Computer Algebra Independent Integration Tests

Summer 2023 edition

1-Algebraic-functions/1.2-Trinomial-products/1.2.3-General/49-1.2.3.5-P-x-d-x-^m-a+b-x^n+c-x^-2-n-^p

Nasser M. Abbasi

September 6, 2023 Compiled on September 6, 2023 at 2:26am

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 [17]. This is test number [49].

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 (17)	0.00 (0)
Mathematica	100.00 (17)	0.00 (0)
Fricas	41.18 (7)	58.82 (10)
Mupad	29.41 (5)	70.59 (12)
Giac	23.53 (4)	76.47 (13)
Maple	17.65 (3)	82.35 (14)
Maxima	11.76 (2)	88.24 (15)
Sympy	11.76 (2)	88.24 (15)

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.

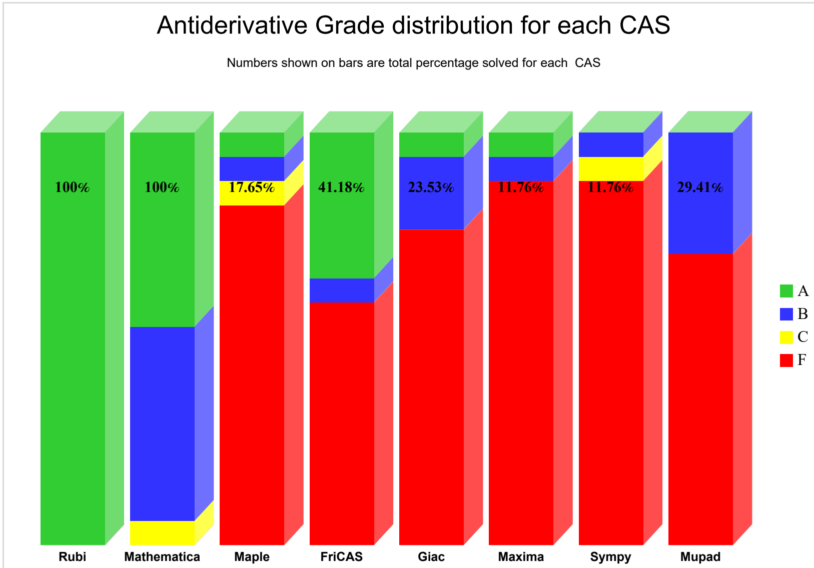
Table 1.2: Description of grading applied to integration result

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

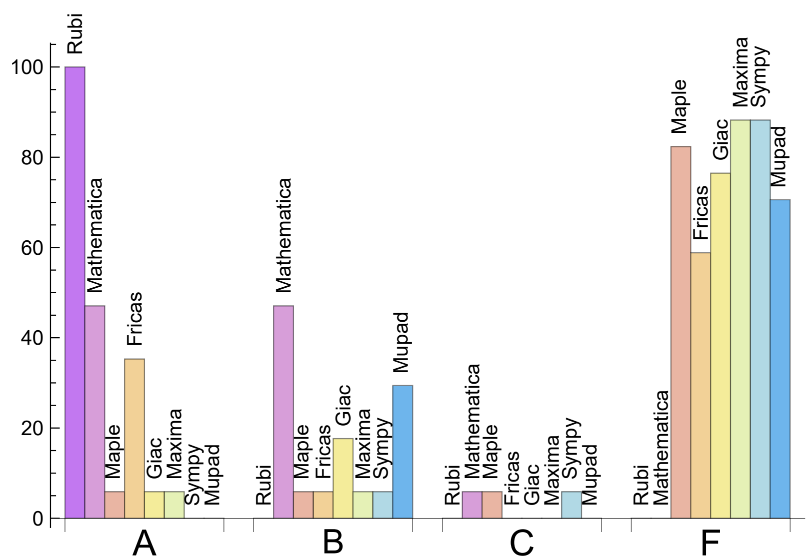
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.000	0.000	0.000	0.000
Mathematica	47.059	47.059	5.882	0.000
Fricas	35.294	5.882	0.000	58.824
Maple	5.882	5.882	5.882	82.353
Giac	5.882	17.647	0.000	76.471
Maxima	5.882	5.882	0.000	88.235
Mupad	0.000	29.412	0.000	70.588
Sympy	0.000	5.882	5.882	88.235

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
Fricas	10	90.00	10.00	0.00
Mupad	12	0.00	100.00	0.00
Giac	13	92.31	7.69	0.00
Maple	14	100.00	0.00	0.00
Maxima	15	100.00	0.00	0.00
Sympy	15	20.00	80.00	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	0.25
Fricas	0.26
Rubi	0.55
Giac	1.74
Mathematica	2.63
Mupad	17.59
Maple	45.52
Sympy	73.58

Table 1.5: Time performance for each CAS

System	Mean size	Normalized mean	Median size	Normalized median
Maxima	47.50	1.91	47.50	1.91
Maple	84.33	1.61	90.00	1.65
Fricas	85.00	1.54	69.00	1.45
Giac	97.00	2.49	81.00	2.90
Sympy	111.50	2.81	111.50	2.81
Rubi	457.12	1.00	263.00	1.00
Mathematica	1695.12	2.70	261.00	2.00
Mupad	71874.60	44.15	50.00	1.72

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.

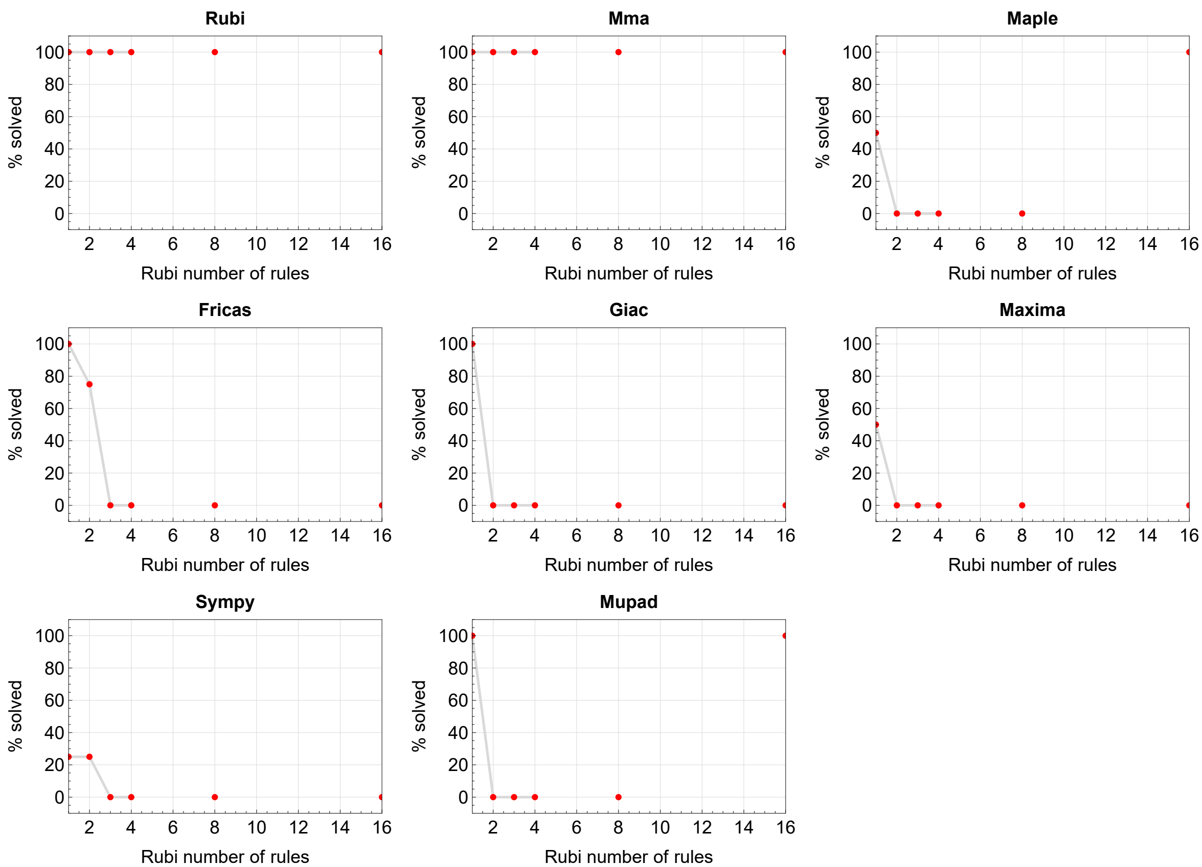


Figure 1.1: Solving statistics per number of Rubi rules used

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.

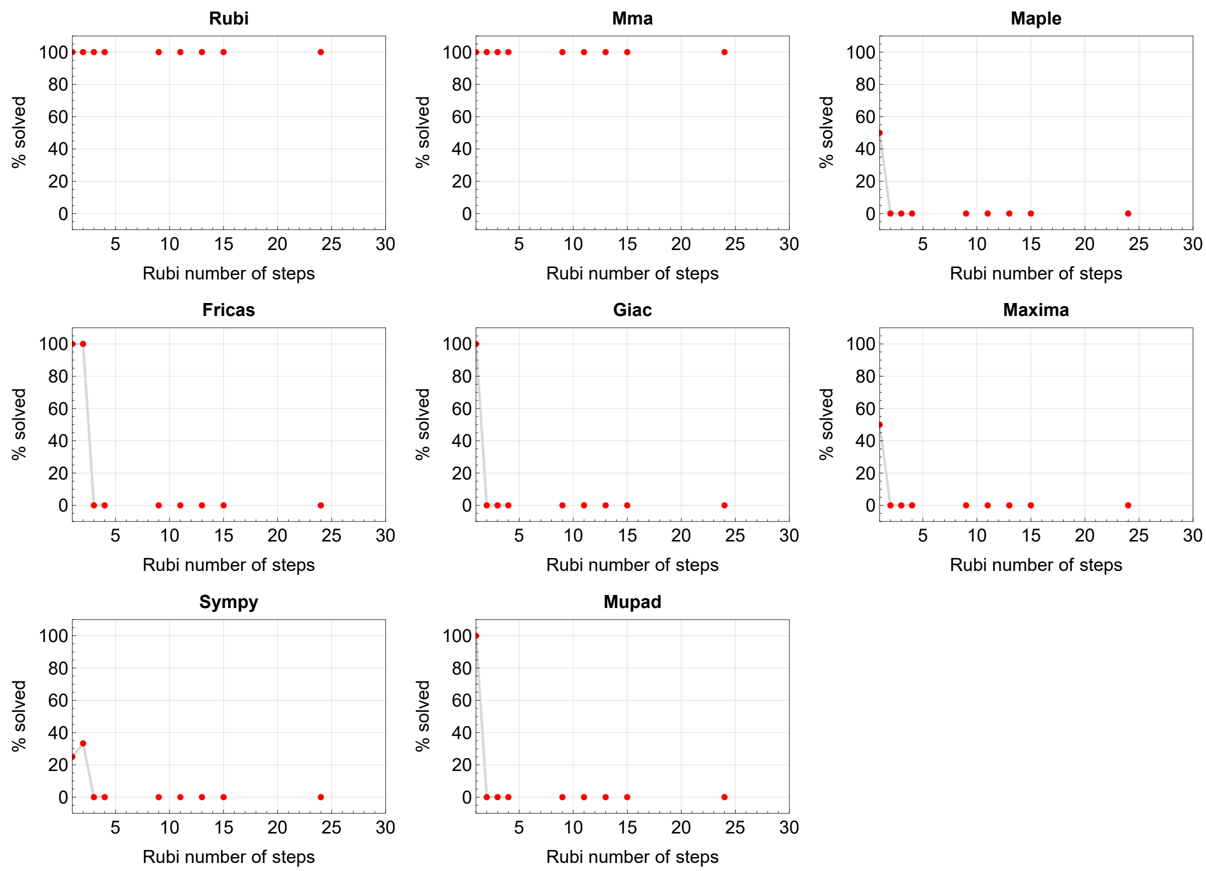


Figure 1.2: Solving statistics per number of Rubi steps used

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.

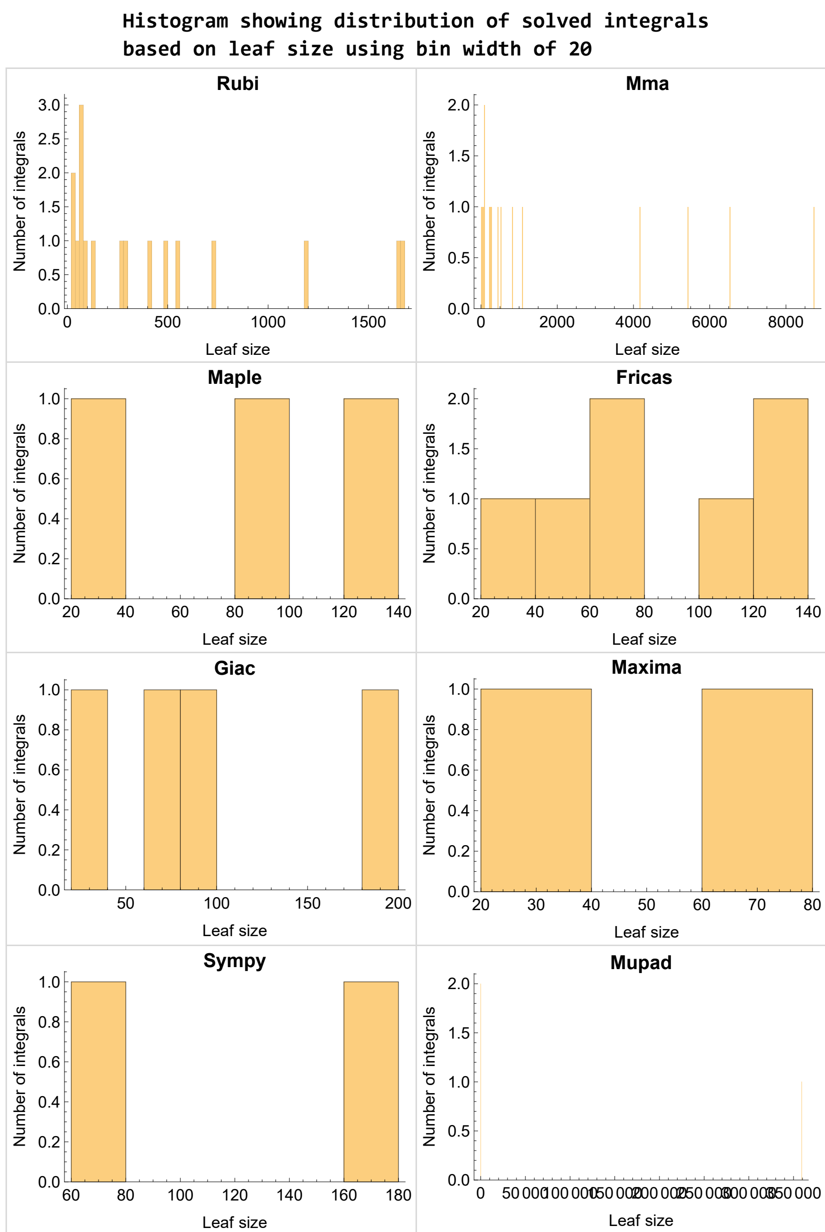


Figure 1.3: Solved integrals based on leaf size distribution

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.

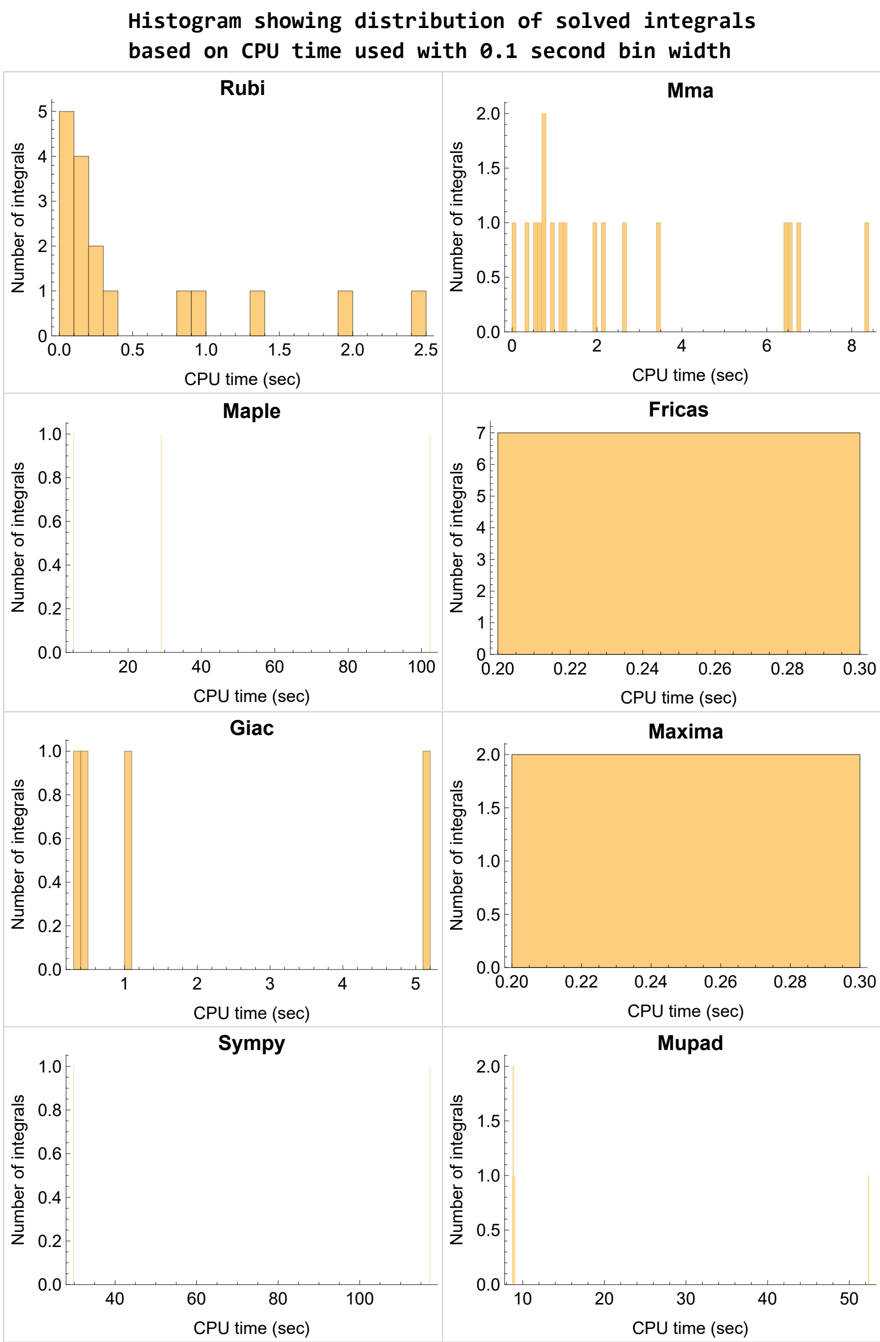


Figure 1.4: Solved integrals histogram based on CPU time used

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.

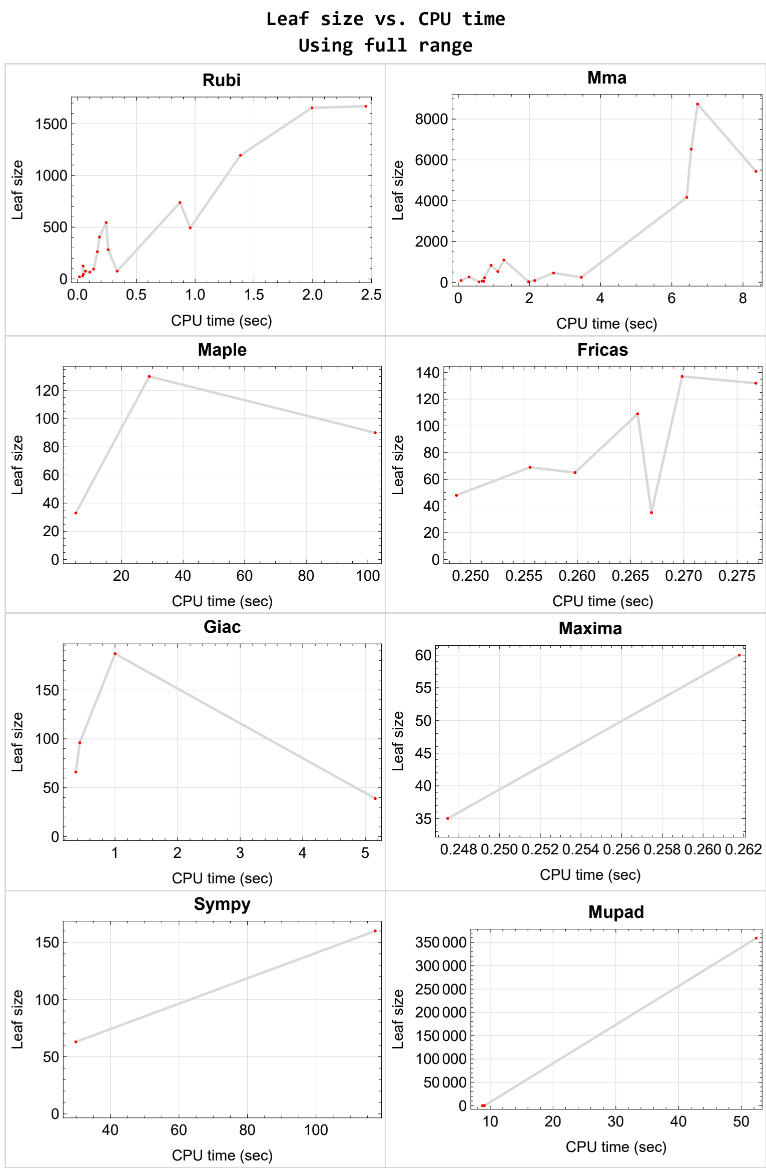


Figure 1.5: Leaf size vs. CPU time. Full range

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 $\{8, 9\}$

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.

