf,erfc,erfi,
        FresnelS,FresnelC,
        Ei,Ei,Li,Si,Ci,Shi,Chi,
        GAMMA,lnGAMMA,Psi,Zeta,polylog,dilog,LambertW,
        EllipticF,EllipticE,EllipticPi])
end proc:

HypergeometricFunctionQ := proc(func)
  member(func,[Hypergeometric1F1,hypergeom,HypergeometricPFQ])
end proc:

AppellFunctionQ := proc(func)
  member(func,[AppellF1])
end proc:

# u is a sum or product.  rest(u) returns all but the
# first term or factor of u.
rest := proc(u) local v;
  if nops(u)=2 then
     op(2,u)
  else
     apply(op(0,u),op(2..nops(u),u))
  end if
end proc:

#leafcount(u) returns the number of nodes in u.
#Nasser 3/23/17  Replaced by build-in leafCount from package in Maple
leafcount := proc(u)
   MmaTranslator[Mma][LeafCount](u);
end proc:
```

Sympy grading function

```
#Dec 24, 2019. Nasser M. Abbasi:
#             Port of original Maple grading function by
#             Albert Rich to use with Sympy/Python
#Dec 27, 2019  Nasser. Added `RootSum`. See problem 177, Timofeev file
#             added 'exp_polar'
from sympy import *

def leaf_count(expr):
    #sympy do not have leaf count function. This is approximation
    return round(1.7*count_ops(expr))

def is_sqrt(expr):
    if isinstance(expr,Pow):
        if expr.args[1] == Rational(1,2):
            return True
        else:
            return False
    else:
        return False

def is_elementary_function(func):
    return func in [exp,log,ln,sin,cos,tan,cot,sec,csc,
            asin,acos,atan,acot,asec,acsc,sinh,cosh,tanh,coth,sech,csch,
            asinh,acosh,atanh,acoth,asech,acsch
        ]

def is_special_function(func):
    return func in [ erf,erfc,erfi,
             fresnels,fresnelc,Ei,Ei,Li,Si,Ci,Shi,Chi,
             gamma,loggamma,digamma,zeta,polylog,LambertW,
             elliptic_f,elliptic_e,elliptic_pi,exp_polar
         ]

def is_hypergeometric_function(func):
    return func in [hyper]

def is_appell_function(func):
    return func in [appellf1]

def is_atom(expn):
    try:
        if expn.isAtom or isinstance(expn,int) or isinstance(expn,float):
           return True
        else:
           return False
```

```
    except AttributeError as error:
        return False

def expnType(expn):
    debug=False
    if debug:
        print("expn=",expn,"type(expn)=",type(expn))

    if is_atom(expn):
      return 1
    elif isinstance(expn,list):
        return max(map(expnType, expn))    #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
            return 1
        else:
            return max(2,expnType(expn.args[0]))   #max(2,ExpnType(op(1,expn)))
    elif isinstance(expn,Pow):    #type(expn,'`^`')
        if isinstance(expn.args[1],Integer):   #type(op(2,expn),'integer')
            return expnType(expn.args[0])    #ExpnType(op(1,expn))
        elif isinstance(expn.args[1],Rational):   #type(op(2,expn),'rational')
            if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
                return 1
            else:
                return max(2,expnType(expn.args[0]))   #max(2,ExpnType(op(1,expn)))
        else:
            return max(3,expnType(expn.args[0]),expnType(expn.args[1])) #max(3,ExpnType(op(1,expn)),ExpnTy
    elif isinstance(expn,Add) or isinstance(expn,Mul): #type(expn,'`+`') or type(expn,'`*`')
        m1 = expnType(expn.args[0])
        m2 = expnType(list(expn.args[1:]))
        return max(m1,m2)   #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.func):   #ElementaryFunctionQ(op(0,expn))
        return max(3,expnType(expn.args[0]))   #max(3,ExpnType(op(1,expn)))
    elif is_special_function(expn.func): #SpecialFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(4,m1)    #max(4,apply(max,map(ExpnType,[op(expn)])))
    elif is_hypergeometric_function(expn.func): #HypergeometricFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(5,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif is_appell_function(expn.func):
        m1 = max(map(expnType, list(expn.args)))
        return max(6,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif isinstance(expn,RootSum):
        m1 = max(map(expnType, list(expn.args))) #Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
        return max(7,m1)
    elif str(expn).find("Integral") != -1:
```