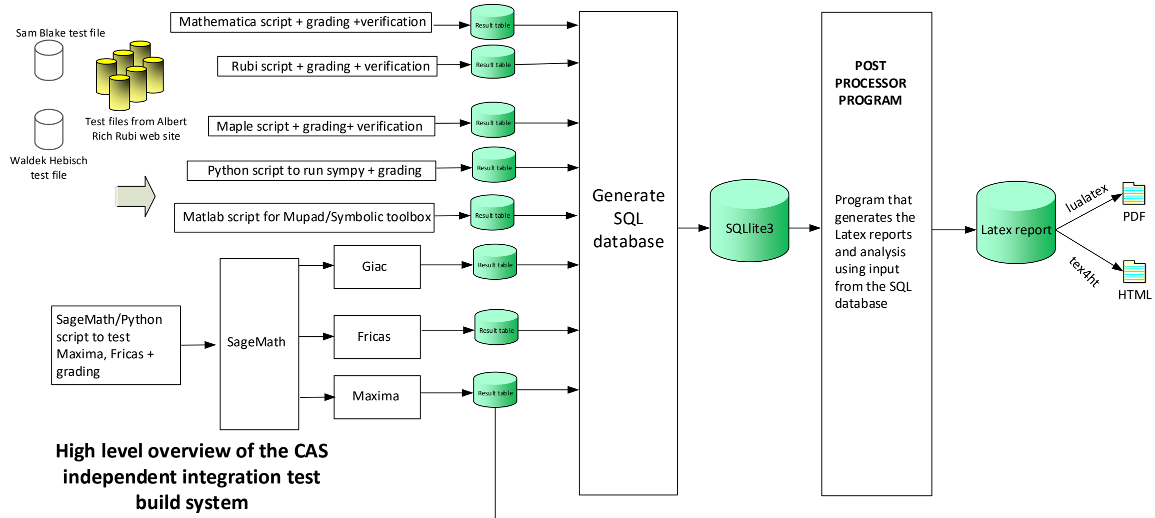
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

```
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.



One record (line) per one integral result. The line is CSV comma separated. This is description of each record

1. integer, the problem number.
2. integer. 0 for failed, 1 for passed, -1 for timeout, -2 for CAS specific exception. (this is not the grade field)
3. integer. Leaf size of result.
4. integer. Leaf size of the optimal antiderivative.
5. number. CPU time used to solve this integral. 0 if failed.
6. string. The integral in Latex format
7. string. The input used in CAS own syntax.
8. string. The result (antiderivative) produced by CAS in Latex format
9. string. The optimal antiderivative in Latex format.
10. integer. 0 or 1. Indicates if problem has known antiderivative or not
11. String. The result (antiderivative) in CAS own syntax.
12. String. The grade of the antiderivative. Can be "A", "B", "C", or "F"
13. String. Small string description of why the grade was given.
14. integer. 1 if result was verified or 0 if not verified. (For mma, rubi and maple only)

*The following fields are present only in **Rubi** Table file*

15. integer. Number of steps used.
16. integer. Number of rules used.
17. integer. Integrand leaf size.
18. real number. Ratio. Field 16 over field 17
19. String of form "{n,n,..}" which is list of the rules used by Rubi
20. String. The optimal antiderivative in Mathematica syntx

Nasser M. Abbasi
June 27, 2023
Design.vsdx

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 }

B grade { }

C grade { }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Mma

A grade { 3, 6, 10, 11, 12, 13, 14, 16 }

B grade { 2, 4, 5, 7, 8, 9, 15, 17 }

C grade { 1 }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Maple

A grade { 11 }

B grade { 16 }

C grade { 1 }

F normal fail { 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 17 }

F(-1) timedout fail { }

F(-2) exception fail { }

Fricas

A grade { 10, 11, 12, 13, 14, 15 }

B grade { 16 }

C grade { }

F normal fail { 2, 3, 4, 5, 6, 7, 8, 9, 17 }

F(-1) timedout fail { 1 }

F(-2) exception fail { }

Maxima

A grade { 11 }

B grade { 16 }

C grade { }

F normal fail { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 17 }

F(-1) timedout fail { }

F(-2) exception fail { }

Giac

A grade { 12 }

B grade { 11, 14, 16 }

C grade { }

F normal fail { 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 15, 17 }

F(-1) timedout fail { 1 }

F(-2) exception fail { }

Mupad

A grade { }

B grade { 1, 11, 12, 14, 16 }

C grade { }

F normal fail { }

F(-1) timedout fail { 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 15, 17 }

F(-2) exception fail { }

Sympy

A grade { }

B grade { 11 }

C grade { 13 }

F normal fail { 2, 3, 4 }

F(-1) timedout fail { 1, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17 }

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	C	C	F	F(-1)	F(-1)	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	1668	1668	223	130	0	0	0	0	359169
N.S.	1	1.00	0.13	0.08	0.00	0.00	0.00	0.00	215.33
time (sec)	N/A	2.452	0.739	29.016	0.000	0.000	0.000	0.000	52.377

Problem 2	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	124	124	261	0	0	0	0	0	0
N.S.	1	1.00	2.10	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.048	0.303	0.000	0.000	0.000	0.000	0.000	0.000

Problem 3	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	263	263	525	0	0	0	0	0	0
N.S.	1	1.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.169	1.110	0.000	0.000	0.000	0.000	0.000	0.000

Problem 4	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	404	404	834	0	0	0	0	0	0
N.S.	1	1.00	2.06	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.187	0.920	0.000	0.000	0.000	0.000	0.000	0.000

Problem 5	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	545	545	1093	0	0	0	0	0	0
N.S.	1	1.00	2.01	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.244	1.283	0.000	0.000	0.000	0.000	0.000	0.000

Problem 6	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	283	283	456	0	0	0	0	0	0
N.S.	1	1.00	1.61	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.261	2.677	0.000	0.000	0.000	0.000	0.000	0.000

Problem 7	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	738	738	4162	0	0	0	0	0	0
N.S.	1	1.00	5.64	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.871	6.421	0.000	0.000	0.000	0.000	0.000	0.000

Problem 8	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	No	N/A	TBD	TBD	TBD	TBD	TBD
size	1194	1194	6525	0	0	0	0	0	0
N.S.	1	1.00	5.46	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.385	6.546	0.000	0.000	0.000	0.000	0.000	0.000

Problem 9	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	No	N/A	TBD	TBD	TBD	TBD	TBD
size	1654	1654	8737	0	0	0	0	0	0
N.S.	1	1.00	5.28	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.994	6.725	0.000	0.000	0.000	0.000	0.000	0.000

Problem 10	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	A	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	75	75	84	0	0	137	0	0	0
N.S.	1	1.00	1.12	0.00	0.00	1.83	0.00	0.00	0.00
time (sec)	N/A	0.338	2.147	0.000	0.000	0.270	0.000	0.000	0.000

Problem 11	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	20	20	19	33	35	35	63	66	35
N.S.	1	1.00	0.95	1.65	1.75	1.75	3.15	3.30	1.75
time (sec)	N/A	0.016	0.584	5.155	0.247	0.267	29.898	0.374	8.880

Problem 12	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	A	F(-1)	A	B
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	45	45	45	0	0	48	0	39	39
N.S.	1	1.00	1.00	0.00	0.00	1.07	0.00	0.87	0.87
time (sec)	N/A	0.049	0.715	0.000	0.000	0.249	0.000	5.158	9.081

Problem 13	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	A	C	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	65	65	64	0	0	69	160	0	0
N.S.	1	1.00	0.98	0.00	0.00	1.06	2.46	0.00	0.00
time (sec)	N/A	0.104	0.673	0.000	0.000	0.256	117.269	0.000	0.000

Problem 14	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	A	F(-1)	B	B
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	75	75	84	0	0	109	0	187	80
N.S.	1	1.00	1.12	0.00	0.00	1.45	0.00	2.49	1.07
time (sec)	N/A	0.067	0.078	0.000	0.000	0.266	0.000	1.001	8.877

Problem 15	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	A	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	95	95	242	0	0	132	0	0	0
N.S.	1	1.00	2.55	0.00	0.00	1.39	0.00	0.00	0.00
time (sec)	N/A	0.138	3.459	0.000	0.000	0.277	0.000	0.000	0.000

Problem 16	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	B	B	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	29	29	24	90	60	65	0	96	50
N.S.	1	1.00	0.83	3.10	2.07	2.24	0.00	3.31	1.72
time (sec)	N/A	0.045	1.978	102.389	0.262	0.260	0.000	0.437	8.729

Problem 17	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F	F	F(-1)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	494	494	5439	0	0	0	0	0	0
N.S.	1	1.00	11.01	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.958	8.364	0.000	0.000	0.000	0.000	0.000	0.000

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 [7] had the largest ratio of [.3635999999999999979]

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	37	16	1.00	55	0.291
2	A	3	2	1.00	16	0.125
3	A	9	4	1.00	22	0.182
4	A	11	4	1.00	27	0.148
5	A	13	4	1.00	32	0.125
6	A	4	3	1.00	16	0.188
7	A	15	8	1.00	22	0.364
8	A	24	8	1.00	27	0.296
9	A	33	8	1.00	32	0.250
10	A	2	2	1.00	63	0.032
11	A	1	1	1.00	45	0.022
12	A	1	1	1.00	52	0.019
13	A	2	2	1.00	54	0.037
14	A	1	1	1.00	61	0.016
15	A	2	2	1.00	63	0.032
16	A	1	1	1.00	56	0.018
17	A	4	3	1.00	38	0.079

CHAPTER 3

LISTING OF INTEGRALS

3.1 $\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx$

Optimal result

Integrand size = 55, antiderivative size = 1668

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx$$

$$= \frac{kx}{c}+\frac{lx^2}{2c}+\frac{mx^3}{3c}-\frac{\left(g-\frac{bk}{c}+\frac{2c^2d+b^2k-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b-\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{\sqrt[3]{2}\sqrt{3}\sqrt[3]{c}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}+\frac{2c^2e+b^2l-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b-\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{2^{2/3}\sqrt{3}c^{2/3}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$-\frac{\left(g-\frac{bk}{c}-\frac{2c^2d-bcg+b^2k-2ack}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b+\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{\sqrt[3]{2}\sqrt{3}\sqrt[3]{c}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}-\frac{2c^2e-bch+b^2l-2acl}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b+\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{2^{2/3}\sqrt{3}c^{2/3}\sqrt[3]{b+\sqrt{b^2-4ac}}}$$

$$-\frac{(2c^2f-bcj+b^2m-2acm)\,\text{arctanh}\left(\frac{b+2cx^3}{\sqrt{b^2-4ac}}\right)}{3c^2\sqrt{b^2-4ac}}$$

$$+\frac{\left(g-\frac{bk}{c}+\frac{2c^2d+b^2k-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{b-\sqrt{b^2-4ac}}+\sqrt[3]{2}\sqrt[3]{c}x\right)}{3\sqrt[3]{2}\sqrt[3]{c}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}+\frac{2c^2e+b^2l-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{b-\sqrt{b^2-4ac}}+\sqrt[3]{2}\sqrt[3]{c}x\right)}{3\ 2^{2/3}c^{2/3}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(g-\frac{bk}{c}-\frac{2c^2d-bcg+b^2k-2ack}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{b+\sqrt{b^2-4ac}}+\sqrt[3]{2}\sqrt[3]{c}x\right)}{3\sqrt[3]{2}\sqrt[3]{c}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\left(h-\frac{bl}{c}-\frac{2c^2e-bch+b^2l-2acl}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{b+\sqrt{b^2-4ac}}+\sqrt[3]{2}\sqrt[3]{c}x\right)$$

```
[Out] k*x/c+1/2*l*x^2/c+1/3*m*x^3/c+1/6*(-b*m+c*j)*ln(c*x^6+b*x^3+a)/c^2+1/6*ln(2
^(1/3)*c^(1/3)*x+(b-(-4*a*c+b^2)^(1/2))^(1/3))*(g-b*k/c+(2*c^2*d+b^2*k-c*(2
*a*k+b*g))/c/(-4*a*c+b^2)^(1/2))*2^(2/3)/c^(1/3)/(b-(-4*a*c+b^2)^(1/2))^(2/
3)-1/12*ln(2^(2/3)*c^(2/3)*x^2-2^(1/3)*c^(1/3)*x*(b-(-4*a*c+b^2)^(1/2))^(1/
3)+(b-(-4*a*c+b^2)^(1/2))^(2/3))*(g-b*k/c+(2*c^2*d+b^2*k-c*(2*a*k+b*g))/c/(
-4*a*c+b^2)^(1/2))*2^(2/3)/c^(1/3)/(b-(-4*a*c+b^2)^(1/2))^(2/3)-1/6*arctan(
1/3*(1-2*2^(1/3)*c^(1/3)*x/(b-(-4*a*c+b^2)^(1/2))^(1/3))*3^(1/2))*(g-b*k/c+
(2*c^2*d+b^2*k-c*(2*a*k+b*g))/c/(-4*a*c+b^2)^(1/2))*2^(2/3)/c^(1/3)*3^(1/2)
/(b-(-4*a*c+b^2)^(1/2))^(2/3)-1/6*ln(2^(1/3)*c^(1/3)*x+(b-(-4*a*c+b^2)^(1/2
))^(1/3))*(h-b*l/c+(2*c^2*e+b^2*l-c*(2*a*l+b*h))/c/(-4*a*c+b^2)^(1/2))*2^(1
/3)/c^(2/3)/(b-(-4*a*c+b^2)^(1/2))^(1/3)+1/12*ln(2^(2/3)*c^(2/3)*x^2-2^(1/3
)*c^(1/3)*x*(b-(-4*a*c+b^2)^(1/2))^(1/3)+(b-(-4*a*c+b^2)^(1/2))^(2/3))*(h-b
*l/c+(2*c^2*e+b^2*l-c*(2*a*l+b*h))/c/(-4*a*c+b^2)^(1/2))*2^(1/3)/c^(2/3)/(b
-(-4*a*c+b^2)^(1/2))^(1/3)-1/6*arctan(1/3*(1-2*2^(1/3)*c^(1/3)*x/(b-(-4*a*c
+b^2)^(1/2))^(1/3))*3^(1/2))*(h-b*l/c+(2*c^2*e+b^2*l-c*(2*a*l+b*h))/c/(-4*a
*c+b^2)^(1/2))*2^(1/3)/c^(2/3)*3^(1/2)/(b-(-4*a*c+b^2)^(1/2))^(1/3)-1/3*(-2
*a*c*m+b^2*m-b*c*j+2*c^2*f)*arctanh((2*c*x^3+b)/(-4*a*c+b^2)^(1/2))/c^2/(-4
*a*c+b^2)^(1/2)+1/6*ln(2^(1/3)*c^(1/3)*x+(b+(-4*a*c+b^2)^(1/2))^(1/3))*(g-b
*k/c+(2*a*c*k-b^2*k+b*c*g-2*c^2*d)/c/(-4*a*c+b^2)^(1/2))*2^(2/3)/c^(1/3)/(b
+(-4*a*c+b^2)^(1/2))^(2/3)-1/12*ln(2^(2/3)*c^(2/3)*x^2-2^(1/3)*c^(1/3)*x*(b
+(-4*a*c+b^2)^(1/2))^(1/3)+(b+(-4*a*c+b^2)^(1/2))^(2/3))*(g-b*k/c+(2*a*c*k-
b^2*k+b*c*g-2*c^2*d)/c/(-4*a*c+b^2)^(1/2))*2^(2/3)/c^(1/3)/(b+(-4*a*c+b^2)^
(1/2))^(2/3)-1/6*arctan(1/3*(1-2*2^(1/3)*c^(1/3)*x/(b+(-4*a*c+b^2)^(1/2))^(
1/3))*3^(1/2))*(g-b*k/c+(2*a*c*k-b^2*k+b*c*g-2*c^2*d)/c/(-4*a*c+b^2)^(1/2))
*2^(2/3)/c^(1/3)*3^(1/2)/(b+(-4*a*c+b^2)^(1/2))^(2/3)-1/6*ln(2^(1/3)*c^(1/3
)*x+(b+(-4*a*c+b^2)^(1/2))^(1/3))*(h-b*l/c+(2*a*c*l-b^2*l+b*c*h-2*c^2*e)/c/
(-4*a*c+b^2)^(1/2))*2^(1/3)/c^(2/3)/(b+(-4*a*c+b^2)^(1/2))^(1/3)+1/12*ln(2^
(2/3)*c^(2/3)*x^2-2^(1/3)*c^(1/3)*x*(b+(-4*a*c+b^2)^(1/2))^(1/3)+(b+(-4*a*c
+b^2)^(1/2))^(2/3))*(h-b*l/c+(2*a*c*l-b^2*l+b*c*h-2*c^2*e)/c/(-4*a*c+b^2)^(
1/2))*2^(1/3)/c^(2/3)/(b+(-4*a*c+b^2)^(1/2))^(1/3)-1/6*arctan(1/3*(1-2*2^(1
/3)*c^(1/3)*x/(b+(-4*a*c+b^2)^(1/2))^(1/3))*3^(1/2))*(h-b*l/c+(2*a*c*l-b^2*
l+b*c*h-2*c^2*e)/c/(-4*a*c+b^2)^(1/2))*2^(1/3)/c^(2/3)*3^(1/2)/(b+(-4*a*c+b
^2)^(1/2))^(1/3)
```

Rubi [A] (verified)

Time = 2.45 (sec) , antiderivative size = 1668, normalized size of antiderivative = 1.00, number of steps used = 37, number of rules used = 16, $\frac{\text{number of rules}}{\text{integrand size}} = 0.291$, Rules used = {1804, 1803, 1436, 206, 31, 648, 631, 210, 642, 1772, 1524, 298, 1759, 1671, 632, 212}

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx$$

$$= \frac{mx^3}{3c}+\frac{lx^2}{2c}+\frac{kx}{c}-\frac{\left(g-\frac{bk}{c}+\frac{kb^2+2c^2d-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b-\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{\sqrt[3]{2}\sqrt{3}\sqrt[3]{c}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}+\frac{lb^2+2c^2e-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b-\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{2^{2/3}\sqrt{3}c^{2/3}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$-\frac{\left(g-\frac{bk}{c}-\frac{kb^2-cgb+2c^2d-2ack}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b+\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{\sqrt[3]{2}\sqrt{3}\sqrt[3]{c}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}-\frac{lb^2-chb+2c^2e-2acl}{c\sqrt{b^2-4ac}}\right)\arctan\left(\frac{1-\frac{2\sqrt[3]{2}\sqrt[3]{c}x}{\sqrt[3]{b+\sqrt{b^2-4ac}}}}{\sqrt{3}}\right)}{2^{2/3}\sqrt{3}c^{2/3}\sqrt[3]{b+\sqrt{b^2-4ac}}}$$

$$-\frac{(mb^2-cjb+2c^2f-2acm)\,\mathrm{arctanh}\left(\frac{2cx^3+b}{\sqrt{b^2-4ac}}\right)}{3c^2\sqrt{b^2-4ac}}$$

$$+\frac{\left(g-\frac{bk}{c}+\frac{kb^2+2c^2d-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{2}\sqrt[3]{c}x+\sqrt[3]{b-\sqrt{b^2-4ac}}\right)}{3\sqrt[3]{2}\sqrt[3]{c}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}+\frac{lb^2+2c^2e-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{2}\sqrt[3]{c}x+\sqrt[3]{b-\sqrt{b^2-4ac}}\right)}{3\ 2^{2/3}c^{2/3}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$+\frac{\left(g-\frac{bk}{c}-\frac{kb^2-cgb+2c^2d-2ack}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{2}\sqrt[3]{c}x+\sqrt[3]{b+\sqrt{b^2-4ac}}\right)}{3\sqrt[3]{2}\sqrt[3]{c}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$-\frac{\left(h-\frac{bl}{c}-\frac{lb^2-chb+2c^2e-2acl}{c\sqrt{b^2-4ac}}\right)\log\left(\sqrt[3]{2}\sqrt[3]{c}x+\sqrt[3]{b+\sqrt{b^2-4ac}}\right)}{3\ 2^{2/3}c^{2/3}\sqrt[3]{b+\sqrt{b^2-4ac}}}$$

```
[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^3 + c*x^6),x]
```

```
[Out] (k*x)/c + (l*x^2)/(2*c) + (m*x^3)/(3*c) - ((g - (b*k)/c + (2*c^2*d + b^2*k
- c*(b*g + 2*a*k))/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(1 - (2*2^(1/3)*c^(1/3)*x)
/(b - Sqrt[b^2 - 4*a*c])^(1/3))/Sqrt[3]])/(2^(1/3)*Sqrt[3]*c^(1/3)*(b - Sqr
t[b^2 - 4*a*c])^(2/3)) - ((h - (b*l)/c + (2*c^2*e + b^2*l - c*(b*h + 2*a*l)
)/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(1 - (2*2^(1/3)*c^(1/3)*x)/(b - Sqrt[b^2 -
4*a*c])^(1/3))/Sqrt[3]])/(2^(2/3)*Sqrt[3]*c^(2/3)*(b - Sqrt[b^2 - 4*a*c])^(
1/3)) - ((g - (b*k)/c - (2*c^2*d - b*c*g + b^2*k - 2*a*c*k)/(c*Sqrt[b^2 - 4
*a*c]))*ArcTan[(1 - (2*2^(1/3)*c^(1/3)*x)/(b + Sqrt[b^2 - 4*a*c])^(1/3))/Sq
rt[3]])/(2^(1/3)*Sqrt[3]*c^(1/3)*(b + Sqrt[b^2 - 4*a*c])^(2/3)) - ((h - (b*
l)/c - (2*c^2*e - b*c*h + b^2*l - 2*a*c*l)/(c*Sqrt[b^2 - 4*a*c]))*ArcTan[(1
 - (2*2^(1/3)*c^(1/3)*x)/(b + Sqrt[b^2 - 4*a*c])^(1/3))/Sqrt[3]])/(2^(2/3)*
Sqrt[3]*c^(2/3)*(b + Sqrt[b^2 - 4*a*c])^(1/3)) - ((2*c^2*f - b*c*j + b^2*m
- 2*a*c*m)*ArcTanh[(b + 2*c*x^3)/Sqrt[b^2 - 4*a*c]])/(3*c^2*Sqrt[b^2 - 4*a*
c]) + ((g - (b*k)/c + (2*c^2*d + b^2*k - c*(b*g + 2*a*k))/(c*Sqrt[b^2 - 4*a
*c]))*Log[(b - Sqrt[b^2 - 4*a*c])^(1/3) + 2^(1/3)*c^(1/3)*x])/(3*2^(1/3)*c^
(1/3)*(b - Sqrt[b^2 - 4*a*c])^(2/3)) - ((h - (b*l)/c + (2*c^2*e + b^2*l - c
*(b*h + 2*a*l))/(c*Sqrt[b^2 - 4*a*c]))*Log[(b - Sqrt[b^2 - 4*a*c])^(1/3) +
2^(1/3)*c^(1/3)*x])/(3*2^(2/3)*c^(2/3)*(b - Sqrt[b^2 - 4*a*c])^(1/3)) + ((g
 - (b*k)/c - (2*c^2*d - b*c*g + b^2*k - 2*a*c*k)/(c*Sqrt[b^2 - 4*a*c]))*Log
[(b + Sqrt[b^2 - 4*a*c])^(1/3) + 2^(1/3)*c^(1/3)*x])/(3*2^(1/3)*c^(1/3)*(b
+ Sqrt[b^2 - 4*a*c])^(2/3)) - ((h - (b*l)/c - (2*c^2*e - b*c*h + b^2*l - 2*
a*c*l)/(c*Sqrt[b^2 - 4*a*c]))*Log[(b + Sqrt[b^2 - 4*a*c])^(1/3) + 2^(1/3)*c
^(1/3)*x])/(3*2^(2/3)*c^(2/3)*(b + Sqrt[b^2 - 4*a*c])^(1/3)) - ((g - (b*k)/
c + (2*c^2*d + b^2*k - c*(b*g + 2*a*k))/(c*Sqrt[b^2 - 4*a*c]))*Log[(b - Sqr
t[b^2 - 4*a*c])^(2/3) - 2^(1/3)*c^(1/3)*(b - Sqrt[b^2 - 4*a*c])^(1/3)*x + 2
^(2/3)*c^(2/3)*x^2])/(6*2^(1/3)*c^(1/3)*(b - Sqrt[b^2 - 4*a*c])^(2/3)) + ((
h - (b*l)/c + (2*c^2*e + b^2*l - c*(b*h + 2*a*l))/(c*Sqrt[b^2 - 4*a*c]))*Lo
g[(b - Sqrt[b^2 - 4*a*c])^(2/3) - 2^(1/3)*c^(1/3)*(b - Sqrt[b^2 - 4*a*c])^(
1/3)*x + 2^(2/3)*c^(2/3)*x^2])/(6*2^(2/3)*c^(2/3)*(b - Sqrt[b^2 - 4*a*c])^(
1/3)) - ((g - (b*k)/c - (2*c^2*d - b*c*g + b^2*k - 2*a*c*k)/(c*Sqrt[b^2 - 4
*a*c]))*Log[(b + Sqrt[b^2 - 4*a*c])^(2/3) - 2^(1/3)*c^(1/3)*(b + Sqrt[b^2 -
 4*a*c])^(1/3)*x + 2^(2/3)*c^(2/3)*x^2])/(6*2^(1/3)*c^(1/3)*(b + Sqrt[b^2 -
 4*a*c])^(2/3)) + ((h - (b*l)/c - (2*c^2*e - b*c*h + b^2*l - 2*a*c*l)/(c*Sq
rt[b^2 - 4*a*c]))*Log[(b + Sqrt[b^2 - 4*a*c])^(2/3) - 2^(1/3)*c^(1/3)*(b +
Sqrt[b^2 - 4*a*c])^(1/3)*x + 2^(2/3)*c^(2/3)*x^2])/(6*2^(2/3)*c^(2/3)*(b +
Sqrt[b^2 - 4*a*c])^(1/3)) + ((c*j - b*m)*Log[a + b*x^3 + c*x^6])/(6*c^2)
```

Rule 31

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

Rule 206

```
Int[((a_) + (b_.)*(x_)^3)^(-1), x_Symbol] :> Dist[1/(3*Rt[a, 3]^2), Int[1/(
Rt[a, 3] + Rt[b, 3]*x), x], x] + Dist[1/(3*Rt[a, 3]^2), Int[(2*Rt[a, 3] - R
t[b, 3]*x)/(Rt[a, 3]^2 - Rt[a, 3]*Rt[b, 3]*x + Rt[b, 3]^2*x^2), x], x] /; F
reeQ[{a, 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 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 298

```
Int[(x_)/((a_) + (b_.)*(x_)^3), x_Symbol] :> Dist[-(3*Rt[a, 3]*Rt[b, 3])^(-
1), Int[1/(Rt[a, 3] + Rt[b, 3]*x), x], x] + Dist[1/(3*Rt[a, 3]*Rt[b, 3]), I
nt[(Rt[a, 3] + Rt[b, 3]*x)/(Rt[a, 3]^2 - Rt[a, 3]*Rt[b, 3]*x + Rt[b, 3]^2*x
^2), x], x] /; FreeQ[{a, b}, x]
```

Rule 631

```
Int[((a_) + (b_.)*(x_) + (c_.)*(x_)^2)^(-1), x_Symbol] :> With[{q = 1 - 4*S
implify[a*(c/b^2)]}, Dist[-2/b, Subst[Int[1/(q - x^2), x], x, 1 + 2*c*(x/b)
], x] /; RationalQ[q] && (EqQ[q^2, 1] ||  !RationalQ[b^2 - 4*a*c])] /; Free
Q[{a, b, c}, x] && NeQ[b^2 - 4*a*c, 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 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
*c] ||  !IGtQ[n/2, 0])
```

Rule 1524

```
Int[(((f_.)*(x_))^(m_.)*((d_) + (e_.)*(x_)^(n_)))/((a_) + (b_.)*(x_)^(n_) +
 (c_.)*(x_)^(n2_)), x_Symbol] :> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[e/2 +
(2*c*d - b*e)/(2*q), Int[(f*x)^m/(b/2 - q/2 + c*x^n), x], x] + Dist[e/2 - (
2*c*d - b*e)/(2*q), Int[(f*x)^m/(b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b
, c, d, e, f, m}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0] && IGtQ[n, 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]
```

Rule 1759

```
Int[(Pq_)*(x_)^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_.), x_
Symbol] :> Dist[1/n, Subst[Int[SubstFor[x^n, Pq, x]*(a + b*x + c*x^2)^p, x]
, x, x^n], x] /; FreeQ[{a, b, c, m, n, p}, x] && EqQ[n2, 2*n] && PolyQ[Pq,
x^n] && EqQ[Simplify[m - n + 1], 0]
```

Rule 1772

```
Int[(Pq_)*((d_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_))^(
p_), x_Symbol] :> With[{q = Expon[Pq, x]}, With[{Pqq = Coeff[Pq, x, q]}, In
t[(d*x)^m*ExpandToSum[Pq - Pqq*x^q - Pqq*((a*(m + q - 2*n + 1)*x^(q - 2*n)
+ b*(m + q + n*(p - 1) + 1)*x^(q - n))/(c*(m + q + 2*n*p + 1))), x]*(a + b*
x^n + c*x^(2*n))^p, x] + Simp[Pqq*(d*x)^(m + q - 2*n + 1)*((a + b*x^n + c*x
^(2*n))^(p + 1)/(c*d^(q - 2*n + 1)*(m + q + 2*n*p + 1))), x]] /; GeQ[q, 2*n
] && NeQ[m + q + 2*n*p + 1, 0] && (IntegerQ[2*p] || (EqQ[n, 1] && IntegerQ[
4*p]) || IntegerQ[p + (q + 1)/(2*n)])] /; FreeQ[{a, b, c, d, m, p}, x] && E
qQ[n2, 2*n] && PolyQ[Pq, x^n] && NeQ[b^2 - 4*a*c, 0] && IGtQ[n, 0]
```

Rule 1803

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_))^(p_), x_Symbol] :> W
ith[{q = Expon[Pq, x]}, With[{Pqq = Coeff[Pq, x, q]}, Int[ExpandToSum[Pq -
Pqq*x^q - Pqq*((a*(q - 2*n + 1)*x^(q - 2*n) + b*(q + n*(p - 1) + 1)*x^(q -
n))/(c*(q + 2*n*p + 1))), x]*(a + b*x^n + c*x^(2*n))^p, x] + Simp[Pqq*x^(q
- 2*n + 1)*((a + b*x^n + c*x^(2*n))^(p + 1)/(c*(q + 2*n*p + 1))), x]] /; Ge
Q[q, 2*n] && NeQ[q + 2*n*p + 1, 0] && (IntegerQ[2*p] || (EqQ[n, 1] && Integ
erQ[4*p]) || IntegerQ[p + (q + 1)/(2*n)])] /; FreeQ[{a, b, c, p}, x] && EqQ
[n2, 2*n] && PolyQ[Pq, x^n] && NeQ[b^2 - 4*a*c, 0] && IGtQ[n, 0]
```

Rule 1804

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_))^(p_), x_Symbol] :> Mo
dule[{q = Expon[Pq, x], j, k}, Int[Sum[x^j*Sum[Coeff[Pq, x, j + k*n]*x^(k*n
), {k, 0, (q - j)/n + 1}]*(a + b*x^n + c*x^(2*n))^p, {j, 0, n - 1}], x]] /;
 FreeQ[{a, b, c, p}, x] && EqQ[n2, 2*n] && PolyQ[Pq, x] && NeQ[b^2 - 4*a*c,
 0] && IGtQ[n, 0] &&  !PolyQ[Pq, x^n]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{d+gx^3+kx^6}{a+bx^3+cx^6}+\frac{x(e+hx^3+lx^6)}{a+bx^3+cx^6}+\frac{x^2(f+jx^3+mx^6)}{a+bx^3+cx^6}\right) dx \\
&= \int \frac{d+gx^3+kx^6}{a+bx^3+cx^6}\,dx+\int \frac{x(e+hx^3+lx^6)}{a+bx^3+cx^6}\,dx+\int \frac{x^2(f+jx^3+mx^6)}{a+bx^3+cx^6}\,dx \\
&= \frac{kx}{c}+\frac{lx^2}{2c}+\frac{1}{3}\text{Subst}\left(\int \frac{f+jx+mx^2}{a+bx+cx^2}\,dx,x,x^3\right) \\
&\quad +\int \frac{d-\frac{ak}{c}+\left(g-\frac{bk}{c}\right)x^3}{a+bx^3+cx^6}\,dx+\int \frac{x\left(e-\frac{al}{c}+\left(h-\frac{bl}{c}\right)x^3\right)}{a+bx^3+cx^6}\,dx \\
&= \frac{kx}{c}+\frac{lx^2}{2c}+\frac{1}{3}\text{Subst}\left(\int \left(\frac{m}{c}+\frac{cf-am+(cj-bm)x}{c\,(a+bx+cx^2)}\right) dx,x,x^3\right) \\
&\quad +\frac{1}{2}\left(g-\frac{bk}{c}-\frac{2c^2d-bcg+b^2k-2ack}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^3}\,dx \\
&\quad +\frac{1}{2}\left(g-\frac{bk}{c}+\frac{2c^2d+b^2k-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^3}\,dx \\
&\quad +\frac{1}{2}\left(h-\frac{bl}{c}-\frac{2c^2e-bch+b^2l-2acl}{c\sqrt{b^2-4ac}}\right)\int \frac{x}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^3}\,dx \\
&\quad +\frac{1}{2}\left(h-\frac{bl}{c}+\frac{2c^2e+b^2l-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\int \frac{x}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^3}\,dx
\end{aligned}$$

$$= \frac{kx}{c} + \frac{lx^2}{2c} + \frac{mx^3}{3c} + \frac{\text{Subst}\left(\int \frac{cf-am+(cj-bm)x}{a+bx+cx^2}\,dx, x, x^3\right)}{3c}$$

$$+ \frac{\left(g - \frac{bk}{c} - \frac{2c^2d-bcg+b^2k-2ack}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{\sqrt[3]{b+\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}\,dx}{3\sqrt[3]{2}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$+ \frac{\left(g - \frac{bk}{c} - \frac{2c^2d-bcg+b^2k-2ack}{c\sqrt{b^2-4ac}}\right)\int \frac{2^{2/3}\sqrt[3]{b+\sqrt{b^2-4ac}}-\sqrt[3]{c}x}{\frac{\left(b+\sqrt{b^2-4ac}\right)^{2/3}}{2^{2/3}}-\frac{\sqrt[3]{c}\sqrt[3]{b+\sqrt{b^2-4ac}}x}{\sqrt[3]{2}}+c^{2/3}x^2}\,dx}{3\sqrt[3]{2}\left(b+\sqrt{b^2-4ac}\right)^{2/3}}$$

$$+ \frac{\left(g - \frac{bk}{c} + \frac{2c^2d+b^2k-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{\sqrt[3]{b-\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}\,dx}{3\sqrt[3]{2}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$+ \frac{\left(g - \frac{bk}{c} + \frac{2c^2d+b^2k-c(bg+2ak)}{c\sqrt{b^2-4ac}}\right)\int \frac{2^{2/3}\sqrt[3]{b-\sqrt{b^2-4ac}}-\sqrt[3]{c}x}{\frac{\left(b-\sqrt{b^2-4ac}\right)^{2/3}}{2^{2/3}}-\frac{\sqrt[3]{c}\sqrt[3]{b-\sqrt{b^2-4ac}}x}{\sqrt[3]{2}}+c^{2/3}x^2}\,dx}{3\sqrt[3]{2}\left(b-\sqrt{b^2-4ac}\right)^{2/3}}$$

$$+ \frac{\left(h - \frac{bl}{c} - \frac{2c^2e-bch+b^2l-2acl}{c\sqrt{b^2-4ac}}\right)\int \frac{\frac{\sqrt[3]{b+\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}{\frac{\left(b+\sqrt{b^2-4ac}\right)^{2/3}}{2^{2/3}}-\frac{\sqrt[3]{c}\sqrt[3]{b+\sqrt{b^2-4ac}}x}{\sqrt[3]{2}}+c^{2/3}x^2}\,dx}{3\ 2^{2/3}\sqrt[3]{c}\sqrt[3]{b+\sqrt{b^2-4ac}}}$$

$$+ \frac{\left(-h + \frac{bl}{c} + \frac{2c^2e-bch+b^2l-2acl}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{\sqrt[3]{b+\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}\,dx}{3\ 2^{2/3}\sqrt[3]{c}\sqrt[3]{b+\sqrt{b^2-4ac}}}$$

$$+ \frac{\left(-h + \frac{bl}{c} - \frac{2c^2e+b^2l-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\int \frac{1}{\frac{\sqrt[3]{b-\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}\,dx}{3\ 2^{2/3}\sqrt[3]{c}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$+ \frac{\left(h - \frac{bl}{c} + \frac{2c^2e+b^2l-c(bh+2al)}{c\sqrt{b^2-4ac}}\right)\int \frac{\frac{\sqrt[3]{b-\sqrt{b^2-4ac}}}{\sqrt[3]{2}}+\sqrt[3]{c}x}{\frac{\left(b-\sqrt{b^2-4ac}\right)^{2/3}}{2^{2/3}}-\frac{\sqrt[3]{c}\sqrt[3]{b-\sqrt{b^2-4ac}}x}{\sqrt[3]{2}}+c^{2/3}x^2}\,dx}{3\ 2^{2/3}\sqrt[3]{c}\sqrt[3]{b-\sqrt{b^2-4ac}}}$$

$$= \text{Too large to display}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3 in optimal.

Time = 0.74 (sec) , antiderivative size = 223, normalized size of antiderivative = 0.13

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx$$
$$= \frac{6kx+3lx^2+2mx^3-2\text{RootSum}\left[a+b\#1^3+c\#1^6\&,\frac{-cd\log(x-\#1)+ak\log(x-\#1)-ce\log(x-\#1)\#1+al\log(x-\#\right.}{}$$

```
[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^3 + c*x^6),x]
```

```
[Out] (6*k*x + 3*l*x^2 + 2*m*x^3 - 2*RootSum[a + b*#1^3 + c*#1^6 & , (-(c*d*Log[x
 - #1]) + a*k*Log[x - #1] - c*e*Log[x - #1]*#1 + a*l*Log[x - #1]*#1 - c*f*L
og[x - #1]*#1^2 + a*m*Log[x - #1]*#1^2 - c*g*Log[x - #1]*#1^3 + b*k*Log[x -
 #1]*#1^3 - c*h*Log[x - #1]*#1^4 + b*l*Log[x - #1]*#1^4 - c*j*Log[x - #1]*#
1^5 + b*m*Log[x - #1]*#1^5)/(b*#1^2 + 2*c*#1^5) & ])/(6*c)
```

Maple [C] (verified)

Result contains higher order function than in optimal. Order 9 vs. order 3.

Time = 29.02 (sec) , antiderivative size = 130, normalized size of antiderivative = 0.08

method	result
default	$\frac{\frac{1}{3}m\,x^3+\frac{1}{2}l\,x^2+kx}{c}+\frac{\sum_{\_R=\text{RootOf}(\_Z^6c+\_Z^3b+a)}\frac{\left((-bm+cj)\_R^5+(-bl+ch)\_R^4+(-bk+gc)\_R^3+(-am+cf)\_R^2+(-al+ec)\_\right.}{2\_R^5c+\_R^2b}}{3c}$
risch	$\frac{m\,x^3}{3c}+\frac{l\,x^2}{2c}+\frac{kx}{c}+\frac{\sum_{\_R=\text{RootOf}(\_Z^6c+\_Z^3b+a)}\frac{\left((-bm+cj)\_R^5+(-bl+ch)\_R^4+(-bk+gc)\_R^3+(-am+cf)\_R^2+(-al+ec)\_\right.}{2\_R^5c+\_R^2b}}{3c}$

```
[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^6+b*x^3+a),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/c*(1/3*m*x^3+1/2*l*x^2+k*x)+1/3/c*sum(((-b*m+c*j)*_R^5+(-b*l+c*h)*_R^4+(-
b*k+c*g)*_R^3+(-a*m+c*f)*_R^2+(-a*l+c*e)*_R-a*k+c*d)/(2*_R^5*c+_R^2*b)*ln(x
-_R),_R=RootOf(_Z^6*c+_Z^3*b+a))
```

Fricas [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx = \text{Timed out}$$

```
[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^6+b*x^3+a)
,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx = \text{Timed out}$$

```
[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**6+
b*x**3+a),x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3+hx^4+jx^5+kx^6+lx^7+mx^8}{a+bx^3+cx^6}\,dx = \int \frac{mx^8+lx^7+kx^6+jx^5+hx^4+gx^3+fx^2+ex+d}{cx^6+bx^3+a}\,dx$$

```
[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^6+b*x^3+a)
,x, algorithm="maxima")
```

```
[Out] 1/6*(2*m*x^3 + 3*l*x^2 + 6*k*x)/c - integrate(-((c*j - b*m)*x^5 + (c*h - b*
l)*x^4 + (c*g - b*k)*x^3 + (c*f - a*m)*x^2 + c*d - a*k + (c*e - a*l)*x)/(c*
x^6 + b*x^3 + a), x)/c
```

Giac [F(-1)]

Timed out.

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + jx^5 + kx^6 + lx^7 + mx^8}{a + bx^3 + cx^6} \, dx = \text{Timed out}$$

```
[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^6+b*x^3+a)
,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 52.38 (sec) , antiderivative size = 359169, normalized size of antiderivative = 215.33

$$\int \frac{d + ex + fx^2 + gx^3 + hx^4 + jx^5 + kx^6 + lx^7 + mx^8}{a + bx^3 + cx^6} \, dx = \text{Too large to display}$$

```
[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^3 + c*x^6),x)
```

```
[Out] symsum(log((x*(c^7*e^5 + c^7*d^4*j - a^5*c^2*l^5 - b^7*e^2*m^3 - a^2*b*c^4*
h^5 - a*c^6*e^2*g^3 - b*c^6*e^2*f^3 + 2*a*c^6*e^3*h^2 + b*c^6*d^3*h^2 + a^2
*c^5*e*h^4 + a^4*b^2*c*l^5 + 3*c^7*d^2*e*f^2 + 3*c^7*d^2*e^2*g + a^2*b^5*d*
m^4 - a^2*c^5*g^4*j + a^3*c^4*g*j^4 + 5*a^4*c^3*e*l^4 + 3*b^2*c^5*e^4*l + b
^6*c*e^2*l^3 - a^3*b^4*g*m^4 - a^3*c^4*h^4*l - a^5*c^2*g*m^4 + a^4*c^3*j*k^
4 + a^4*b^3*k*m^4 + b^2*c^5*e^2*g^3 + 3*b^2*c^5*e^3*h^2 - b^3*c^4*e^2*h^3 +
 a^2*c^5*e^2*j^3 + a^2*c^5*g^3*h^2 + b^4*c^3*e^2*j^3 + 10*a^2*c^5*e^3*l^2 -
 10*a^3*c^4*e^2*l^3 + b^3*c^4*d^3*l^2 - b^5*c^2*e^2*k^3 - a^3*c^4*h^2*j^3 +
 3*b^4*c^3*e^3*l^2 - a^3*c^4*g^3*l^2 - a^2*b^5*h^2*m^3 - 2*a^4*c^3*h^2*l^3
+ a^4*c^3*j^3*l^2 - a^4*b^3*l^2*m^3 + b*c^6*d*f^4 - a*c^6*f^4*g - 3*b*c^6*e
^4*h - 4*c^7*d*e^3*f - 2*c^7*d^3*e*h - 2*c^7*d^3*f*g - 5*a*c^6*e^4*l - b*c^
6*d^4*m + b^7*d*f*m^3 + a*b*c^5*f*g^4 - 2*a*c^6*d*f*g^3 + 2*a*c^6*e*f^3*h +
 3*b*c^6*e^3*f*g + 2*a*c^6*d*f^3*j + 3*b*c^6*d*e^3*j + 4*a*c^6*d*e^3*m + 4*
a*c^6*e^3*f*k + 4*a*c^6*e^3*g*j + 2*b*c^6*d^3*e*l + 2*b*c^6*d^3*f*k - b*c^6
*d^3*g*j - b^6*c*d*f*l^3 + 2*a*c^6*d^3*g*m + 2*a*c^6*d^3*h*l + 2*a*b^6*e*h*
m^3 - a*b^6*f*g*m^3 - 4*a*c^6*d^3*j*k - a*b^6*d*j*m^3 - 2*a^5*b*c*k*m^4 + 1
2*a^2*b^2*c^3*e^2*l^3 + a^2*b^2*c^3*h^2*j^3 - 10*a^2*b^3*c^2*e^2*m^3 - a^2*
b^3*c^2*h^2*k^3 - 3*a^2*b^3*c^2*h^3*l^2 + 3*a^3*b^2*c^2*h^2*l^3 - 4*a*b*c^5
*e^2*h^3 + 2*a*b^2*c^4*e*h^4 + a*b^3*c^3*d*j^4 - 2*a^2*b*c^4*d*j^4 - 3*b*c^
6*d*e^2*g^2 - 2*a*b*c^5*d^3*l^2 + 3*a*c^6*e*f^2*g^2 + 3*b*c^6*d^2*f*g^2 - b
^2*c^5*d*f*g^3 + 3*a*c^6*d^2*e*j^2 - 3*a*c^6*d^2*g*h^2 - 3*b*c^6*d^2*f^2*h
+ 2*a^2*b^4*c*e*l^4 - 2*a^3*b*c^3*f*k^4 - 4*a^3*b^3*c*d*m^4 + 3*a^4*b*c^2*d
*m^4 + b^3*c^4*d*f*h^3 + 6*a*b^5*c*e^2*m^3 + 2*a^2*c^5*d*f*j^3 - 3*a*c^6*e^
2*f^2*j - 3*b*c^6*d^2*e^2*k - 2*a^2*c^5*f*g*h^3 - 3*b^2*c^5*d*f^3*j - b^4*c
```

```
^3*d*f*j^3 - 3*a*c^6*d^2*f^2*l - 2*a^2*c^5*d*h^3*j - 2*a^3*b^3*c*h*l^4 + 4*
a^3*c^4*d*f*l^3 + a^4*b*c^2*h*l^4 + 3*a^4*b^2*c*g*m^4 + b^5*c^2*d*f*k^3 + a
^3*b*c^3*j^4*k - 3*b^2*c^5*d*e^3*m - 3*b^2*c^5*e^3*f*k - 3*b^2*c^5*e^3*g*j
+ 2*a^2*c^5*d*g^3*m + 2*a^2*c^5*e*g^3*l + 2*a^2*c^5*f*g^3*k + 2*a^3*c^4*e*h
*k^3 + 2*a^3*c^4*f*g*k^3 + 3*b^3*c^4*d*f^3*m - 4*a^3*c^4*d*j*k^3 + b^2*c^5*
d^3*g*m - 2*b^2*c^5*d^3*h*l + 4*a^2*c^5*f^3*g*m - 2*a^2*c^5*f^3*h*l + a^4*b
*c^2*k^4*m - 2*a^4*c^3*e*h*m^3 + 4*a^4*c^3*f*g*m^3 + b^2*c^5*d^3*j*k + 3*b^
3*c^4*e^3*g*m - 6*b^3*c^4*e^3*h*l - 2*a^2*c^5*f^3*j*k - 2*a^3*c^4*d*j^3*m -
 2*a^3*c^4*e*j^3*l - 2*a^3*c^4*f*j^3*k - 2*a^4*c^3*d*j*m^3 + 2*a^5*b*c*l^2*
m^3 + 3*b^3*c^4*e^3*j*k + 2*a^3*c^4*g*h^3*m - 2*a^2*b^5*e*l*m^3 + a^2*b^5*f
*k*m^3 + a^2*b^5*g*j*m^3 - 4*a^2*c^5*e^3*k*m + 2*a^3*c^4*h^3*j*k - 4*a^4*c^
3*d*l^3*m - 4*a^4*c^3*f*k*l^3 - 4*a^4*c^3*g*j*l^3 - b^3*c^4*d^3*k*m + 3*b^6
*c*e^2*j*m^2 - 3*b^4*c^3*e^3*k*m - 2*a^3*c^4*g^3*k*m - 2*a^4*c^3*g*k^3*m -
2*a^4*c^3*h*k^3*l + 2*a^3*b^4*h*l*m^3 - a^3*b^4*j*k*m^3 + 2*a^5*c^2*h*l*m^3
 + 2*a^4*c^3*j^3*k*m + 2*a^5*c^2*j*k*m^3 + 4*a^5*c^2*k*l^3*m - 3*a*b^2*c^4*
e^2*j^3 + 4*a*b^3*c^3*e^2*k^3 - 3*a^2*b*c^4*e^2*k^3 - 10*a*b^2*c^4*e^3*l^2
- 5*a*b^4*c^2*e^2*l^3 - a^2*b^2*c^3*g*j^4 + a^2*b^3*c^2*f*k^4 - 6*a^3*b^2*c
^2*e*l^4 + a^2*b*c^4*f^3*l^2 + 4*a^3*b*c^3*e^2*m^3 + 2*a^3*b*c^3*h^2*k^3 -
3*b^3*c^4*d*e^2*k^2 + 3*b^3*c^4*d*f^2*j^2 + 3*a^2*b^2*c^3*h^4*l + a^2*b^4*c
*h^2*l^3 + 3*a^2*c^5*e*f^2*k^2 + 3*a^2*c^5*e*g^2*j^2 + 3*a^2*c^5*d^2*e*m^2
+ 3*b^2*c^5*e^2*f^2*j + 3*b^3*c^4*d^2*f*k^2 - 3*b^3*c^4*e^2*f*j^2 + 3*a^2*c
^5*d^2*g*l^2 + 3*a^2*c^5*e^2*g*k^2 - a^3*b^2*c^2*j*k^4 + 4*a^3*b^3*c*h^2*m^
3 - 3*a^4*b*c^2*h^2*m^3 + 3*b^2*c^5*d^2*f^2*l + 3*a^2*c^5*f^2*h^2*j - 3*b^3
*c^4*e^2*g^2*k + 3*b^4*c^3*e^2*g*k^2 + 3*b^5*c^2*d*f^2*m^2 + 6*a^2*c^5*d^2*
j*k^2 - 6*a^2*c^5*e^2*h^2*l - 3*a^2*c^5*f^2*g^2*l + 3*a^3*c^4*e*g^2*m^2 + 6
*a^3*c^4*e*h^2*l^2 - a^4*b*c^2*k^3*l^2 - 3*b^3*c^4*e^2*f^2*m - 3*b^5*c^2*e^
2*f*m^2 - 3*a^2*c^5*d^2*j^2*l + 3*a^3*c^4*e*j^2*k^2 - 3*a^3*c^4*g*h^2*k^2 -
 6*a^3*c^4*f^2*g*m^2 + 3*b^4*c^3*e^2*h^2*l - 3*b^5*c^2*e^2*h*l^2 - 3*a^3*c^
4*e^2*j*m^2 - 3*a^3*c^4*f^2*j*l^2 - 3*a^3*c^4*d^2*l*m^2 - 3*a^3*c^4*f^2*k^2
*l - 3*a^3*c^4*g^2*j^2*l + 3*a^4*c^3*e*k^2*m^2 - 3*b^5*c^2*e^2*j^2*m + 3*a^
4*c^3*g*k^2*l^2 + 3*a^4*c^3*h^2*j*m^2 - 3*a^4*c^3*g^2*l*m^2 - 3*a^4*c^3*j^2
*k^2*l - 3*a^5*c^2*j*l^2*m^2 - 3*a^5*c^2*k^2*l*m^2 - 6*a^2*b^2*c^3*d*f*l^3
- 3*a*b^2*c^4*d^2*g*l^2 - 9*a*b^2*c^4*e^2*g*k^2 - 3*a*b^2*c^4*f^2*g*j^2 - 1
2*a*b^3*c^3*d*f^2*m^2 + 12*a^2*b*c^4*d*f^2*m^2 + 3*a^2*b*c^4*d*g^2*l^2 - 3*
a^2*b*c^4*d*h^2*k^2 + 13*a^2*b^3*c^2*d*f*m^3 + 3*a*b^3*c^3*f*g^2*k^2 + 12*a
*b^3*c^3*e^2*f*m^2 - 3*a^2*b*c^4*f*g^2*k^2 - 3*a^2*b*c^4*f*h^2*j^2 - 9*a^2*
b*c^4*e^2*f*m^2 - 6*a^2*b^2*c^3*e*h*k^3 - 3*a*b^2*c^4*d^2*j*k^2 + 3*a*b^2*c
^4*e^2*h^2*l + 3*a*b^2*c^4*f^2*g^2*l + 6*a*b^3*c^3*e^2*h*l^2 + 6*a*b^4*c^2*
e*h^2*l^2 - 3*a^2*b*c^4*d^2*h*m^2 - 6*a^2*b*c^4*e^2*h*l^2 - 3*a^2*b*c^4*f^2
*h*k^2 - 3*a^2*b*c^4*g^2*h*j^2 + 3*a^2*b^2*c^3*d*j*k^3 + 2*a^2*b^3*c^2*e*h*
l^3 - 4*a^2*b^3*c^2*f*g*l^3 + 3*a*b^2*c^4*d^2*j^2*l - 3*a*b^4*c^2*f^2*g*m^2
 - 3*a^2*b*c^4*g^2*h^2*k - 4*a^2*b^3*c^2*d*j*l^3 + 12*a^3*b^2*c^2*e*h*m^3 -
 9*a^3*b^2*c^2*f*g*m^3 + 3*a^2*b*c^4*d^2*k*l^2 - 3*a^2*b*c^4*f^2*h^2*m + 3*
a^2*b*c^4*f^2*j^2*k + 8*a^2*b^2*c^3*d*j^3*m + 2*a^2*b^2*c^3*e*j^3*l - a^2*b
^2*c^3*f*j^3*k + 3*a^3*b*c^3*d*j^2*m^2 + 6*a^3*b*c^3*f*h^2*m^2 + 3*a^3*b^2*
```