```
        m1 = max(map(expnType, list(expn.args)))
        return max(8,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):

    #print ("Enter grade_antiderivative for sagemath")
    #print("Enter grade_antiderivative, result=",result," optimal=",optimal)

    leaf_count_result  = leaf_count(result)
    leaf_count_optimal = leaf_count(optimal)

    #print("leaf_count_result=",leaf_count_result)
    #print("leaf_count_optimal=",leaf_count_optimal)

    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if str(result).find("Integral") != -1:
        grade = "F"
        grade_annotation =""
    else:
        if expnType_result <= expnType_optimal:
            if result.has(I):
                if optimal.has(I): #both result and optimal complex
                    if leaf_count_result <= 2*leaf_count_optimal:
                        grade = "A"
                        grade_annotation =""
                    else:
                        grade = "B"
                        grade_annotation ="Both result and optimal contain complex but leaf count of result is larg
                else: #result contains complex but optimal is not
                    grade = "C"
                    grade_annotation ="Result contains complex when optimal does not."
            else: # result do not contain complex, this assumes optimal do not as well
                if leaf_count_result <= 2*leaf_count_optimal:
                    grade = "A"
                    grade_annotation =""
                else:
                    grade = "B"
                    grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(le
        else:
            grade = "C"
            grade_annotation ="Result contains higher order function than in optimal. Order "+str(ExpnType_
```

```
    #print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

    return grade, grade_annotation
```

SageMath grading function

```
#Dec 24, 2019. Nasser: Ported original Maple grading function by
#          Albert Rich to use with Sagemath. This is used to
#          grade Fricas, Giac and Maxima results.
#Dec 24, 2019. Nasser: Added 'exp_integral_e' and 'sng', 'sin_integral'
#          'arctan2','floor','abs','log_integral'
#June 4, 2022  Made default grade_annotation "none" instead of "" due
#          issue later when reading the file.
#July 14, 2022. Added ellipticF. This is until they fix sagemath, then remove it.

from sage.all import *
from sage.symbolic.operators import add_vararg, mul_vararg

debug=False;

def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    #print("Enter tree_size, expr is ",expr)

    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)

def is_sqrt(expr):
    if expr.operator() == operator.pow:   #isinstance(expr,Pow):
        if expr.operands()[1]==1/2: #expr.args[1] == Rational(1,2):
            if debug: print ("expr is sqrt")
            return True
        else:
            return False
    else:
        return False
```

```
def is_elementary_function(func):
    #debug=False
    m = func.name() in ['exp','log','ln',
            'sin','cos','tan','cot','sec','csc',
            'arcsin','arccos','arctan','arccot','arcsec','arccsc',
            'sinh','cosh','tanh','coth','sech','csch',
            'arcsinh','arccosh','arctanh','arccoth','arcsech','arccsch','sgn',
        'arctan2','floor','abs'
        ]
    if debug:
        if m:
            print ("func ", func , " is elementary_function")
        else:
            print ("func ", func , " is NOT elementary_function")


    return m

def is_special_function(func):
    #debug=False
    if debug:
        print ("type(func)=", type(func))

    m= func.name() in ['erf','erfc','erfi','fresnel_sin','fresnel_cos','Ei',
          'Ei','Li','Si','sin_integral','Ci','cos_integral','Shi','sinh_integral'
          'Chi','cosh_integral','gamma','log_gamma','psi,zeta',
          'polylog','lambert_w','elliptic_f','elliptic_e','ellipticF',
          'elliptic_pi','exp_integral_e','log_integral']

    if debug:
        print ("m=",m)
        if m:
            print ("func ", func ," is special_function")
        else:
            print ("func ", func ," is NOT special_function")


    return m


def is_hypergeometric_function(func):
    return func.name() in ['hypergeometric','hypergeometric_M','hypergeometric_U']

def is_appell_function(func):
    return func.name() in ['hypergeometric']   #[appellf1] can't find this in sagemath
```

```
def is_atom(expn):