c^2*d*j*m^3 + 12*a*b^3*c^3*e^2*j^2*m - 15*a*b^4*c^2*e^2*j*m^2 - 9*a^2*b*c^4
*e^2*j^2*m - 3*a^2*b^2*c^3*g*h^3*m + a^2*b^3*c^2*d*k^3*m - 2*a^2*b^3*c^2*e*
k^3*l + a^2*b^3*c^2*g*j*k^3 - 3*a^3*b*c^3*g^2*h*m^2 - 3*a^2*b^2*c^3*h^3*j*k
 - 3*a^3*b*c^3*h*j^2*k^2 + 3*a^3*b^2*c^2*d*l^3*m + 3*a^3*b^2*c^2*f*k*l^3 +
3*a^3*b^2*c^2*g*j*l^3 + 3*a^2*b^3*c^2*g*j^3*m - 3*a^2*b^4*c*g*j^2*m^2 - 6*a
^3*b*c^3*f^2*k*m^2 + 3*a^2*b^4*c*h^2*j*m^2 + 3*a^3*b*c^3*g^2*k^2*m + 6*a^3*
b*c^3*h^2*j^2*m - a^3*b^2*c^2*g*k^3*m + 2*a^3*b^2*c^2*h*k^3*l - 3*a^4*b*c^2
*f*l^2*m^2 + 3*a^2*b^3*c^2*h^3*k*m - 3*a^4*b*c^2*h*k^2*m^2 - 3*a^3*b^2*c^2*
j^3*k*m + 3*a^3*b^3*c*j^2*k*m^2 - 3*a^4*b*c^2*j^2*k*m^2 - 3*a^4*b*c^2*j^2*l
^2*m + 3*a^4*b^2*c*j*l^2*m^2 + 3*a^4*b^2*c*k^2*l*m^2 - 2*a*b*c^5*d*f*h^3 -
2*a*b*c^5*e*g^3*h + a*b*c^5*d*g^3*j - 3*b*c^6*d*e*f^2*g + 6*a*c^6*d*e*g^2*h
 + 6*b*c^6*d*e^2*f*h - 6*a*b*c^5*d*f^3*m + 3*a*b*c^5*f^3*g*j - 6*a*b^5*c*d*
f*m^3 - 6*a*c^6*d*e*f^2*k - 6*a*c^6*e^2*f*g*h - 3*b*c^6*d^2*e*f*j - 7*a*b*c
^5*e^3*g*m + 8*a*b*c^5*e^3*h*l - 2*a*b^5*c*e*h*l^3 + a*b^5*c*f*g*l^3 + 12*a
*c^6*d*e^2*f*l - 6*a*c^6*d*e^2*g*k - 6*a*c^6*d*e^2*h*j - 7*a*b*c^5*e^3*j*k
+ a*b^5*c*d*j*l^3 - 6*a*c^6*d^2*e*f*m - 6*a*c^6*d^2*e*g*l + 6*a*c^6*d^2*e*h
*k + 6*a*c^6*d^2*f*g*k + 2*a*b*c^5*d^3*k*m - 3*b^6*c*d*f*j*m^2 - 3*b^6*c*e^
2*k*l*m - 9*a^2*b^2*c^3*e*h^2*l^2 + 3*a^2*b^2*c^3*g*h^2*k^2 + 9*a^2*b^2*c^3
*f^2*g*m^2 - 9*a^2*b^3*c^2*d*j^2*m^2 - 3*a^2*b^3*c^2*f*h^2*m^2 + 18*a^2*b^2
*c^3*e^2*j*m^2 - 3*a^2*b^2*c^3*g^2*j*k^2 + 3*a^2*b^2*c^3*d^2*l*m^2 + 3*a^2*
b^2*c^3*f^2*k^2*l + 3*a^2*b^2*c^3*g^2*j^2*l + 3*a^2*b^3*c^2*f^2*k*m^2 + 6*a
^3*b^2*c^2*g*j^2*m^2 - 3*a^2*b^3*c^2*h^2*j^2*m - 9*a^3*b^2*c^2*h^2*j*m^2 +
3*a^3*b^2*c^2*g^2*l*m^2 + 3*a^3*b^2*c^2*j^2*k^2*l + 6*a*b*c^5*d*e^2*k^2 - 3
*a*b*c^5*d*f^2*j^2 - 6*a*b*c^5*d^2*f*k^2 + 6*a*b*c^5*e^2*f*j^2 - 3*a*b*c^5*
f^2*g^2*h - a*b^2*c^4*f*g*h^3 - 4*a*b^3*c^3*d*f*k^3 + 6*a^2*b*c^4*d*f*k^3 -
 3*a*b*c^5*d^2*h*j^2 - a*b^2*c^4*d*h^3*j + 5*a*b^4*c^2*d*f*l^3 - 3*b^2*c^5*
d*e*f*h^2 + 6*a*b*c^5*e^2*g^2*k - 2*a*b^3*c^3*e*h*j^3 + a*b^3*c^3*f*g*j^3 +
 4*a^2*b*c^4*e*h*j^3 + a^2*b*c^4*f*g*j^3 - 10*a^3*b*c^3*d*f*m^3 - 3*a*b*c^5
*d^2*g^2*m + 6*a*b*c^5*e^2*f^2*m - 3*a*b^2*c^4*f*g^3*k + 2*a*b^4*c^2*e*h*k^
3 - a*b^4*c^2*f*g*k^3 + 3*b^2*c^5*d*f^2*g*h - 6*a*b^3*c^3*e*h^3*l - a*b^4*c
^2*d*j*k^3 + 4*a^2*b*c^4*d*h^3*m + 4*a^2*b*c^4*e*h^3*l + 4*a^2*b*c^4*f*h^3*
k + 4*a^2*b*c^4*g*h^3*j - 12*a^2*c^5*d*e*f*l^2 + 3*a^3*b*c^3*f*g*l^3 + 3*b^
2*c^5*d*e*f^2*k - 3*b^2*c^5*e^2*f*g*h - 3*a*b^2*c^4*f^3*g*m - 10*a^2*b^4*c*
e*h*m^3 + 5*a^2*b^4*c*f*g*m^3 - 6*a^2*c^5*d*e*h*k^2 - 6*a^2*c^5*d*f*g*k^2 +
 3*a^3*b*c^3*d*j*l^3 - 6*b^2*c^5*d*e^2*f*l + 6*b^2*c^5*d*e^2*g*k - 3*b^2*c^
5*d*e^2*h*j - 3*b^4*c^3*d*e*f*l^2 - 3*a*b^4*c^2*d*j^3*m + 3*a*b^5*c*d*j^2*m
^2 + 2*a^2*b^4*c*d*j*m^3 - 6*a^2*c^5*e*f*h*j^2 + 3*b^2*c^5*d^2*e*f*m - 6*b^
2*c^5*d^2*f*g*k + 3*b^2*c^5*d^2*f*h*j + 3*b^3*c^4*d*f*g^2*k - 3*b^4*c^3*d*f
*g*k^2 + 3*a^2*b*c^4*g^3*j*k - 6*a^2*c^5*d*e*j^2*k + 6*a^2*c^5*d*g*h^2*k -
2*a^3*b*c^3*d*k^3*m + 4*a^3*b*c^3*e*k^3*l + a^3*b*c^3*g*j*k^3 - 3*b^3*c^4*d
*f^2*g*l - 3*b^3*c^4*d*f^2*h*k + 10*a*b^2*c^4*e^3*k*m - a^2*b^4*c*d*l^3*m -
 a^2*b^4*c*f*k*l^3 - a^2*b^4*c*g*j*l^3 - 6*a^2*c^5*d*g^2*h*l - 6*a^2*c^5*e*
f*g^2*m - 6*a^2*c^5*e*g^2*h*k + 3*b^3*c^4*d*e^2*h*m + 3*b^3*c^4*e^2*f*g*l +
 3*b^3*c^4*e^2*f*h*k + 3*b^3*c^4*e^2*g*h*j - 3*b^4*c^3*d*f*h^2*l + 3*b^5*c^
2*d*f*h*l^2 + 2*a^2*b*c^4*f^3*k*m - 6*a^2*c^5*e*f^2*h*m - 5*a^3*b*c^3*g*j^3

```
*m - 2*a^3*b*c^3*h*j^3*l + 8*a^3*b^3*c*e*l*m^3 - 4*a^3*b^3*c*f*k*m^3 - a^3*
b^3*c*g*j*m^3 + 6*a^3*c^4*e*f*h*m^2 - 6*a^4*b*c^2*e*l*m^3 + 6*a^4*b*c^2*f*k
*m^3 - 3*a^4*b*c^2*g*j*m^3 + 3*b^3*c^4*d*e^2*j*l - 3*b^3*c^4*d^2*f*h*m - 6*
a^2*c^5*d*f^2*j*m + 6*a^2*c^5*d*f^2*k*l + 6*a^2*c^5*e*f^2*j*l + 6*a^2*c^5*e
^2*g*h*m - 6*a^3*c^4*d*e*k*m^2 + 6*a^3*c^4*d*f*j*m^2 - 6*a^3*c^4*f*g*h*l^2
- 3*b^3*c^4*d^2*f*j*l - 12*a^2*c^5*d*e^2*l*m + 6*a^2*c^5*e^2*f*j*m - 12*a^2
*c^5*e^2*f*k*l - 12*a^2*c^5*e^2*g*j*l + 6*a^2*c^5*e^2*h*j*k - 6*a^3*b*c^3*h
^3*k*m + a^3*b^3*c*g*l^3*m + 12*a^3*c^4*d*e*l^2*m - 6*a^3*c^4*d*g*k*l^2 - 6
*a^3*c^4*d*h*j*l^2 + 12*a^3*c^4*e*f*k*l^2 + 12*a^3*c^4*e*g*j*l^2 + a^4*b*c^
2*g*l^3*m - 6*b^4*c^3*d*f^2*j*m + 3*b^4*c^3*d*f^2*k*l - 3*b^4*c^3*e^2*g*h*m
 + 3*b^5*c^2*d*f*j^2*m + 6*a^2*c^5*d^2*f*l*m - 6*a^2*c^5*d^2*g*k*m - 6*a^2*
c^5*d^2*h*k*l + a^3*b^3*c*j*k*l^3 + 6*a^3*c^4*d*g*k^2*m + 6*a^3*c^4*d*h*k^2
*l - 6*a^3*c^4*e*f*k^2*m - 6*a^3*c^4*e*g*k^2*l + a^4*b*c^2*j*k*l^3 - 6*a^4*
b^2*c*h*l*m^3 - 3*b^4*c^3*d*e^2*l*m + 6*b^4*c^3*e^2*f*j*m - 3*b^4*c^3*e^2*f
*k*l - 3*b^4*c^3*e^2*g*j*l - 3*b^4*c^3*e^2*h*j*k + 6*a^3*c^4*e*h*j^2*m + 6*
a^3*c^4*f*h*j^2*l + 3*b^4*c^3*d^2*f*l*m + 6*a^3*c^4*d*j^2*k*l - 6*a^3*c^4*f
*h^2*j*m + 6*a^3*c^4*f*g^2*l*m + 6*a^3*c^4*g^2*h*k*l - 4*a^4*b^2*c*k*l^3*m
+ 3*b^5*c^2*e^2*g*l*m + 3*b^5*c^2*e^2*h*k*m + 6*a^3*c^4*f^2*h*l*m - 6*a^4*c
^3*f*h*l*m^2 + 3*b^5*c^2*e^2*j*k*l + 6*a^3*c^4*f^2*j*k*m + 6*a^4*c^3*d*k*l*
m^2 + 6*a^4*c^3*e*j*l*m^2 - 6*a^4*c^3*f*j*k*m^2 + 6*a^4*c^3*g*h*l^2*m + 12*
a^3*c^4*e^2*k*l*m - 12*a^4*c^3*e*k*l^2*m + 6*a^4*c^3*f*j*l^2*m + 6*a^4*c^3*
h*j*k*l^2 + 6*a^4*c^3*f*k^2*l*m - 6*a^4*c^3*h*j^2*l*m + 12*a*b^2*c^4*d*e*f*
l^2 + 6*a*b^2*c^4*d*e*h*k^2 + 6*a*b^2*c^4*d*f*g*k^2 + 3*a*b^2*c^4*d*g*h*j^2
 + 6*a*b^2*c^4*e*f*h*j^2 - 3*a^2*b*c^4*d*e*g*m^2 - 6*a*b^2*c^4*d*e*h^2*m +
3*a*b^2*c^4*d*e*j^2*k + 9*a*b^2*c^4*d*f*h^2*l - 6*a*b^2*c^4*e*f*h^2*k - 6*a
*b^2*c^4*e*g*h^2*j - 12*a*b^3*c^3*d*f*h*l^2 + 3*a*b^3*c^3*e*f*g*l^2 + 6*a^2
*b*c^4*d*f*h*l^2 - 3*a^2*b*c^4*e*f*g*l^2 + 3*a*b^2*c^4*e*f*g^2*m + 6*a*b^2*
c^4*e*g^2*h*k + 3*a*b^2*c^4*f*g^2*h*j + 3*a*b^3*c^3*d*e*j*l^2 - 6*a*b^3*c^3
*e*g*h*k^2 - 3*a^2*b*c^4*d*e*j*l^2 + 12*a^2*b*c^4*e*g*h*k^2 - 3*a*b^2*c^4*d
*g^2*j*k + 6*a*b^2*c^4*e*f^2*h*m + 3*a*b^2*c^4*f^2*g*h*k + 3*a*b^3*c^3*d*g*
j*k^2 + 6*a*b^4*c^2*e*f*h*m^2 - 6*a^2*b*c^4*d*e*k^2*l - 3*a^2*b*c^4*d*g*j*k
^2 - 3*a^2*b*c^4*e*f*j*k^2 + 15*a*b^2*c^4*d*f^2*j*m - 9*a*b^2*c^4*d*f^2*k*l
 + 3*a*b^2*c^4*e^2*g*h*m - 6*a*b^3*c^3*d*f*j^2*m - 3*a*b^3*c^3*d*g*j^2*l -
3*a*b^3*c^3*d*h*j^2*k + 6*a*b^3*c^3*e*g*h^2*m + 3*a*b^3*c^3*f*g*h^2*l + 12*
a*b^4*c^2*d*f*j*m^2 - 3*a*b^4*c^2*f*g*h*l^2 + 3*a^2*b*c^4*d*g*j^2*l + 6*a^2
*b*c^4*d*h*j^2*k - 6*a^2*b*c^4*e*f*j^2*l - 9*a^2*b*c^4*e*g*h^2*m - 3*a^2*b*
c^4*e*g*j^2*k - 3*a^2*b*c^4*f*g*h^2*l + 9*a*b^2*c^4*d*e^2*l*m - 18*a*b^2*c^
4*e^2*f*j*m + 9*a*b^2*c^4*e^2*f*k*l + 9*a*b^2*c^4*e^2*g*j*l + 3*a*b^2*c^4*e
^2*h*j*k + 3*a*b^3*c^3*d*h^2*j*l + 6*a*b^3*c^3*e*h^2*j*k - 3*a*b^3*c^3*f*g^
2*h*m - 3*a*b^4*c^2*d*h*j*l^2 - 3*a^2*b*c^4*d*h^2*j*l - 9*a^2*b*c^4*e*h^2*j
*k + 6*a^2*b*c^4*f*g^2*h*m - 9*a*b^2*c^4*d^2*f*l*m + 3*a*b^2*c^4*d^2*g*k*m
+ 3*a*b^2*c^4*d^2*h*j*m - 3*a*b^3*c^3*f*g^2*j*l - 3*a^2*b*c^4*e*g^2*j*m - 6
*a^2*b*c^4*e*g^2*k*l + 3*a^2*b*c^4*f*g^2*j*l + 6*a*b^3*c^3*f^2*g*j*m - 3*a*
b^3*c^3*f^2*g*k*l + 6*a*b^4*c^2*e*h*j^2*m - 3*a*b^4*c^2*f*g*j^2*m - 6*a^2*b
*c^4*e*f^2*l*m - 9*a^2*b*c^4*f^2*g*j*m + 3*a^2*b*c^4*f^2*g*k*l + 3*a^3*b*c^
```

3*d*g*l*m^2 + 6*a^3*b*c^3*d*h*k*m^2 + 12*a^3*b*c^3*e*f*l*m^2 - 3*a^3*b*c^3*
e*g*k*m^2 - 18*a^3*b*c^3*e*h*j*m^2 + 9*a^3*b*c^3*f*g*j*m^2 - 12*a*b^3*c^3*e
^2*g*l*m - 6*a*b^3*c^3*e^2*h*k*m + 3*a*b^4*c^2*d*j^2*k*l - 6*a*b^4*c^2*e*h^
2*k*m + 15*a^2*b*c^4*e^2*g*l*m - 6*a^2*b*c^4*e^2*h*k*m - 9*a^3*b*c^3*e*g*l^
2*m - 12*a*b^3*c^3*e^2*j*k*l + 3*a*b^4*c^2*f*g^2*l*m + 15*a^2*b*c^4*e^2*j*k
*l - 9*a^3*b*c^3*e*j*k*l^2 + 6*a^3*b*c^3*f*h*k^2*m - 6*a^3*b*c^3*g*h*k^2*l
- 3*a*b^3*c^3*d^2*j*l*m + 3*a^2*b*c^4*d^2*j*l*m - 3*a^3*b*c^3*e*j*k^2*m + 3
*a^3*b*c^3*f*j*k^2*l + 3*a^2*b^4*c*d*k*l*m^2 + 6*a^2*b^4*c*e*j*l*m^2 - 3*a^
2*b^4*c*f*j*k*m^2 + 12*a^3*b*c^3*e*j^2*l*m + 3*a^3*b*c^3*g*h^2*l*m + 3*a^3*
b*c^3*g*j^2*k*l + 15*a*b^4*c^2*e^2*k*l*m - 6*a^2*b^4*c*e*k*l^2*m + 3*a^3*b*
c^3*h^2*j*k*l + 3*a^2*b^4*c*f*k^2*l*m + 3*a^3*b*c^3*g^2*j*l*m - 3*a^3*b^3*c
*g*k*l*m^2 - 6*a^3*b^3*c*h*j*l*m^2 + 3*a^4*b*c^2*g*k*l*m^2 + 12*a^4*b*c^2*h
*j*l*m^2 - 3*a^2*b^4*c*h^2*k*l*m + 6*a^3*b^3*c*h*k*l^2*m - 6*a^4*b*c^2*h*k*
l^2*m - 3*a^3*b^3*c*j*k^2*l*m + 3*a^4*b*c^2*j*k^2*l*m + 3*b^6*c*d*f*k*l*m +
3*a^2*b^2*c^3*d*g*h*m^2 - 18*a^2*b^2*c^3*e*f*h*m^2 + 3*a^2*b^2*c^3*d*e*k*m
^2 - 15*a^2*b^2*c^3*d*f*j*m^2 + 9*a^2*b^2*c^3*f*g*h*l^2 - 9*a^2*b^2*c^3*d*e
*l^2*m + 9*a^2*b^2*c^3*d*h*j*l^2 - 9*a^2*b^2*c^3*e*f*k*l^2 - 9*a^2*b^2*c^3*
e*g*j*l^2 - 3*a^2*b^2*c^3*d*g*k^2*m + 3*a^2*b^2*c^3*e*f*k^2*m + 6*a^2*b^2*c
^3*e*g*k^2*l + 3*a^2*b^2*c^3*f*h*j*k^2 - 18*a^2*b^2*c^3*e*h*j^2*m + 3*a^2*b
^2*c^3*f*g*j^2*m + 3*a^2*b^2*c^3*g*h*j^2*k - 3*a^2*b^3*c^2*d*g*l*m^2 - 3*a^
2*b^3*c^2*d*h*k*m^2 - 6*a^2*b^3*c^2*e*f*l*m^2 + 24*a^2*b^3*c^2*e*h*j*m^2 -
9*a^2*b^3*c^2*f*g*j*m^2 - 6*a^2*b^2*c^3*d*h^2*l*m - 9*a^2*b^2*c^3*d*j^2*k*l
+ 15*a^2*b^2*c^3*e*h^2*k*m + 6*a^2*b^2*c^3*f*h^2*j*m - 6*a^2*b^2*c^3*f*h^2
*k*l - 6*a^2*b^2*c^3*g*h^2*j*l + 3*a^2*b^3*c^2*d*h*l^2*m + 6*a^2*b^3*c^2*e*
g*l^2*m + 3*a^2*b^3*c^2*f*h*k*l^2 + 3*a^2*b^3*c^2*g*h*j*l^2 - 9*a^2*b^2*c^3
*f*g^2*l*m + 3*a^2*b^2*c^3*g^2*h*j*m + 6*a^2*b^3*c^2*e*j*k*l^2 - 3*a^2*b^3*
c^2*f*h*k^2*m - 3*a^2*b^3*c^2*f*j*k^2*l + 6*a^3*b^2*c^2*f*h*l*m^2 + 3*a^3*b
^2*c^2*g*h*k*m^2 - 6*a^2*b^2*c^3*f^2*j*k*m - 6*a^2*b^3*c^2*e*j^2*l*m + 3*a^
2*b^3*c^2*f*j^2*k*m + 3*a^2*b^3*c^2*g*h^2*l*m - 3*a^2*b^3*c^2*g*j^2*k*l - 9
*a^3*b^2*c^2*d*k*l*m^2 - 18*a^3*b^2*c^2*e*j*l*m^2 + 6*a^3*b^2*c^2*f*j*k*m^2
- 9*a^3*b^2*c^2*g*h*l^2*m - 24*a^2*b^2*c^3*e^2*k*l*m + 3*a^2*b^3*c^2*h^2*j
*k*l + 18*a^3*b^2*c^2*e*k*l^2*m - 9*a^3*b^2*c^2*h*j*k*l^2 - 3*a^2*b^3*c^2*g
^2*j*l*m - 9*a^3*b^2*c^2*f*k^2*l*m + 3*a^3*b^2*c^2*h*j*k^2*m + 6*a^3*b^2*c^
2*h*j^2*l*m + 3*a^3*b^2*c^2*h^2*k*l*m - 3*a*b*c^5*d*e*g*j^2 + 9*a*b*c^5*e*f
*g*h^2 + 9*a*b*c^5*d*e*h^2*j - 3*a*b*c^5*e*f*g^2*j + 3*a*b*c^5*d*f^2*g*l +
6*a*b*c^5*d*f^2*h*k - 3*a*b*c^5*e*f^2*g*k - 6*a*b*c^5*e*f^2*h*j - 3*a*b*c^5
*e^2*f*g*l - 3*a*b*c^5*d*e^2*j*l + 6*a*b*c^5*d^2*f*h*m + 6*a*b*c^5*d^2*g*h*
l + 3*b^2*c^5*d*e*f*g*j - 3*a*b*c^5*d^2*e*j*m + 3*a*b*c^5*d^2*f*j*l + 3*a*b
*c^5*d^2*g*j*k - 3*b^3*c^4*d*e*f*g*m + 6*b^3*c^4*d*e*f*h*l - 3*b^3*c^4*d*f*
g*h*j - 3*b^3*c^4*d*e*f*j*k - 6*a*b^5*c*e*h*j*m^2 + 3*a*b^5*c*f*g*j*m^2 + 1
2*a^2*c^5*e*f*g*h*l + 12*a^2*c^5*d*e*f*k*m + 12*a^2*c^5*d*e*g*k*l + 12*a^2*
c^5*d*e*h*j*l + 3*b^4*c^3*d*f*g*h*m + 3*b^4*c^3*d*e*f*k*m + 3*b^4*c^3*d*f*g
*j*l + 3*b^4*c^3*d*f*h*j*k - 3*b^5*c^2*d*f*g*l*m - 3*b^5*c^2*d*f*h*k*m - 3*
b^5*c^2*d*f*j*k*l - 12*a^3*c^4*e*g*h*l*m - 12*a^3*c^4*d*f*k*l*m - 12*a^3*c^
4*e*f*j*l*m - 12*a^3*c^4*e*h*j*k*l - 6*a*b^2*c^4*d*f*g*h*m - 12*a*b^2*c^4*e

```
*f*g*h*l - 12*a*b^2*c^4*d*e*f*k*m + 3*a*b^2*c^4*d*e*g*j*m - 12*a*b^2*c^4*d*
e*h*j*l - 3*a*b^2*c^4*d*f*g*j*l - 6*a*b^2*c^4*d*f*h*j*k + 3*a*b^2*c^4*e*f*g
*j*k + 6*a*b^3*c^3*d*e*h*l*m + 9*a*b^3*c^3*d*f*g*l*m + 12*a*b^3*c^3*d*f*h*k
*m - 3*a*b^3*c^3*d*g*h*j*m - 3*a*b^3*c^3*e*f*g*k*m - 12*a*b^3*c^3*e*f*h*j*m
 + 6*a*b^3*c^3*e*f*h*k*l + 6*a*b^3*c^3*e*g*h*j*l - 3*a*b^3*c^3*f*g*h*j*k -
6*a^2*b*c^4*d*f*g*l*m - 12*a^2*b*c^4*d*f*h*k*m + 6*a^2*b*c^4*e*f*g*k*m + 24
*a^2*b*c^4*e*f*h*j*m - 3*a*b^3*c^3*d*e*j*k*m + 9*a*b^3*c^3*d*f*j*k*l + 6*a^
2*b*c^4*d*e*j*k*m - 6*a^2*b*c^4*d*f*j*k*l - 6*a*b^4*c^2*e*g*h*l*m + 3*a*b^4
*c^2*f*g*h*k*m - 15*a*b^4*c^2*d*f*k*l*m + 3*a*b^4*c^2*d*g*j*l*m + 3*a*b^4*c
^2*d*h*j*k*m - 6*a*b^4*c^2*e*h*j*k*l + 3*a*b^4*c^2*f*g*j*k*l + 12*a^3*b*c^3
*e*h*k*l*m - 6*a^3*b*c^3*f*g*k*l*m - 12*a^3*b*c^3*f*h*j*l*m - 6*a^3*b*c^3*d
*j*k*l*m + 3*a^2*b^4*c*g*j*k*l*m + 12*a^2*b^2*c^3*e*g*h*l*m - 6*a^2*b^2*c^3
*f*g*h*k*m + 24*a^2*b^2*c^3*d*f*k*l*m - 3*a^2*b^2*c^3*d*g*j*l*m - 6*a^2*b^2
*c^3*d*h*j*k*m + 12*a^2*b^2*c^3*e*f*j*l*m + 3*a^2*b^2*c^3*e*g*j*k*m + 12*a^
2*b^2*c^3*e*h*j*k*l - 3*a^2*b^2*c^3*f*g*j*k*l - 18*a^2*b^3*c^2*e*h*k*l*m +
9*a^2*b^3*c^2*f*g*k*l*m - 3*a^2*b^3*c^2*g*h*j*k*m + 9*a^2*b^3*c^2*d*j*k*l*m
 - 3*a^3*b^2*c^2*g*j*k*l*m + 6*a*b*c^5*d*e*f*g*m - 12*a*b*c^5*d*e*f*h*l - 1
2*a*b*c^5*d*e*g*h*k + 6*a*b*c^5*d*e*f*j*k + 6*a*b^5*c*e*h*k*l*m - 3*a*b^5*c
*f*g*k*l*m - 3*a*b^5*c*d*j*k*l*m))/c^3 - (a*c^6*f^5 - c^7*d*e^4 + c^7*d^4*h
 - a^6*c*m^5 - c^7*d^3*f^2 + a^5*b^2*m^5 + a^2*c^5*d*h^4 - a^3*b*c^3*j^5 +
a*c^6*d^3*j^2 + 3*c^7*d^2*e^2*f - a^2*c^5*g^4*h + a^3*c^4*f*j^4 - a^4*c^3*d
*l^4 + a*b^6*f^2*m^3 + 2*a^3*b^4*f*m^4 - 5*a^2*c^5*f^4*m - a^3*c^4*h^4*k +
a^4*c^3*h*k^4 + 5*a^5*c^2*f*m^4 - 2*a^4*b^3*j*m^4 - a^4*c^3*j^4*m + a^5*c^2
*k*l^4 - a^2*c^5*f^2*h^3 - b^2*c^5*d^3*j^2 + 2*a^2*c^5*f^3*j^2 - a^2*c^5*d^
3*m^2 + a^3*c^4*f^2*k^3 + a^3*c^4*h^3*j^2 - b^4*c^3*d^3*m^2 + 10*a^3*c^4*f^
3*m^2 - 10*a^4*c^3*f^2*m^3 - a^4*c^3*h^3*m^2 - a^4*c^3*j^2*k^3 + a^3*b^4*j^
2*m^3 - 2*a^5*c^2*j^2*m^3 + a^5*c^2*k^3*m^2 - 2*c^7*d^3*e*g + a*c^6*e^4*k -
 b*c^6*d^4*l + b^7*d*e*m^3 + a*b*c^5*e*g^4 - 2*a*c^6*d*e*g^3 + b*c^6*d*e*f^
3 - 3*a*b*c^5*f^4*j - 4*a*c^6*d*f^3*h - 4*a*c^6*e*f^3*g + 3*b*c^6*d*e^3*h -
 2*a*c^6*e^3*g*h - b*c^6*d^3*g*h + 4*a*c^6*d*e^3*l - 2*a*c^6*e^3*f*j + 2*b*
c^6*d^3*e*k + 2*b*c^6*d^3*f*j - b^6*c*d*e*l^3 + 2*a*c^6*d^3*f*m + 2*a*c^6*d
^3*g*l - 4*a*c^6*d^3*h*k - a*b^6*d*h*m^3 - a*b^6*e*g*m^3 + a^5*b*c*j*m^4 -
3*a^2*b^2*c^3*f^2*k^3 + 4*a^2*b^3*c^2*f^2*l^3 - 10*a^2*b^2*c^3*f^3*m^2 + 12
*a^3*b^2*c^2*f^2*m^3 + a^3*b^2*c^2*j^2*k^3 - a*b^2*c^4*d*h^4 - a*b*c^5*f^2*
g^3 + a*b*c^5*e^3*j^2 + 3*a*c^6*d*f^2*g^2 + 3*b*c^6*d^2*e*g^2 + a^2*b*c^4*g
*h^4 - b^2*c^5*d*e*g^3 + 3*a*c^6*d^2*f*h^2 + 3*a*c^6*e^2*f*g^2 - a^2*b^4*c*
d*l^4 - 2*a^3*b*c^3*e*k^4 + b^3*c^4*d*e*h^3 + 3*a*c^6*e^2*f^2*h + 2*a^2*c^5
*d*e*j^3 - a*b^5*c*f^2*l^3 - 3*b*c^6*d^2*e^2*j - 2*a^2*c^5*e*g*h^3 - b^4*c^
3*d*e*j^3 + 3*a*b^2*c^4*f^4*m + 3*a*c^6*d^2*f^2*k + a^3*b^3*c*g*l^4 + 4*a^3
*c^4*d*e*l^3 - 2*a^4*b*c^2*g*l^4 - 6*a^4*b^2*c*f*m^4 + b^5*c^2*d*e*k^3 - 3*
a*c^6*d^2*e^2*m - 3*b^2*c^5*d*e^3*l + 2*a^2*c^5*d*g^3*l + 2*a^2*c^5*e*g^3*k
 + 2*a^2*c^5*f*g^3*j - 4*a^3*c^4*d*h*k^3 + 2*a^3*c^4*e*g*k^3 - 2*b^2*c^5*d^
3*f*m + b^2*c^5*d^3*g*l + b^2*c^5*d^3*h*k + 4*a^2*c^5*f^3*g*l + 4*a^2*c^5*f
^3*h*k - 2*a^3*c^4*g*h*j^3 + a^4*b*c^2*k^4*l - a^4*b^2*c*k*l^4 + 4*a^4*c^3*
d*h*m^3 + 4*a^4*c^3*e*g*m^3 - 2*a^3*c^4*d*j^3*l - 2*a^3*c^4*e*j^3*k + a^2*b
```

^5*g*h*m^3 + 2*a^3*c^4*f*h^3*m + 2*a^3*c^4*g*h^3*l + 2*a^4*c^3*g*h*l^3 - a^
5*b*c*l^3*m^2 + a^2*b^5*d*l*m^3 + a^2*b^5*e*k*m^3 - 2*a^2*b^5*f*j*m^3 + 2*a
^2*c^5*e^3*j*m - 4*a^2*c^5*e^3*k*l - 4*a^4*c^3*e*k*l^3 + 2*a^4*c^3*f*j*l^3
+ 2*b^3*c^4*d^3*j*m - b^3*c^4*d^3*k*l - 2*a^3*c^4*g^3*j*m - 2*a^3*c^4*g^3*k
*l - 2*a^4*c^3*f*k^3*m - 2*a^4*c^3*g*k^3*l - a^3*b^4*g*l*m^3 - a^3*b^4*h*k*
m^3 - 4*a^5*c^2*g*l*m^3 - 4*a^5*c^2*h*k*m^3 + 2*a^4*c^3*j^3*k*l + a^4*b^3*k
*l*m^3 - 2*a^5*c^2*j*l^3*m + a*b^2*c^4*f^2*h^3 + 3*a*b^2*c^4*f^3*j^2 - a*b^
3*c^3*f^2*j^3 - 4*a^2*b*c^4*f^2*j^3 + 2*a^2*b^2*c^3*f*j^4 + a^2*b^3*c^2*e*k
^4 + 3*a^3*b^2*c^2*d*l^4 - 3*b^2*c^5*d*e^2*h^2 + 3*a*b^2*c^4*d^3*m^2 + a*b^
4*c^2*f^2*k^3 + a*b^3*c^3*e^3*m^2 - 2*a^2*b*c^4*e^3*m^2 - 3*a^3*b*c^3*f^2*l
^3 + 3*a*b^4*c^2*f^3*m^2 - 5*a^2*b^4*c*f^2*m^3 - 6*a^2*c^5*d*e^2*l^2 - 3*a^
2*c^5*d*f^2*k^2 - 3*a^2*c^5*d*g^2*j^2 + 3*a^2*c^5*f*g^2*h^2 - a^3*b^2*c^2*h
*k^4 + 3*b^3*c^4*d^2*e*k^2 + 3*a^2*c^5*d^2*f*l^2 + 3*a^2*c^5*e^2*f*k^2 + a^
3*b*c^3*g^3*m^2 - 3*b^4*c^3*d*e^2*l^2 + 6*a^2*c^5*d^2*h*k^2 - 3*a^2*c^5*e^2
*h*j^2 + 3*b^2*c^5*d^2*e^2*m - 3*a^2*c^5*f^2*g^2*k - a^3*b^3*c*j^2*l^3 + 3*
a^3*c^4*d*g^2*m^2 + 2*a^4*b*c^2*j^2*l^3 - 3*a^2*c^5*d^2*h^2*m - 3*a^2*c^5*d
^2*j^2*k - 3*a^2*c^5*e^2*g^2*m + 3*a^3*b^2*c^2*j^4*m - 3*a^3*b^3*c*j^3*m^2
+ 3*a^3*c^4*d*j^2*k^2 + 3*a^3*c^4*f*g^2*l^2 + 3*a^3*c^4*f*h^2*k^2 + 3*a^4*b
^2*c*j^2*m^3 + 3*a^3*c^4*e^2*h*m^2 + 3*a^3*c^4*f^2*h*l^2 + 3*a^3*c^4*d^2*k*
m^2 + 6*a^3*c^4*e^2*k*l^2 - 3*a^3*c^4*g^2*h^2*m + 3*a^3*c^4*g^2*j^2*k - 3*a
^4*c^3*d*k^2*m^2 - 3*a^3*c^4*d^2*l^2*m - 3*a^3*c^4*e^2*k^2*m - 6*a^3*c^4*f^
2*j^2*m + 6*a^4*c^3*f*j^2*m^2 + 3*a^4*c^3*f*k^2*l^2 - 3*a^4*c^3*h*j^2*l^2 -
3*a^4*c^3*g^2*k*m^2 - 3*a^4*c^3*g^2*l^2*m - 3*a^4*c^3*h^2*k^2*m + 3*a^5*c^
2*h*l^2*m^2 - 3*a^5*c^2*k^2*l^2*m + 9*a*b^2*c^4*d*e^2*l^2 + 3*a*b^2*c^4*d*f
^2*k^2 - 9*a^2*b^2*c^3*d*e*l^3 + 10*a^2*b^3*c^2*d*e*m^3 - 3*a*b^2*c^4*d^2*h
*k^2 - 3*a*b^3*c^3*e*f^2*l^2 + 3*a*b^3*c^3*e*g^2*k^2 + 6*a^2*b*c^4*e*f^2*l^
2 - 3*a^2*b*c^4*e*g^2*k^2 + 3*a^2*b^2*c^3*d*h*k^3 + 3*a*b^2*c^4*f^2*g^2*k +
3*a*b^3*c^3*d^2*g*m^2 + 3*a*b^3*c^3*e^2*g*l^2 - 3*a*b^3*c^3*f^2*g*k^2 - 3*
a*b^4*c^2*d*h^2*l^2 - 6*a^2*b*c^4*d^2*g*m^2 - 6*a^2*b*c^4*e^2*g*l^2 + 6*a^2
*b*c^4*f^2*g*k^2 - a^2*b^3*c^2*d*h*l^3 - 4*a^2*b^3*c^2*e*g*l^3 + 3*a*b^2*c^
4*d^2*h^2*m + 3*a*b^2*c^4*e^2*g^2*m - 3*a^2*b*c^4*g^2*h^2*j - a^2*b^2*c^3*g
*h*j^3 - 6*a^3*b^2*c^2*d*h*m^3 - 6*a^3*b^2*c^2*e*g*m^3 - 3*a*b^3*c^3*f^2*h^
2*l + 3*a*b^4*c^2*f^2*h*l^2 - 3*a^2*b*c^4*d^2*j*l^2 - 3*a^2*b*c^4*e^2*j*k^2
+ 6*a^2*b*c^4*f^2*h^2*l - a^2*b^2*c^3*d*j^3*l - a^2*b^2*c^3*e*j^3*k + a^2*
b^3*c^2*g*h*k^3 + 3*a^3*b*c^3*e*h^2*m^2 - 3*a^2*b^2*c^3*g*h^3*l + a^2*b^3*c
^2*d*k^3*l - 2*a^2*b^3*c^2*f*j*k^3 - 3*a^3*b*c^3*e*j^2*l^2 - 3*a^3*b*c^3*g*
h^2*l^2 - 3*a^3*b*c^3*g*j^2*k^2 + 3*a^3*b^2*c^2*e*k*l^3 - 6*a^3*b^2*c^2*f*j
*l^3 + 3*a*b^4*c^2*f^2*j^2*m - 6*a^2*b^3*c^2*f*j^3*m + 6*a^2*b^4*c*f*j^2*m^
2 - 6*a^3*b*c^3*f^2*j*m^2 - 3*a^3*b*c^3*g^2*j*l^2 - 3*a^3*b*c^3*h^2*j*k^2 +
3*a^3*b*c^3*e^2*l*m^2 + 3*a^3*b*c^3*g^2*k^2*l - 3*a^3*b*c^3*h^2*j^2*l + 2*
a^3*b^2*c^2*f*k^3*m - a^3*b^2*c^2*g*k^3*l - 3*a^3*b^2*c^2*j^3*k*l - 3*a^4*b
*c^2*j*k^2*l^2 + 3*a^4*b^2*c*k^2*l^2*m + a*b*c^5*d*e*h^3 + a*b*c^5*d*g^3*h
+ 3*a*b*c^5*f^3*g*h - 3*b*c^6*d*e^2*f*g + 3*a*b*c^5*d*f^3*l + 3*a*b*c^5*e*f
^3*k - 6*a*b^5*c*d*e*m^3 - 3*b*c^6*d^2*e*f*h + 6*a*c^6*d*e*f^2*j + 2*a*b*c^
5*e^3*f*m + 2*a*b*c^5*e^3*g*l - a*b*c^5*e^3*h*k + a*b^5*c*d*h*l^3 + a*b^5*c