   #debug=False
   if debug:
      print ("Enter is_atom, expn=",expn)

   if not hasattr(expn, 'parent'):
      return False


   #thanks to answer at https://ask.sagemath.org/question/49179/what-is-sagemath-equivalent-to-atomic-t
   try:
      if expn.parent() is SR:
         return expn.operator() is None
      if expn.parent() in (ZZ, QQ, AA, QQbar):
         return expn in expn.parent() # Should always return True
      if hasattr(expn.parent(),"base_ring") and hasattr(expn.parent(),"gens"):
         return expn in expn.parent().base_ring() or expn in expn.parent().gens()

      return False

   except AttributeError as error:
      print("Exception,AttributeError in  is_atom")
      print ("cought exception" , type(error).__name__ )
      return False


def expnType(expn):

   if debug:
      print (">>>>>Enter expnType, expn=", expn)
      print (">>>>>is_atom(expn)=", is_atom(expn))

   if is_atom(expn):
      return 1
   elif type(expn)==list:   #isinstance(expn,list):
      return max(map(expnType, expn))   #apply(max,map(ExpnType,expn))
   elif  is_sqrt(expn):
      if  type(expn.operands()[0])==Rational:  #type(isinstance(expn.args[0],Rational):
         return 1
      else:
         return max(2,expnType(expn.operands()[0]))   #max(2,expnType(expn.args[0]))
   elif expn.operator() == operator.pow:   #isinstance(expn,Pow)
      if type(expn.operands()[1])==Integer:   #isinstance(expn.args[1],Integer)
         return expnType(expn.operands()[0])   #expnType(expn.args[0])
      elif type(expn.operands()[1])==Rational:   #isinstance(expn.args[1],Rational)
         if type(expn.operands()[0])==Rational:  #isinstance(expn.args[0],Rational)
```

```
            return 1
         else:
            return max(2,expnType(expn.operands()[0]))  #max(2,expnType(expn.args[0]))
      else:
         return max(3,expnType(expn.operands()[0]),expnType(expn.operands()[1])) #max(3,expnType(expn.
   elif expn.operator() == add_vararg or expn.operator() == mul_vararg: #isinstance(expn,Add) or isinsta
      m1 = expnType(expn.operands()[0]) #expnType(expn.args[0])
      m2 = expnType(expn.operands()[1:]) #expnType(list(expn.args[1:]))
      return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
   elif is_elementary_function(expn.operator()):  #is_elementary_function(expn.func)
      return max(3,expnType(expn.operands()[0]))
   elif is_special_function(expn.operator()): #is_special_function(expn.func)
      m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(expn.args)))
      return max(4,m1)   #max(4,m1)
   elif is_hypergeometric_function(expn.operator()): #is_hypergeometric_function(expn.func)
      m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(expn.args)))
      return max(5,m1)   #max(5,m1)
   elif is_appell_function(expn.operator()):
      m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(expn.args)))
      return max(6,m1)   #max(6,m1)
   elif str(expn).find("Integral") != -1: #this will never happen, since it
            #is checked before calling the grading function that is passed.
            #but kept it here.
      m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(expn.args)))
      return max(8,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
   else:
      return 9

#main function
def grade_antiderivative(result,optimal):


   if debug:
      print ("Enter grade_antiderivative for sagemath")
      print("Enter grade_antiderivative, result=",result)
      print("Enter grade_antiderivative, optimal=",optimal)
      print("type(anti)=",type(result))
      print("type(optimal)=",type(optimal))

   leaf_count_result  = tree_size(result) #leaf_count(result)
   leaf_count_optimal = tree_size(optimal) #leaf_count(optimal)

   #if debug: print ("leaf_count_result=", leaf_count_result, "leaf_count_optimal=",leaf_count_optimal)


   expnType_result  = expnType(result)
   expnType_optimal = expnType(optimal)
```

```
if debug: print ("expnType_result=", expnType_result, "expnType_optimal=",expnType_optimal)

if expnType_result <= expnType_optimal:
    if result.has(I):
        if optimal.has(I): #both result and optimal complex
            if leaf_count_result <= 2*leaf_count_optimal:
                grade = "A"
                grade_annotation ="none"
            else:
                grade = "B"
                grade_annotation ="Both result and optimal contain complex but leaf count of result is larger th
        else: #result contains complex but optimal is not
            grade = "C"
            grade_annotation ="Result contains complex when optimal does not."
    else: # result do not contain complex, this assumes optimal do not as well
        if leaf_count_result <= 2*leaf_count_optimal:
            grade = "A"
            grade_annotation ="none"
        else:
            grade = "B"
            grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(leaf_c
else:
    grade = "C"
    grade_annotation ="Result contains higher order function than in optimal. Order "+str(expnType_result


print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

return grade, grade_annotation
```