*e*g*l^3 - 6*a*c^6*d*e^2*f*k - 6*a*c^6*d^2*e*f*l + 6*a*c^6*d^2*e*g*k - 6*a*
c^6*d^2*f*g*j - 4*a*b*c^5*d^3*j*m + 2*a*b*c^5*d^3*k*l - 3*b^6*c*d*e*j*m^2 +
a^5*b*c*k*l*m^3 - 3*a^2*b^2*c^3*d*g^2*m^2 + 6*a^2*b^2*c^3*d*h^2*l^2 - 3*a^
2*b^2*c^3*e^2*h*m^2 - 9*a^2*b^2*c^3*f^2*h*l^2 - 3*a^2*b^2*c^3*g^2*h*k^2 + 3
*a^2*b^3*c^2*g*h^2*l^2 - 3*a^2*b^2*c^3*d^2*k*m^2 - 3*a^2*b^2*c^3*e^2*k*l^2
+ 3*a^2*b^2*c^3*g^2*h^2*m + 3*a^2*b^2*c^3*d^2*l^2*m + 3*a^2*b^2*c^3*e^2*k^2
*m + 3*a^2*b^2*c^3*f^2*j^2*m + 6*a^2*b^3*c^2*f^2*j*m^2 - 9*a^3*b^2*c^2*f*j^
2*m^2 + 3*a^3*b^2*c^2*h*j^2*l^2 - 3*a^3*b^2*c^2*h^2*k*l^2 + 3*a^3*b^2*c^2*g
^2*l^2*m + 3*a^3*b^2*c^2*h^2*k^2*m - 3*a*b*c^5*e*f^2*h^2 + 3*a*b^2*c^4*d*e*
j^3 - 6*a*b*c^5*d^2*e*k^2 + 3*a*b*c^5*e^2*g*h^2 - a*b^2*c^4*e*g*h^3 - 4*a*b
^3*c^3*d*e*k^3 + 6*a^2*b*c^4*d*e*k^3 + 3*a*b*c^5*d^2*g*j^2 + 5*a*b^4*c^2*d*
e*l^3 - 3*a*b*c^5*d^2*h^2*j - 3*a*b*c^5*e^2*g^2*j + a*b^3*c^3*d*h*j^3 + a*b
^3*c^3*e*g*j^3 - 2*a^2*b*c^4*d*h*j^3 - 2*a^2*b*c^4*e*g*j^3 - 7*a^3*b*c^3*d*
e*m^3 - 3*a*b*c^5*d^2*g^2*l - 3*a*b*c^5*e^2*f^2*l - 3*a*b^2*c^4*e*g^3*k - a
*b^4*c^2*d*h*k^3 - a*b^4*c^2*e*g*k^3 + 3*a*b^3*c^3*d*h^3*l - 5*a^2*b*c^4*d*
h^3*l + a^2*b*c^4*e*h^3*k - 2*a^2*b*c^4*f*h^3*j - 3*a^3*b*c^3*d*h*l^3 + 6*a
^3*b*c^3*e*g*l^3 - 3*b^2*c^5*d*e*f^2*j + 3*b^3*c^4*d*e*f*j^2 - 3*a*b^2*c^4*
f^3*g*l - 3*a*b^2*c^4*f^3*h*k + 5*a^2*b^4*c*d*h*m^3 + 5*a^2*b^4*c*e*g*m^3 -
6*a^2*c^5*d*e*g*k^2 + 3*b^2*c^5*d*e^2*f*k + 3*b^2*c^5*d*e^2*g*j + 3*a^2*b*
c^4*g^3*h*k + a^3*b*c^3*g*h*k^3 + 3*b^2*c^5*d^2*e*f*l - 6*b^2*c^5*d^2*e*g*k
+ 3*b^2*c^5*d^2*e*h*j + 3*b^3*c^4*d*e*g^2*k - 3*b^4*c^3*d*e*g*k^2 - a^2*b^
4*c*g*h*l^3 - 6*a^2*c^5*d*f*h^2*k + 6*a^2*c^5*e*f*h^2*j - 2*a^3*b*c^3*d*k^3
*l + 4*a^3*b*c^3*f*j*k^3 + 3*b^3*c^4*d*e*f^2*m + 3*b^5*c^2*d*e*f*m^2 - 2*a*
b^2*c^4*e^3*j*m + a*b^2*c^4*e^3*k*l - a^2*b^4*c*e*k*l^3 + 2*a^2*b^4*c*f*j*l
^3 - 6*a^2*c^5*d*f*g^2*m - 6*a^2*c^5*e*f*g^2*l - 4*a^3*b^3*c*g*h*m^3 + 3*a^
4*b*c^2*g*h*m^3 - 3*b^3*c^4*d*e^2*g*m + 6*b^3*c^4*d*e^2*h*l - 3*b^4*c^3*d*e
*h^2*l + 3*b^5*c^2*d*e*h*l^2 - 6*a*b^3*c^3*f^3*j*m + 3*a*b^3*c^3*f^3*k*l -
3*a*b^5*c*f^2*j*m^2 + 8*a^2*b*c^4*f^3*j*m - 7*a^2*b*c^4*f^3*k*l + 12*a^2*c^
5*d*f^2*h*m + 12*a^2*c^5*e*f^2*g*m - 6*a^2*c^5*e*f^2*h*l - 6*a^2*c^5*f^2*g*
h*j + 4*a^3*b*c^3*f*j^3*m + 4*a^3*b*c^3*g*j^3*l + 4*a^3*b*c^3*h*j^3*k - 4*a
^3*b^3*c*d*l*m^3 - 4*a^3*b^3*c*e*k*m^3 + 2*a^3*b^3*c*f*j*m^3 - 12*a^3*c^4*d
*f*h*m^2 - 12*a^3*c^4*e*f*g*m^2 + 3*a^4*b*c^2*d*l*m^3 + 3*a^4*b*c^2*e*k*m^3
- 3*b^3*c^4*d*e^2*j*k - 3*b^3*c^4*d^2*e*h*m - 6*a^2*c^5*d*f^2*j*l - 6*a^2*
c^5*e*f^2*j*k - 6*a^2*c^5*e^2*f*h*m + 6*a^2*c^5*e^2*g*h*l + 6*a^3*c^4*d*e*j
*m^2 - 6*a^3*c^4*e*g*h*l^2 - 3*b^3*c^4*d^2*e*j*l + 6*a^2*c^5*d*e^2*k*m + 6*
a^2*c^5*e^2*f*j*l + 3*a^3*b*c^3*h^3*k*l - 2*a^3*b^3*c*f*l^3*m + a^3*b^3*c*h
*k*l^3 - 6*a^3*c^4*d*f*k*l^2 - 6*a^3*c^4*e*f*j*l^2 + 4*a^4*b*c^2*f*l^3*m +
a^4*b*c^2*h*k*l^3 + 3*b^5*c^2*d*e*j^2*m + 6*a^2*c^5*d^2*e*l*m - 6*a^2*c^5*d
^2*f*k*m + 6*a^2*c^5*d^2*g*j*m - 6*a^2*c^5*d^2*g*k*l + 6*a^3*c^4*d*f*k^2*m
+ 6*a^3*c^4*d*g*k^2*l - 6*a^3*c^4*e*f*k^2*l - 6*a^3*c^4*f*g*j*k^2 + 3*a^4*b
^2*c*g*l*m^3 + 3*a^4*b^2*c*h*k*m^3 + 3*b^4*c^3*d*e^2*k*m + 6*a^3*c^4*e*h*j^
2*l + 3*b^4*c^3*d^2*e*l*m + 6*a^3*c^4*d*h^2*k*m - 6*a^3*c^4*e*h^2*j*m - 6*a
^3*c^4*f*h^2*j*l - 2*a^4*b*c^2*j*k^3*m + 6*a^3*c^4*e*g^2*l*m + 6*a^3*c^4*f*
g^2*k*m + 2*a^4*b^2*c*j*l^3*m - 12*a^3*c^4*f^2*g*l*m - 12*a^3*c^4*f^2*h*k*m
- 6*a^4*c^3*e*h*l*m^2 + 12*a^4*c^3*f*g*l*m^2 + 12*a^4*c^3*f*h*k*m^2 - 6*a^

4*c^3*g*h*j*m^2 + 6*a^3*c^4*f^2*j*k*l - 6*a^4*c^3*d*j*l*m^2 - 6*a^4*c^3*e*j
*k*m^2 - 6*a^4*c^3*f*h*l^2*m - 6*a^3*c^4*e^2*j*l*m + 6*a^4*c^3*d*k*l^2*m +
6*a^4*c^3*e*j*l^2*m + 6*a^4*c^3*e*k^2*l*m + 6*a^4*c^3*g*j*k^2*m + 6*a^4*c^3
*h^2*j*l*m + 6*a^5*c^2*j*k*l*m^2 + 6*a*b^2*c^4*d*e*g*k^2 - 3*a*b^2*c^4*d*f*
h*j^2 - 3*a*b^2*c^4*e*f*g*j^2 - 15*a*b^3*c^3*d*e*f*m^2 + 15*a^2*b*c^4*d*e*f
*m^2 + 3*a*b^2*c^4*d*e*h^2*l + 3*a*b^2*c^4*d*f*h^2*k + 3*a*b^2*c^4*d*g*h^2*
j - 9*a*b^3*c^3*d*e*h*l^2 + 9*a^2*b*c^4*d*e*h*l^2 - 3*a^2*b*c^4*d*f*g*l^2 -
3*a*b^2*c^4*d*g^2*h*k + 3*a*b^2*c^4*e*f*g^2*l + 3*a*b^2*c^4*e*g^2*h*j + 3*
a*b^3*c^3*d*g*h*k^2 - 3*a^2*b*c^4*d*g*h*k^2 - 3*a^2*b*c^4*e*f*h*k^2 - 6*a*b
^2*c^4*d*f^2*h*m - 6*a*b^2*c^4*e*f^2*g*m + 6*a*b^2*c^4*e*f^2*h*l - 3*a*b^2*
c^4*f^2*g*h*j - 3*a*b^4*c^2*d*f*h*m^2 - 3*a*b^4*c^2*e*f*g*m^2 - 6*a^2*b*c^4
*d*f*j*k^2 + 9*a^2*b*c^4*f*g*h*j^2 - 3*a*b^2*c^4*d*f^2*j*l - 3*a*b^2*c^4*e*
f^2*j*k - 6*a*b^2*c^4*e^2*g*h*l - 12*a*b^3*c^3*d*e*j^2*m - 3*a*b^3*c^3*d*g*
h^2*m + 3*a*b^3*c^3*e*g*h^2*l + 15*a*b^4*c^2*d*e*j*m^2 - 3*a*b^4*c^2*e*g*h*
l^2 + 3*a^2*b*c^4*d*e*j^2*m + 9*a^2*b*c^4*d*f*j^2*l + 3*a^2*b*c^4*d*g*h^2*m
+ 9*a^2*b*c^4*e*f*j^2*k - 3*a^2*b*c^4*f*g*h^2*k - 9*a*b^2*c^4*d*e^2*k*m +
3*a*b^2*c^4*e^2*g*j*k - 3*a*b^3*c^3*d*h^2*j*k - 3*a*b^3*c^3*e*g^2*h*m + 6*a
^2*b*c^4*d*h^2*j*k + 3*a^2*b*c^4*e*g^2*h*m - 3*a^2*b*c^4*f*g^2*h*l - 9*a*b^
2*c^4*d^2*e*l*m - 6*a*b^2*c^4*d^2*g*j*m + 3*a*b^2*c^4*d^2*g*k*l + 3*a*b^2*c
^4*d^2*h*j*l - 3*a*b^3*c^3*e*g^2*j*l + 3*a*b^3*c^3*f^2*g*h*m + 6*a^2*b*c^4*
d*g^2*j*m + 6*a^2*b*c^4*e*g^2*j*l - 6*a^2*b*c^4*f*g^2*j*k - 3*a^2*b*c^4*f^2
*g*h*m - 3*a^3*b*c^3*f*g*h*m^2 + 3*a*b^3*c^3*d*f^2*l*m + 3*a*b^3*c^3*e*f^2*
k*m + 3*a*b^3*c^3*f^2*g*j*l + 3*a*b^3*c^3*f^2*h*j*k - 3*a*b^4*c^2*d*h*j^2*m
- 3*a*b^4*c^2*e*g*j^2*m - 3*a^2*b*c^4*d*f^2*l*m - 3*a^2*b*c^4*e*f^2*k*m -
3*a^3*b*c^3*d*f*l*m^2 + 6*a^3*b*c^3*d*g*k*m^2 + 6*a^3*b*c^3*d*h*j*m^2 - 3*a
^3*b*c^3*e*f*k*m^2 + 6*a^3*b*c^3*e*g*j*m^2 - 3*a*b^3*c^3*e^2*g*k*m + 3*a*b^
4*c^2*d*h^2*k*m + 3*a^2*b*c^4*e^2*g*k*m + 6*a^2*b*c^4*e^2*h*j*m + 3*a^2*b*c
^4*e^2*h*k*l + 3*a^3*b*c^3*d*g*l^2*m - 6*a^3*b*c^3*e*f*l^2*m - 3*a^3*b*c^3*
e*h*k*l^2 - 3*a^3*b*c^3*f*g*k*l^2 + 12*a^3*b*c^3*f*h*j*l^2 - 3*a*b^3*c^3*d^
2*h*l*m + 3*a*b^4*c^2*e*g^2*l*m + 3*a^2*b*c^4*d^2*h*l*m + 6*a^3*b*c^3*d*j*k
*l^2 + 3*a^3*b*c^3*e*h*k^2*m - 6*a^3*b*c^3*f*g*k^2*m - 3*a^3*b*c^3*f*h*k^2*
l - 3*a*b^4*c^2*f^2*g*l*m - 3*a*b^4*c^2*f^2*h*k*m + 6*a^2*b*c^4*d^2*j*k*m -
3*a^2*b^4*c*g*h*j*m^2 + 6*a^3*b*c^3*e*j*k^2*l - 3*a^3*b*c^3*g*h*j^2*m - 3*
a*b^4*c^2*f^2*j*k*l - 3*a^2*b^4*c*d*j*l*m^2 - 3*a^2*b^4*c*e*j*k*m^2 - 3*a^3
*b*c^3*d*j^2*l*m - 3*a^3*b*c^3*e*j^2*k*m - 6*a^3*b*c^3*f*h^2*l*m - 9*a^3*b*
c^3*f*j^2*k*l + 3*a^3*b*c^3*g*h^2*k*m + 3*a^2*b^4*c*d*k*l^2*m + 3*a^3*b*c^3
*g^2*h*l*m + 3*a^2*b^4*c*e*k^2*l*m + 15*a^3*b*c^3*f^2*k*l*m + 6*a^3*b^3*c*f
*k*l*m^2 + 3*a^3*b^3*c*g*j*l*m^2 + 3*a^3*b^3*c*h*j*k*m^2 - 9*a^4*b*c^2*f*k*
l*m^2 - 3*a^3*b^3*c*g*k*l^2*m + 3*a^4*b*c^2*g*k*l^2*m - 6*a^4*b*c^2*h*j*l^2
*m - 3*a^3*b^3*c*h*k^2*l*m + 3*a^4*b*c^2*h*k^2*l*m + 3*a^3*b^3*c*j^2*k*l*m
+ 3*a^4*b*c^2*j^2*k*l*m - 9*a^4*b^2*c*j*k*l*m^2 + 3*b^6*c*d*e*k*l*m + 12*a^
2*b^2*c^3*d*f*h*m^2 + 12*a^2*b^2*c^3*e*f*g*m^2 - 15*a^2*b^2*c^3*d*e*j*m^2 +
6*a^2*b^2*c^3*e*g*h*l^2 + 3*a^2*b^2*c^3*d*f*k*l^2 + 3*a^2*b^2*c^3*d*g*j*l^
2 + 6*a^2*b^2*c^3*e*f*j*l^2 - 3*a^2*b^2*c^3*d*g*k^2*l + 3*a^2*b^2*c^3*e*f*k
^2*l + 3*a^2*b^2*c^3*e*h*j*k^2 + 6*a^2*b^2*c^3*f*g*j*k^2 + 3*a^2*b^3*c^2*f*

g*h*m^2 + 9*a^2*b^2*c^3*d*h*j^2*m + 9*a^2*b^2*c^3*e*g*j^2*m - 6*a^2*b^2*c^3
*f*g*j^2*l - 6*a^2*b^2*c^3*f*h*j^2*k + 3*a^2*b^3*c^2*d*f*l*m^2 - 12*a^2*b^3
*c^2*d*h*j*m^2 + 3*a^2*b^3*c^2*e*f*k*m^2 - 12*a^2*b^3*c^2*e*g*j*m^2 - 9*a^2
*b^2*c^3*d*h^2*k*m - 3*a^2*b^2*c^3*e*h^2*k*l + 6*a^2*b^2*c^3*f*h^2*j*l + 3*
a^2*b^2*c^3*g*h^2*j*k - 3*a^2*b^3*c^2*d*g*l^2*m + 3*a^2*b^3*c^2*e*h*k*l^2 -
6*a^2*b^3*c^2*f*h*j*l^2 - 9*a^2*b^2*c^3*e*g^2*l*m + 3*a^2*b^2*c^3*g^2*h*j*
l - 3*a^2*b^3*c^2*d*j*k*l^2 - 3*a^2*b^3*c^2*e*h*k^2*m + 9*a^2*b^2*c^3*f^2*g
*l*m + 9*a^2*b^2*c^3*f^2*h*k*m - 3*a^2*b^3*c^2*e*j*k^2*l + 3*a^2*b^3*c^2*g*
h*j^2*m - 9*a^3*b^2*c^2*f*g*l*m^2 - 9*a^3*b^2*c^2*f*h*k*m^2 + 9*a^3*b^2*c^2
*g*h*j*m^2 + 3*a^2*b^2*c^3*f^2*j*k*l + 3*a^2*b^3*c^2*d*j^2*l*m + 3*a^2*b^3*
c^2*e*j^2*k*m + 6*a^2*b^3*c^2*f*j^2*k*l - 3*a^2*b^3*c^2*g*h^2*k*m + 9*a^3*b
^2*c^2*d*j*l*m^2 + 9*a^3*b^2*c^2*e*j*k*m^2 + 6*a^3*b^2*c^2*f*h*l^2*m - 3*a^
2*b^3*c^2*g^2*h*l*m - 9*a^3*b^2*c^2*d*k*l^2*m + 3*a^3*b^2*c^2*g*j*k*l^2 - 9
*a^3*b^2*c^2*e*k^2*l*m + 3*a^3*b^2*c^2*h*j*k^2*l - 12*a^2*b^3*c^2*f^2*k*l*m
- 6*a^3*b^2*c^2*g*j^2*l*m - 6*a^3*b^2*c^2*h*j^2*k*m - 9*a*b*c^5*d*e*f*j^2
- 3*a*b*c^5*d*f*g*h^2 - 3*a*b*c^5*e*f*g^2*h - 9*a*b*c^5*d*e*f^2*m - 6*a*b*c
^5*d*f^2*g*k + 6*a*b*c^5*d*f^2*h*j + 6*a*b*c^5*e*f^2*g*j + 3*a*b*c^5*d*e^2*
g*m - 9*a*b*c^5*d*e^2*h*l - 3*a*b*c^5*e^2*f*g*k + 3*b^2*c^5*d*e*f*g*h + 6*a
*b*c^5*d*e^2*j*k + 3*a*b*c^5*d^2*e*h*m + 6*a*b*c^5*d^2*f*g*m - 3*a*b*c^5*d^
2*f*h*l + 3*a*b*c^5*d^2*g*h*k + 6*a*b*c^5*d^2*e*j*l - 3*b^3*c^4*d*e*f*g*l -
3*b^3*c^4*d*e*f*h*k - 3*b^3*c^4*d*e*g*h*j + 3*a*b^5*c*d*h*j*m^2 + 3*a*b^5*
c*e*g*j*m^2 - 12*a^2*c^5*d*e*f*j*m + 12*a^2*c^5*d*e*f*k*l + 12*a^2*c^5*d*f*
g*j*k + 3*b^4*c^3*d*e*g*h*m - 6*b^4*c^3*d*e*f*j*m + 3*b^4*c^3*d*e*f*k*l + 3
*b^4*c^3*d*e*g*j*l + 3*b^4*c^3*d*e*h*j*k - 3*b^5*c^2*d*e*g*l*m - 3*b^5*c^2*
d*e*h*k*m - 3*b^5*c^2*d*e*j*k*l + 3*a*b^5*c*f^2*k*l*m + 12*a^3*c^4*e*f*h*l*
m + 12*a^3*c^4*f*g*h*j*m - 12*a^3*c^4*d*e*k*l*m + 12*a^3*c^4*d*f*j*l*m - 12
*a^3*c^4*d*g*j*k*m + 12*a^3*c^4*e*f*j*k*m - 12*a^4*c^3*f*j*k*l*m - 3*a*b^2*
c^4*d*e*g*h*m + 3*a*b^2*c^4*d*f*g*h*l + 3*a*b^2*c^4*e*f*g*h*k + 24*a*b^2*c^
4*d*e*f*j*m - 12*a*b^2*c^4*d*e*f*k*l - 6*a*b^2*c^4*d*e*g*j*l - 6*a*b^2*c^4*
d*e*h*j*k + 9*a*b^3*c^3*d*e*g*l*m + 9*a*b^3*c^3*d*e*h*k*m + 6*a*b^3*c^3*d*f
*h*j*m - 3*a*b^3*c^3*d*f*h*k*l - 3*a*b^3*c^3*d*g*h*j*l + 6*a*b^3*c^3*e*f*g*
j*m - 3*a*b^3*c^3*e*f*g*k*l - 3*a*b^3*c^3*e*g*h*j*k - 6*a^2*b*c^4*d*e*g*l*m
- 6*a^2*b*c^4*d*e*h*k*m - 12*a^2*b*c^4*d*f*h*j*m + 6*a^2*b*c^4*d*f*h*k*l -
12*a^2*b*c^4*e*f*g*j*m + 6*a^2*b*c^4*e*f*g*k*l - 12*a^2*b*c^4*e*f*h*j*l +
12*a*b^3*c^3*d*e*j*k*l - 12*a^2*b*c^4*d*e*j*k*l + 3*a*b^4*c^2*d*g*h*l*m + 3
*a*b^4*c^2*e*g*h*k*m - 15*a*b^4*c^2*d*e*k*l*m + 3*a*b^4*c^2*d*h*j*k*l + 3*a
*b^4*c^2*e*g*j*k*l - 6*a^3*b*c^3*d*h*k*l*m - 6*a^3*b*c^3*e*g*k*l*m + 3*a^2*
b^4*c*g*h*k*l*m - 6*a^2*b^4*c*f*j*k*l*m - 3*a^2*b^2*c^3*d*g*h*l*m - 3*a^2*b
^2*c^3*e*g*h*k*m - 12*a^2*b^2*c^3*f*g*h*j*m + 3*a^2*b^2*c^3*f*g*h*k*l + 24*
a^2*b^2*c^3*d*e*k*l*m - 12*a^2*b^2*c^3*d*f*j*l*m - 6*a^2*b^2*c^3*d*h*j*k*l
- 12*a^2*b^2*c^3*e*f*j*k*m - 6*a^2*b^2*c^3*e*g*j*k*l + 9*a^2*b^3*c^2*d*h*k*
l*m + 9*a^2*b^3*c^2*e*g*k*l*m + 6*a^2*b^3*c^2*f*g*j*l*m + 6*a^2*b^3*c^2*f*h
*j*k*m - 3*a^2*b^3*c^2*g*h*j*k*l - 3*a^3*b^2*c^2*g*h*k*l*m + 12*a^3*b^2*c^2
*f*j*k*l*m + 6*a*b*c^5*d*e*f*g*l + 6*a*b*c^5*d*e*f*h*k - 3*a*b^5*c*d*h*k*l*
m - 3*a*b^5*c*e*g*k*l*m)/c^3 - root(34992*a^4*b^2*c^8*z^6 - 8748*a^3*b^4*c^

7*z^6 + 729*a^2*b^6*c^6*z^6 - 46656*a^5*c^9*z^6 + 34992*a^4*b^3*c^6*m*z^5 -
 8748*a^3*b^5*c^5*m*z^5 + 729*a^2*b^7*c^4*m*z^5 - 34992*a^4*b^2*c^7*j*z^5 +
 8748*a^3*b^4*c^6*j*z^5 - 729*a^2*b^6*c^5*j*z^5 - 46656*a^5*b*c^7*m*z^5 + 4
6656*a^5*c^8*j*z^5 + 34992*a^5*b*c^6*j*m*z^4 - 11664*a^5*b*c^6*k*l*z^4 + 38
88*a^4*b*c^7*f*j*z^4 + 3888*a^4*b*c^7*e*k*z^4 + 3888*a^4*b*c^7*d*l*z^4 + 38
88*a^4*b*c^7*g*h*z^4 + 3888*a^3*b*c^8*d*e*z^4 + 243*a*b^5*c^6*d*e*z^4 - 252
72*a^4*b^3*c^5*j*m*z^4 + 9720*a^4*b^3*c^5*k*l*z^4 + 6075*a^3*b^5*c^4*j*m*z^
4 - 2673*a^3*b^5*c^4*k*l*z^4 - 486*a^2*b^7*c^3*j*m*z^4 + 243*a^2*b^7*c^3*k*
l*z^4 - 7776*a^4*b^2*c^6*h*k*z^4 - 7776*a^4*b^2*c^6*g*l*z^4 - 7776*a^4*b^2*
c^6*f*m*z^4 + 2430*a^3*b^4*c^5*h*k*z^4 + 2430*a^3*b^4*c^5*g*l*z^4 + 2430*a^
3*b^4*c^5*f*m*z^4 - 243*a^2*b^6*c^4*h*k*z^4 - 243*a^2*b^6*c^4*g*l*z^4 - 243
*a^2*b^6*c^4*f*m*z^4 - 1944*a^3*b^3*c^6*f*j*z^4 - 1944*a^3*b^3*c^6*e*k*z^4
- 1944*a^3*b^3*c^6*d*l*z^4 + 243*a^2*b^5*c^5*f*j*z^4 + 243*a^2*b^5*c^5*e*k*
z^4 + 243*a^2*b^5*c^5*d*l*z^4 - 1944*a^3*b^3*c^6*g*h*z^4 + 243*a^2*b^5*c^5*
g*h*z^4 + 3888*a^3*b^2*c^7*e*g*z^4 + 3888*a^3*b^2*c^7*d*h*z^4 - 486*a^2*b^4
*c^6*e*g*z^4 - 486*a^2*b^4*c^6*d*h*z^4 - 1944*a^2*b^3*c^7*d*e*z^4 + 7776*a^
5*c^7*h*k*z^4 + 7776*a^5*c^7*g*l*z^4 + 7776*a^5*c^7*f*m*z^4 - 7776*a^4*c^8*
e*g*z^4 - 7776*a^4*c^8*d*h*z^4 - 13608*a^5*b^2*c^5*m^2*z^4 + 11421*a^4*b^4*
c^4*m^2*z^4 - 2916*a^3*b^6*c^3*m^2*z^4 + 243*a^2*b^8*c^2*m^2*z^4 + 13608*a^
4*b^2*c^6*j^2*z^4 - 3159*a^3*b^4*c^5*j^2*z^4 + 243*a^2*b^6*c^4*j^2*z^4 + 19
44*a^3*b^2*c^7*f^2*z^4 - 243*a^2*b^4*c^6*f^2*z^4 - 3888*a^6*c^6*m^2*z^4 - 1
9440*a^5*c^7*j^2*z^4 - 3888*a^4*c^8*f^2*z^4 + 3078*a^4*b^4*c^3*k*l*m*z^3 -
2592*a^5*b^2*c^4*k*l*m*z^3 - 891*a^3*b^6*c^2*k*l*m*z^3 - 4536*a^4*b^3*c^4*j
*k*l*z^3 + 1053*a^3*b^5*c^3*j*k*l*z^3 - 81*a^2*b^7*c^2*j*k*l*z^3 - 2592*a^4
*b^3*c^4*h*k*m*z^3 - 2592*a^4*b^3*c^4*g*l*m*z^3 + 810*a^3*b^5*c^3*h*k*m*z^3
 + 810*a^3*b^5*c^3*g*l*m*z^3 - 81*a^2*b^7*c^2*h*k*m*z^3 - 81*a^2*b^7*c^2*g*
l*m*z^3 + 7776*a^4*b^2*c^5*f*j*m*z^3 + 3888*a^4*b^2*c^5*h*j*k*z^3 + 3888*a^
4*b^2*c^5*g*j*l*z^3 - 3888*a^4*b^2*c^5*f*k*l*z^3 - 2916*a^3*b^4*c^4*f*j*m*z
^3 + 1458*a^3*b^4*c^4*f*k*l*z^3 - 972*a^3*b^4*c^4*h*j*k*z^3 - 972*a^3*b^4*c
^4*g*j*l*z^3 - 486*a^3*b^4*c^4*e*k*m*z^3 - 486*a^3*b^4*c^4*d*l*m*z^3 + 324*
a^2*b^6*c^3*f*j*m*z^3 - 162*a^2*b^6*c^3*f*k*l*z^3 + 81*a^2*b^6*c^3*h*j*k*z^
3 + 81*a^2*b^6*c^3*g*j*l*z^3 + 81*a^2*b^6*c^3*e*k*m*z^3 + 81*a^2*b^6*c^3*d*
l*m*z^3 - 486*a^3*b^4*c^4*g*h*m*z^3 + 81*a^2*b^6*c^3*g*h*m*z^3 + 648*a^3*b^
3*c^5*e*j*k*z^3 + 648*a^3*b^3*c^5*d*j*l*z^3 - 81*a^2*b^5*c^4*e*j*k*z^3 - 81
*a^2*b^5*c^4*d*j*l*z^3 + 2592*a^3*b^3*c^5*e*g*m*z^3 + 2592*a^3*b^3*c^5*d*h*
m*z^3 - 1296*a^3*b^3*c^5*f*h*k*z^3 - 1296*a^3*b^3*c^5*f*g*l*z^3 - 1296*a^3*
b^3*c^5*e*h*l*z^3 + 648*a^3*b^3*c^5*g*h*j*z^3 - 324*a^2*b^5*c^4*e*g*m*z^3 -
 324*a^2*b^5*c^4*d*h*m*z^3 + 162*a^2*b^5*c^4*f*h*k*z^3 + 162*a^2*b^5*c^4*f*
g*l*z^3 + 162*a^2*b^5*c^4*e*h*l*z^3 - 81*a^2*b^5*c^4*g*h*j*z^3 + 5184*a^3*b
^2*c^6*d*e*m*z^3 - 2592*a^3*b^2*c^6*e*g*j*z^3 - 2592*a^3*b^2*c^6*d*h*j*z^3
- 2106*a^2*b^4*c^5*d*e*m*z^3 + 1296*a^3*b^2*c^6*e*f*k*z^3 + 1296*a^3*b^2*c^
6*d*g*k*z^3 + 1296*a^3*b^2*c^6*d*f*l*z^3 + 324*a^2*b^4*c^5*e*g*j*z^3 + 324*
a^2*b^4*c^5*d*h*j*z^3 - 162*a^2*b^4*c^5*e*f*k*z^3 - 162*a^2*b^4*c^5*d*g*k*z
^3 - 162*a^2*b^4*c^5*d*f*l*z^3 + 1296*a^3*b^2*c^6*f*g*h*z^3 - 162*a^2*b^4*c
^5*f*g*h*z^3 + 1944*a^2*b^3*c^6*d*e*j*z^3 - 1296*a^2*b^2*c^7*d*e*f*z^3 + 81

*a^2*b^8*c*k*l*m*z^3 + 6480*a^5*b*c^5*j*k*l*z^3 + 2592*a^5*b*c^5*h*k*m*z^3
+ 2592*a^5*b*c^5*g*l*m*z^3 - 1296*a^4*b*c^6*e*j*k*z^3 - 1296*a^4*b*c^6*d*j*
l*z^3 - 5184*a^4*b*c^6*e*g*m*z^3 - 5184*a^4*b*c^6*d*h*m*z^3 + 2592*a^4*b*c^
6*f*h*k*z^3 + 2592*a^4*b*c^6*f*g*l*z^3 + 2592*a^4*b*c^6*e*h*l*z^3 - 1296*a^
4*b*c^6*g*h*j*z^3 + 243*a*b^6*c^4*d*e*m*z^3 - 3888*a^3*b*c^7*d*e*j*z^3 - 24
3*a*b^5*c^5*d*e*j*z^3 + 162*a*b^4*c^6*d*e*f*z^3 - 2592*a^6*c^5*k*l*m*z^3 -
5184*a^5*c^6*h*j*k*z^3 - 5184*a^5*c^6*g*j*l*z^3 - 5184*a^5*c^6*f*j*m*z^3 +
2592*a^5*c^6*f*k*l*z^3 + 2592*a^5*c^6*e*k*m*z^3 + 2592*a^5*c^6*d*l*m*z^3 +
2592*a^5*c^6*g*h*m*z^3 + 5184*a^4*c^7*e*g*j*z^3 + 5184*a^4*c^7*d*h*j*z^3 -
2592*a^4*c^7*e*f*k*z^3 - 2592*a^4*c^7*d*g*k*z^3 - 2592*a^4*c^7*d*f*l*z^3 -
2592*a^4*c^7*d*e*m*z^3 - 2592*a^4*c^7*f*g*h*z^3 + 2592*a^3*c^8*d*e*f*z^3 +
6480*a^5*b^2*c^4*j*m^2*z^3 + 6480*a^4*b^3*c^4*j^2*m*z^3 - 5022*a^4*b^4*c^3*
j*m^2*z^3 - 1296*a^3*b^5*c^3*j^2*m*z^3 + 1134*a^3*b^6*c^2*j*m^2*z^3 + 81*a^
2*b^7*c^2*j^2*m*z^3 + 2592*a^4*b^3*c^4*h*l^2*z^3 - 1944*a^4*b^2*c^5*h^2*l*z
^3 - 810*a^3*b^5*c^3*h*l^2*z^3 + 729*a^3*b^4*c^4*h^2*l*z^3 + 81*a^2*b^7*c^2
*h*l^2*z^3 - 81*a^2*b^6*c^3*h^2*l*z^3 - 5184*a^4*b^3*c^4*f*m^2*z^3 + 1620*a
^3*b^5*c^3*f*m^2*z^3 + 1296*a^3*b^3*c^5*f^2*m*z^3 - 162*a^2*b^7*c^2*f*m^2*z
^3 - 162*a^2*b^5*c^4*f^2*m*z^3 - 1944*a^4*b^2*c^5*g*k^2*z^3 + 729*a^3*b^4*c
^4*g*k^2*z^3 - 648*a^3*b^3*c^5*g^2*k*z^3 - 81*a^2*b^6*c^3*g*k^2*z^3 + 81*a^
2*b^5*c^4*g^2*k*z^3 - 1944*a^4*b^2*c^5*e*l^2*z^3 + 729*a^3*b^4*c^4*e*l^2*z^
3 + 648*a^3*b^2*c^6*e^2*l*z^3 - 81*a^2*b^6*c^3*e*l^2*z^3 - 81*a^2*b^4*c^5*e
^2*l*z^3 + 1296*a^3*b^3*c^5*f*j^2*z^3 - 1296*a^3*b^2*c^6*f^2*j*z^3 - 162*a^
2*b^5*c^4*f*j^2*z^3 + 162*a^2*b^4*c^5*f^2*j*z^3 - 648*a^3*b^3*c^5*d*k^2*z^3
 + 81*a^2*b^5*c^4*d*k^2*z^3 + 648*a^3*b^2*c^6*e*h^2*z^3 - 81*a^2*b^4*c^5*e*
h^2*z^3 - 648*a^2*b^2*c^7*d^2*g*z^3 - 10368*a^5*b*c^5*j^2*m*z^3 - 81*a^2*b^
8*c*j*m^2*z^3 - 2592*a^5*b*c^5*h*l^2*z^3 + 5184*a^5*b*c^5*f*m^2*z^3 - 2592*
a^4*b*c^6*f^2*m*z^3 + 1296*a^4*b*c^6*g^2*k*z^3 - 2592*a^4*b*c^6*f*j^2*z^3 +
 1296*a^4*b*c^6*d*k^2*z^3 + 81*a*b^4*c^6*d^2*g*z^3 + 2592*a^6*c^5*j*m^2*z^3
 + 1296*a^5*c^6*h^2*l*z^3 + 1296*a^5*c^6*g*k^2*z^3 + 1296*a^5*c^6*e*l^2*z^3
 - 1296*a^4*c^7*e^2*l*z^3 + 2592*a^4*c^7*f^2*j*z^3 - 2592*a^6*b*c^4*m^3*z^3
 - 324*a^3*b^7*c*m^3*z^3 - 27*a^2*b^8*c*l^3*z^3 - 1296*a^4*c^7*e*h^2*z^3 -
864*a^5*b*c^5*k^3*z^3 + 1296*a^3*c^8*d^2*g*z^3 + 432*a^4*b*c^6*h^3*z^3 + 27
*a*b^4*c^6*e^3*z^3 - 432*a^2*b*c^8*d^3*z^3 + 216*a*b^3*c^7*d^3*z^3 + 1134*a
^4*b^5*c^2*m^3*z^3 - 432*a^5*b^3*c^3*m^3*z^3 + 1512*a^5*b^2*c^4*l^3*z^3 - 1
107*a^4*b^4*c^3*l^3*z^3 + 297*a^3*b^6*c^2*l^3*z^3 + 864*a^4*b^3*c^4*k^3*z^3
 - 270*a^3*b^5*c^3*k^3*z^3 + 27*a^2*b^7*c^2*k^3*z^3 - 2592*a^4*b^2*c^5*j^3*
z^3 + 486*a^3*b^4*c^4*j^3*z^3 - 27*a^2*b^6*c^3*j^3*z^3 - 216*a^3*b^3*c^5*h^
3*z^3 + 27*a^2*b^5*c^4*h^3*z^3 + 216*a^3*b^2*c^6*g^3*z^3 - 27*a^2*b^4*c^5*g
^3*z^3 - 216*a^2*b^2*c^7*e^3*z^3 - 432*a^6*c^5*l^3*z^3 + 27*a^2*b^9*m^3*z^3
 + 4320*a^5*c^6*j^3*z^3 - 432*a^4*c^7*g^3*z^3 + 432*a^3*c^8*e^3*z^3 - 27*b^
5*c^6*d^3*z^3 + 81*a^3*b^6*c*j*k*l*m*z^2 - 1296*a^5*b*c^4*h*j*k*m*z^2 - 129
6*a^5*b*c^4*g*j*l*m*z^2 + 1296*a^5*b*c^4*f*k*l*m*z^2 - 81*a^2*b^7*c*f*k*l*m
*z^2 + 2592*a^4*b*c^5*e*g*j*m*z^2 + 2592*a^4*b*c^5*d*h*j*m*z^2 - 1296*a^4*b
*c^5*f*h*j*k*z^2 - 1296*a^4*b*c^5*f*g*j*l*z^2 - 1296*a^4*b*c^5*e*f*k*m*z^2
- 1296*a^4*b*c^5*d*f*l*m*z^2 - 648*a^4*b*c^5*e*h*j*l*z^2 - 648*a^4*b*c^5*e*

g*k*l*z^2 - 648*a^4*b*c^5*d*h*k*l*z^2 - 648*a^4*b*c^5*d*g*k*m*z^2 - 1296*a^
4*b*c^5*f*g*h*m*z^2 - 162*a*b^6*c^3*d*e*j*m*z^2 + 81*a*b^6*c^3*d*e*k*l*z^2
+ 1296*a^3*b*c^6*d*e*f*m*z^2 - 648*a^3*b*c^6*d*f*g*k*z^2 - 648*a^3*b*c^6*d*
e*h*k*z^2 - 648*a^3*b*c^6*d*e*g*l*z^2 - 81*a*b^5*c^4*d*e*h*k*z^2 - 81*a*b^5
*c^4*d*e*g*l*z^2 + 81*a*b^5*c^4*d*e*f*m*z^2 - 81*a*b^4*c^5*d*e*f*j*z^2 + 81
*a*b^4*c^5*d*e*g*h*z^2 + 648*a^5*b^2*c^3*j*k*l*m*z^2 - 567*a^4*b^4*c^2*j*k*
l*m*z^2 - 1944*a^4*b^3*c^3*f*k*l*m*z^2 + 729*a^3*b^5*c^2*f*k*l*m*z^2 + 648*
a^4*b^3*c^3*h*j*k*m*z^2 + 648*a^4*b^3*c^3*g*j*l*m*z^2 - 81*a^3*b^5*c^2*h*j*
k*m*z^2 - 81*a^3*b^5*c^2*g*j*l*m*z^2 + 1944*a^4*b^2*c^4*f*j*k*l*z^2 - 729*a
^3*b^4*c^3*f*j*k*l*z^2 + 648*a^4*b^2*c^4*e*j*k*m*z^2 + 648*a^4*b^2*c^4*d*j*
l*m*z^2 - 81*a^3*b^4*c^3*e*j*k*m*z^2 - 81*a^3*b^4*c^3*d*j*l*m*z^2 + 81*a^2*
b^6*c^2*f*j*k*l*z^2 + 1296*a^4*b^2*c^4*f*h*k*m*z^2 + 1296*a^4*b^2*c^4*f*g*l
*m*z^2 + 648*a^4*b^2*c^4*g*h*j*m*z^2 - 648*a^3*b^4*c^3*f*h*k*m*z^2 - 648*a^
3*b^4*c^3*f*g*l*m*z^2 - 324*a^4*b^2*c^4*g*h*k*l*z^2 - 324*a^4*b^2*c^4*e*h*l
*m*z^2 + 81*a^3*b^4*c^3*g*h*k*l*z^2 - 81*a^3*b^4*c^3*g*h*j*m*z^2 + 81*a^2*b
^6*c^2*f*h*k*m*z^2 + 81*a^2*b^6*c^2*f*g*l*m*z^2 - 1296*a^3*b^3*c^4*e*g*j*m*
z^2 - 1296*a^3*b^3*c^4*d*h*j*m*z^2 + 648*a^3*b^3*c^4*f*h*j*k*z^2 + 648*a^3*
b^3*c^4*f*g*j*l*z^2 + 648*a^3*b^3*c^4*e*f*k*m*z^2 + 648*a^3*b^3*c^4*d*f*l*m
*z^2 + 486*a^3*b^3*c^4*e*g*k*l*z^2 + 486*a^3*b^3*c^4*d*h*k*l*z^2 + 162*a^3*
b^3*c^4*e*h*j*l*z^2 + 162*a^3*b^3*c^4*d*g*k*m*z^2 + 162*a^2*b^5*c^3*e*g*j*m
*z^2 + 162*a^2*b^5*c^3*d*h*j*m*z^2 - 81*a^2*b^5*c^3*f*h*j*k*z^2 - 81*a^2*b^
5*c^3*f*g*j*l*z^2 - 81*a^2*b^5*c^3*e*g*k*l*z^2 - 81*a^2*b^5*c^3*e*f*k*m*z^2
 - 81*a^2*b^5*c^3*d*h*k*l*z^2 - 81*a^2*b^5*c^3*d*f*l*m*z^2 + 648*a^3*b^3*c^
4*f*g*h*m*z^2 - 81*a^2*b^5*c^3*f*g*h*m*z^2 - 3240*a^3*b^2*c^5*d*e*j*m*z^2 +
 1620*a^3*b^2*c^5*d*e*k*l*z^2 + 1377*a^2*b^4*c^4*d*e*j*m*z^2 - 648*a^3*b^2*
c^5*e*f*j*k*z^2 - 648*a^3*b^2*c^5*d*f*j*l*z^2 - 648*a^2*b^4*c^4*d*e*k*l*z^2
 - 324*a^3*b^2*c^5*d*g*j*k*z^2 + 81*a^2*b^4*c^4*e*f*j*k*z^2 + 81*a^2*b^4*c^
4*d*f*j*l*z^2 + 972*a^3*b^2*c^5*e*f*h*l*z^2 - 648*a^3*b^2*c^5*f*g*h*j*z^2 -
 324*a^3*b^2*c^5*e*g*h*k*z^2 - 324*a^3*b^2*c^5*d*g*h*l*z^2 - 162*a^2*b^4*c^
4*e*f*h*l*z^2 + 81*a^2*b^4*c^4*f*g*h*j*z^2 + 81*a^2*b^4*c^4*e*g*h*k*z^2 + 8
1*a^2*b^4*c^4*d*g*h*l*z^2 - 648*a^2*b^3*c^5*d*e*f*m*z^2 + 486*a^2*b^3*c^5*d
*e*h*k*z^2 + 486*a^2*b^3*c^5*d*e*g*l*z^2 + 162*a^2*b^3*c^5*d*f*g*k*z^2 + 64
8*a^2*b^2*c^6*d*e*f*j*z^2 - 324*a^2*b^2*c^6*d*e*g*h*z^2 - 1296*a^6*b*c^3*k*
l*m^2*z^2 - 81*a^4*b^5*c*k*l*m^2*z^2 - 1296*a^5*b*c^4*j^2*k*l*z^2 - 324*a^5
*b*c^4*h^2*l*m*z^2 + 324*a^5*b*c^4*h*k^2*l*z^2 - 324*a^5*b*c^4*g*k^2*m*z^2
+ 972*a^5*b*c^4*h*j*l^2*z^2 + 324*a^5*b*c^4*g*k*l^2*z^2 - 324*a^5*b*c^4*e*l
^2*m*z^2 - 324*a^4*b*c^5*e^2*l*m*z^2 - 1944*a^5*b*c^4*f*j*m^2*z^2 + 1296*a^
5*b*c^4*e*k*m^2*z^2 + 1296*a^5*b*c^4*d*l*m^2*z^2 + 648*a^4*b*c^5*f^2*j*m*z^
2 + 81*a^2*b^7*c*f*j*m^2*z^2 + 1296*a^5*b*c^4*g*h*m^2*z^2 - 324*a^4*b*c^5*g
^2*j*k*z^2 + 324*a^4*b*c^5*g^2*h*l*z^2 + 972*a^4*b*c^5*f*h^2*l*z^2 + 324*a^
4*b*c^5*g*h^2*k*z^2 - 324*a^4*b*c^5*e*h^2*m*z^2 - 324*a^4*b*c^5*d*j*k^2*z^2
 - 324*a^3*b*c^6*d^2*j*k*z^2 + 972*a^4*b*c^5*f*g*k^2*z^2 + 972*a^3*b*c^6*d^
2*g*m*z^2 + 324*a^4*b*c^5*e*h*k^2*z^2 + 324*a^3*b*c^6*d^2*h*l*z^2 + 81*a*b^
5*c^4*d^2*g*m*z^2 + 972*a^4*b*c^5*e*f*l^2*z^2 + 324*a^4*b*c^5*d*g*l^2*z^2 -
 324*a^3*b*c^6*e^2*h*j*z^2 + 324*a^3*b*c^6*e^2*g*k*z^2 - 324*a^3*b*c^6*e^2*

f*l*z^2 - 1296*a^4*b*c^5*d*e*m^2*z^2 + 81*a*b^7*c^2*d*e*m^2*z^2 - 324*a^3*b
*c^6*d*g^2*j*z^2 - 81*a*b^4*c^5*d^2*g*j*z^2 + 81*a*b^4*c^5*d^2*e*l*z^2 + 32
4*a^3*b*c^6*e*g^2*h*z^2 + 81*a*b^4*c^5*d*e^2*k*z^2 + 1296*a^3*b*c^6*d*e*j^2
*z^2 - 324*a^3*b*c^6*e*f*h^2*z^2 + 324*a^3*b*c^6*d*g*h^2*z^2 + 81*a*b^5*c^4
*d*e*j^2*z^2 - 324*a^2*b*c^7*d^2*f*g*z^2 + 324*a^2*b*c^7*d^2*e*h*z^2 + 81*a
*b^3*c^6*d^2*f*g*z^2 - 81*a*b^3*c^6*d^2*e*h*z^2 + 324*a^2*b*c^7*d*e^2*g*z^2
 - 81*a*b^3*c^6*d*e^2*g*z^2 + 1296*a^6*c^4*j*k*l*m*z^2 - 1296*a^5*c^5*f*j*k
*l*z^2 - 1296*a^5*c^5*e*j*k*m*z^2 - 1296*a^5*c^5*d*j*l*m*z^2 - 1296*a^5*c^5
*g*h*j*m*z^2 + 1296*a^5*c^5*e*h*l*m*z^2 + 1296*a^4*c^6*e*f*j*k*z^2 + 1296*a
^4*c^6*d*g*j*k*z^2 + 1296*a^4*c^6*d*f*j*l*z^2 - 1296*a^4*c^6*d*e*k*l*z^2 +
1296*a^4*c^6*d*e*j*m*z^2 + 1296*a^4*c^6*f*g*h*j*z^2 - 1296*a^4*c^6*e*f*h*l*
z^2 - 1296*a^3*c^7*d*e*f*j*z^2 + 648*a^5*b^3*c^2*k*l*m^2*z^2 + 648*a^4*b^3*
c^3*j^2*k*l*z^2 + 486*a^5*b^2*c^3*h*l^2*m*z^2 - 81*a^4*b^4*c^2*h*l^2*m*z^2
+ 81*a^4*b^3*c^3*h^2*l*m*z^2 - 81*a^3*b^5*c^2*j^2*k*l*z^2 - 162*a^4*b^2*c^4
*g^2*k*m*z^2 - 81*a^4*b^3*c^3*h*k^2*l*z^2 + 81*a^4*b^3*c^3*g*k^2*m*z^2 - 56
7*a^4*b^3*c^3*h*j*l^2*z^2 + 486*a^4*b^2*c^4*h^2*j*l*z^2 - 81*a^4*b^3*c^3*g*
k*l^2*z^2 + 81*a^4*b^3*c^3*e*l^2*m*z^2 + 81*a^3*b^5*c^2*h*j*l^2*z^2 - 81*a^
3*b^4*c^3*h^2*j*l*z^2 + 81*a^3*b^3*c^4*e^2*l*m*z^2 + 2430*a^4*b^3*c^3*f*j*m
^2*z^2 - 2268*a^4*b^2*c^4*f*j^2*m*z^2 - 810*a^3*b^5*c^2*f*j*m^2*z^2 + 810*a
^3*b^4*c^3*f*j^2*m*z^2 - 648*a^4*b^3*c^3*e*k*m^2*z^2 - 648*a^4*b^3*c^3*d*l*
m^2*z^2 - 648*a^4*b^2*c^4*h*j^2*k*z^2 - 648*a^4*b^2*c^4*g*j^2*l*z^2 - 162*a
^3*b^3*c^4*f^2*j*m*z^2 + 81*a^3*b^5*c^2*e*k*m^2*z^2 + 81*a^3*b^5*c^2*d*l*m^
2*z^2 + 81*a^3*b^4*c^3*h*j^2*k*z^2 + 81*a^3*b^4*c^3*g*j^2*l*z^2 - 81*a^2*b^
6*c^2*f*j^2*m*z^2 - 648*a^4*b^3*c^3*g*h*m^2*z^2 + 486*a^4*b^2*c^4*g*j*k^2*z
^2 - 486*a^4*b^2*c^4*e*k^2*l*z^2 + 486*a^3*b^2*c^5*d^2*k*m*z^2 - 162*a^4*b^
2*c^4*d*k^2*m*z^2 + 81*a^3*b^5*c^2*g*h*m^2*z^2 - 81*a^3*b^4*c^3*g*j*k^2*z^2
 + 81*a^3*b^4*c^3*e*k^2*l*z^2 + 81*a^3*b^3*c^4*g^2*j*k*z^2 - 81*a^2*b^4*c^4
*d^2*k*m*z^2 + 486*a^4*b^2*c^4*e*j*l^2*z^2 - 486*a^4*b^2*c^4*d*k*l^2*z^2 -
162*a^3*b^2*c^5*e^2*j*l*z^2 - 81*a^3*b^4*c^3*e*j*l^2*z^2 + 81*a^3*b^4*c^3*d
*k*l^2*z^2 - 81*a^3*b^3*c^4*g^2*h*l*z^2 - 1458*a^4*b^2*c^4*f*h*l^2*z^2 + 64
8*a^3*b^4*c^3*f*h*l^2*z^2 - 567*a^3*b^3*c^4*f*h^2*l*z^2 + 486*a^3*b^2*c^5*e
^2*h*m*z^2 - 81*a^3*b^3*c^4*g*h^2*k*z^2 + 81*a^3*b^3*c^4*e*h^2*m*z^2 - 81*a
^2*b^6*c^2*f*h*l^2*z^2 + 81*a^2*b^5*c^3*f*h^2*l*z^2 - 81*a^2*b^4*c^4*e^2*h*
m*z^2 - 1296*a^4*b^2*c^4*e*g*m^2*z^2 - 1296*a^4*b^2*c^4*d*h*m^2*z^2 + 648*a
^3*b^4*c^3*e*g*m^2*z^2 + 648*a^3*b^4*c^3*d*h*m^2*z^2 + 81*a^3*b^3*c^4*d*j*k
^2*z^2 - 81*a^2*b^6*c^2*e*g*m^2*z^2 - 81*a^2*b^6*c^2*d*h*m^2*z^2 + 81*a^2*b
^3*c^5*d^2*j*k*z^2 - 567*a^3*b^3*c^4*f*g*k^2*z^2 - 567*a^2*b^3*c^5*d^2*g*m*
z^2 + 486*a^3*b^2*c^5*f*g^2*k*z^2 - 486*a^3*b^2*c^5*e*g^2*l*z^2 + 486*a^3*b
^2*c^5*d*g^2*m*z^2 - 81*a^3*b^3*c^4*e*h*k^2*z^2 + 81*a^2*b^5*c^3*f*g*k^2*z^
2 - 81*a^2*b^4*c^4*f*g^2*k*z^2 + 81*a^2*b^4*c^4*e*g^2*l*z^2 - 81*a^2*b^4*c^
4*d*g^2*m*z^2 - 81*a^2*b^3*c^5*d^2*h*l*z^2 - 567*a^3*b^3*c^4*e*f*l^2*z^2 -
486*a^3*b^2*c^5*d*h^2*k*z^2 - 162*a^3*b^2*c^5*e*h^2*j*z^2 - 81*a^3*b^3*c^4*
d*g*l^2*z^2 + 81*a^2*b^5*c^3*e*f*l^2*z^2 + 81*a^2*b^4*c^4*d*h^2*k*z^2 + 81*
a^2*b^3*c^5*e^2*h*j*z^2 - 81*a^2*b^3*c^5*e^2*g*k*z^2 + 81*a^2*b^3*c^5*e^2*f
*l*z^2 + 1944*a^3*b^3*c^4*d*e*m^2*z^2 - 729*a^2*b^5*c^3*d*e*m^2*z^2 + 648*a

^3*b^2*c^5*e*g*j^2*z^2 + 648*a^3*b^2*c^5*d*h*j^2*z^2 - 81*a^2*b^4*c^4*e*g*j
^2*z^2 - 81*a^2*b^4*c^4*d*h*j^2*z^2 + 486*a^3*b^2*c^5*d*f*k^2*z^2 + 486*a^2
*b^2*c^6*d^2*g*j*z^2 - 486*a^2*b^2*c^6*d^2*e*l*z^2 - 162*a^2*b^2*c^6*d^2*f*
k*z^2 - 81*a^2*b^4*c^4*d*f*k^2*z^2 + 81*a^2*b^3*c^5*d*g^2*j*z^2 - 486*a^2*b
^2*c^6*d*e^2*k*z^2 - 81*a^2*b^3*c^5*e*g^2*h*z^2 - 648*a^2*b^3*c^5*d*e*j^2*z
^2 - 162*a^2*b^2*c^6*e^2*f*h*z^2 + 81*a^2*b^3*c^5*e*f*h^2*z^2 - 81*a^2*b^3*
c^5*d*g*h^2*z^2 - 162*a^2*b^2*c^6*d*f*g^2*z^2 - 189*a^5*b^3*c^2*l^3*m*z^2 +
162*a^5*b^2*c^3*k^3*m*z^2 - 27*a^4*b^4*c^2*k^3*m*z^2 - 702*a^4*b^3*c^3*j^3
*m*z^2 - 81*a^3*b^6*c*j^2*m^2*z^2 + 81*a^3*b^5*c^2*j^3*m*z^2 - 54*a^5*b^3*c
^2*j*m^3*z^2 - 486*a^5*b^2*c^3*j*l^3*z^2 + 216*a^4*b^4*c^2*j*l^3*z^2 - 189*
a^4*b^3*c^3*j*k^3*z^2 - 54*a^4*b^2*c^4*h^3*m*z^2 + 27*a^3*b^5*c^2*j*k^3*z^2
+ 27*a^3*b^3*c^4*g^3*m*z^2 - 810*a^4*b^4*c^2*f*m^3*z^2 + 540*a^5*b^2*c^3*f
*m^3*z^2 - 324*a^3*b^2*c^5*f^3*m*z^2 + 54*a^2*b^4*c^4*f^3*m*z^2 + 675*a^4*b
^3*c^3*f*l^3*z^2 - 243*a^3*b^5*c^2*f*l^3*z^2 - 189*a^2*b^3*c^5*e^3*m*z^2 +
27*a^3*b^3*c^4*h^3*j*z^2 - 486*a^4*b^2*c^4*f*k^3*z^2 - 486*a^2*b^2*c^6*d^3*
m*z^2 + 216*a^3*b^4*c^3*f*k^3*z^2 - 54*a^3*b^2*c^5*g^3*j*z^2 - 27*a^2*b^6*c
^2*f*k^3*z^2 - 270*a^3*b^3*c^4*f*j^3*z^2 - 54*a^2*b^3*c^5*f^3*j*z^2 + 27*a^
2*b^5*c^3*f*j^3*z^2 + 162*a^2*b^2*c^6*e^3*j*z^2 + 162*a^3*b^2*c^5*f*h^3*z^2
- 27*a^2*b^4*c^4*f*h^3*z^2 + 27*a^2*b^3*c^5*f*g^3*z^2 + 81*a*b^2*c^7*d^2*e
^2*z^2 - 648*a^6*c^4*h*l^2*m*z^2 + 648*a^5*c^5*g^2*k*m*z^2 - 648*a^5*c^5*h^
2*j*l*z^2 + 1296*a^5*c^5*h*j^2*k*z^2 + 1296*a^5*c^5*g*j^2*l*z^2 + 1296*a^5*
c^5*f*j^2*m*z^2 - 648*a^5*c^5*g*j*k^2*z^2 + 648*a^5*c^5*e*k^2*l*z^2 + 648*a
^5*c^5*d*k^2*m*z^2 - 648*a^4*c^6*d^2*k*m*z^2 - 648*a^5*c^5*e*j*l^2*z^2 + 64
8*a^5*c^5*d*k*l^2*z^2 + 648*a^4*c^6*e^2*j*l*z^2 + 324*a^6*b*c^3*l^3*m*z^2 +
27*a^4*b^5*c*l^3*m*z^2 + 648*a^5*c^5*f*h*l^2*z^2 - 648*a^4*c^6*e^2*h*m*z^2
+ 1512*a^5*b*c^4*j^3*m*z^2 + 1080*a^6*b*c^3*j*m^3*z^2 - 162*a^4*b^5*c*j*m^
3*z^2 - 648*a^4*c^6*f*g^2*k*z^2 + 648*a^4*c^6*e*g^2*l*z^2 - 648*a^4*c^6*d*g
^2*m*z^2 - 27*a^3*b^6*c*j*l^3*z^2 + 648*a^4*c^6*e*h^2*j*z^2 + 648*a^4*c^6*d
*h^2*k*z^2 + 324*a^5*b*c^4*j*k^3*z^2 - 1296*a^4*c^6*e*g*j^2*z^2 - 1296*a^4*
c^6*d*h*j^2*z^2 - 108*a^4*b*c^5*g^3*m*z^2 - 648*a^4*c^6*d*f*k^2*z^2 - 648*a
^3*c^7*d^2*g*j*z^2 + 648*a^3*c^7*d^2*f*k*z^2 + 648*a^3*c^7*d^2*e*l*z^2 + 27
0*a^3*b^6*c*f*m^3*z^2 + 648*a^3*c^7*d*e^2*k*z^2 - 540*a^5*b*c^4*f*l^3*z^2 +
324*a^3*b*c^6*e^3*m*z^2 - 108*a^4*b*c^5*h^3*j*z^2 + 27*a^2*b^7*c*f*l^3*z^2
+ 27*a*b^5*c^4*e^3*m*z^2 + 648*a^3*c^7*e^2*f*h*z^2 + 216*a*b^4*c^5*d^3*m*z
^2 + 648*a^4*b*c^5*f*j^3*z^2 + 216*a^3*b*c^6*f^3*j*z^2 + 648*a^3*c^7*d*f*g^
2*z^2 - 27*a*b^4*c^5*e^3*j*z^2 + 324*a^2*b*c^7*d^3*j*z^2 - 189*a*b^3*c^6*d^
3*j*z^2 - 108*a^3*b*c^6*f*g^3*z^2 - 108*a^2*b*c^7*e^3*f*z^2 + 27*a*b^3*c^6*
e^3*f*z^2 + 162*a*b^2*c^7*d^3*f*z^2 - 1134*a^5*b^2*c^3*j^2*m^2*z^2 + 648*a^
4*b^4*c^2*j^2*m^2*z^2 + 81*a^5*b^2*c^3*k^2*l^2*z^2 + 162*a^4*b^2*c^4*f^2*m^
2*z^2 + 81*a^4*b^2*c^4*h^2*k^2*z^2 + 81*a^4*b^2*c^4*g^2*l^2*z^2 + 162*a^3*b
^2*c^5*f^2*j^2*z^2 + 81*a^3*b^2*c^5*e^2*k^2*z^2 + 81*a^3*b^2*c^5*d^2*l^2*z^
2 + 81*a^3*b^2*c^5*g^2*h^2*z^2 + 81*a^2*b^2*c^6*e^2*g^2*z^2 + 81*a^2*b^2*c^
6*d^2*h^2*z^2 - 216*a^6*c^4*k^3*m*z^2 + 216*a^6*c^4*j*l^3*z^2 + 27*a^3*b^7*
j*m^3*z^2 + 216*a^5*c^5*h^3*m*z^2 + 432*a^6*c^4*f*m^3*z^2 + 432*a^4*c^6*f^3
*m*z^2 - 27*b^6*c^4*d^3*m*z^2 - 27*a^2*b^8*f*m^3*z^2 + 216*a^5*c^5*f*k^3*z^

```
2 + 216*a^4*c^6*g^3*j*z^2 + 216*a^3*c^7*d^3*m*z^2 + 216*a^5*b^4*c*m^4*z^2 -
 216*a^3*c^7*e^3*j*z^2 + 27*b^5*c^5*d^3*j*z^2 - 216*a^4*c^6*f*h^3*z^2 - 27*
b^4*c^6*d^3*f*z^2 - 216*a^2*c^8*d^3*f*z^2 - 648*a^6*c^4*j^2*m^2*z^2 - 324*a
^6*c^4*k^2*l^2*z^2 - 648*a^5*c^5*f^2*m^2*z^2 - 324*a^5*c^5*h^2*k^2*z^2 - 32
4*a^5*c^5*g^2*l^2*z^2 - 648*a^4*c^6*f^2*j^2*z^2 - 324*a^4*c^6*e^2*k^2*z^2 -
 324*a^4*c^6*d^2*l^2*z^2 - 405*a^6*b^2*c^2*m^4*z^2 - 324*a^4*c^6*g^2*h^2*z^
2 - 324*a^3*c^7*e^2*g^2*z^2 - 324*a^3*c^7*d^2*h^2*z^2 + 243*a^4*b^2*c^4*j^4
*z^2 - 27*a^3*b^4*c^3*j^4*z^2 - 324*a^2*c^8*d^2*e^2*z^2 + 27*a^2*b^2*c^6*f^
4*z^2 - 108*a^7*c^3*m^4*z^2 - 27*a^4*b^6*m^4*z^2 - 540*a^5*c^5*j^4*z^2 - 10
8*a^3*c^7*f^4*z^2 - 216*a^5*b*c^3*f*j*k*l*m*z - 54*a^3*b^5*c*f*j*k*l*m*z +
27*a^3*b^5*c*g*h*k*l*m*z - 27*a^2*b^6*c*e*g*k*l*m*z - 27*a^2*b^6*c*d*h*k*l*
m*z + 432*a^4*b*c^4*d*g*j*k*m*z - 432*a^4*b*c^4*d*e*k*l*m*z + 216*a^4*b*c^4
*e*g*j*k*l*z + 216*a^4*b*c^4*e*f*j*k*m*z + 216*a^4*b*c^4*d*h*j*k*l*z + 216*
a^4*b*c^4*d*f*j*l*m*z + 216*a^4*b*c^4*f*g*h*j*m*z - 27*a*b^6*c^2*d*e*j*k*l*
z - 27*a*b^6*c^2*d*e*h*k*m*z - 27*a*b^6*c^2*d*e*g*l*m*z + 216*a^3*b*c^5*d*e
*h*j*k*z + 216*a^3*b*c^5*d*e*g*j*l*z - 216*a^3*b*c^5*d*e*f*j*m*z + 27*a*b^5
*c^3*d*e*h*j*k*z + 27*a*b^5*c^3*d*e*g*j*l*z + 27*a*b^5*c^3*d*e*g*h*m*z - 27
*a*b^4*c^4*d*e*g*h*j*z + 27*a*b^7*c*d*e*k*l*m*z + 270*a^4*b^3*c^2*f*j*k*l*m
*z - 108*a^4*b^3*c^2*g*h*k*l*m*z - 216*a^4*b^2*c^3*f*h*j*k*m*z - 216*a^4*b^
2*c^3*f*g*j*l*m*z - 216*a^4*b^2*c^3*e*g*k*l*m*z - 216*a^4*b^2*c^3*d*h*k*l*m
*z + 162*a^3*b^4*c^2*e*g*k*l*m*z + 162*a^3*b^4*c^2*d*h*k*l*m*z + 108*a^4*b^
2*c^3*g*h*j*k*l*z + 108*a^4*b^2*c^3*e*h*j*l*m*z + 54*a^3*b^4*c^2*f*h*j*k*m*
z + 54*a^3*b^4*c^2*f*g*j*l*m*z - 27*a^3*b^4*c^2*g*h*j*k*l*z + 540*a^3*b^3*c
^3*d*e*k*l*m*z - 216*a^2*b^5*c^2*d*e*k*l*m*z - 162*a^3*b^3*c^3*e*g*j*k*l*z
 - 162*a^3*b^3*c^3*d*h*j*k*l*z - 108*a^3*b^3*c^3*d*g*j*k*m*z - 54*a^3*b^3*c^
3*e*f*j*k*m*z - 54*a^3*b^3*c^3*d*f*j*l*m*z + 27*a^2*b^5*c^2*e*g*j*k*l*z + 2
7*a^2*b^5*c^2*d*h*j*k*l*z - 108*a^3*b^3*c^3*e*g*h*k*m*z - 108*a^3*b^3*c^3*d
*g*h*l*m*z - 54*a^3*b^3*c^3*f*g*h*j*m*z + 27*a^2*b^5*c^2*e*g*h*k*m*z + 27*a
^2*b^5*c^2*d*g*h*l*m*z - 540*a^3*b^2*c^4*d*e*j*k*l*z + 216*a^2*b^4*c^3*d*e*
j*k*l*z - 216*a^3*b^2*c^4*d*e*h*k*m*z - 216*a^3*b^2*c^4*d*e*g*l*m*z + 162*a
^2*b^4*c^3*d*e*h*k*m*z + 162*a^2*b^4*c^3*d*e*g*l*m*z + 108*a^3*b^2*c^4*e*g*
h*j*k*z - 108*a^3*b^2*c^4*e*f*h*j*l*z + 108*a^3*b^2*c^4*d*g*h*j*l*z + 108*a
^3*b^2*c^4*d*f*g*k*m*z - 27*a^2*b^4*c^3*e*g*h*j*k*z - 27*a^2*b^4*c^3*d*g*h*
j*l*z - 162*a^2*b^3*c^4*d*e*h*j*k*z - 162*a^2*b^3*c^4*d*e*g*j*l*z + 54*a^2*
b^3*c^4*d*e*f*j*m*z - 108*a^2*b^3*c^4*d*e*g*h*m*z + 108*a^2*b^2*c^5*d*e*g*h
*j*z + 324*a^6*b*c^2*j*k*l*m^2*z - 81*a^5*b^3*c*j*k*l*m^2*z + 27*a^4*b^4*c*
j^2*k*l*m*z - 27*a^4*b^4*c*h*k^2*l*m*z - 27*a^4*b^4*c*g*k*l^2*m*z + 216*a^5
*b*c^3*h*j^2*k*m*z + 216*a^5*b*c^3*g*j^2*l*m*z + 54*a^4*b^4*c*f*k*l*m^2*z +
 27*a^4*b^4*c*h*j*k*m^2*z + 27*a^4*b^4*c*g*j*l*m^2*z + 27*a^2*b^6*c*f^2*k*l
*m*z + 216*a^5*b*c^3*e*k^2*l*m*z - 108*a^5*b*c^3*h*j*k^2*l*z + 27*a^3*b^5*c
*e*k^2*l*m*z + 216*a^5*b*c^3*d*k*l^2*m*z + 216*a^4*b*c^4*e^2*j*l*m*z - 108*
a^5*b*c^3*g*j*k*l^2*z + 27*a^3*b^5*c*d*k*l^2*m*z - 324*a^5*b*c^3*e*j*k*m^2*
z - 324*a^5*b*c^3*d*j*l*m^2*z - 216*a^5*b*c^3*f*h*l^2*m*z - 108*a^4*b*c^4*f
^2*j*k*l*z - 27*a^3*b^5*c*e*j*k*m^2*z - 27*a^3*b^5*c*d*j*l*m^2*z - 324*a^5*
b*c^3*g*h*j*m^2*z + 216*a^5*b*c^3*f*h*k*m^2*z + 216*a^5*b*c^3*f*g*l*m^2*z +
```

216*a^5*b*c^3*e*h*l*m^2*z - 216*a^4*b*c^4*f^2*h*k*m*z - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5*c*g*h*j*m^2*z + 216*a^4*b*c^4*e*g^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 216*a^4*b*c^4*f*h^2*j*l*z + 216*a^4*b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k*m*z - 108*a^4*b*c^4*g*h^2*j*k*z - 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^4*d*h*j^2*m*z + 216*a^4*b*c^4*f*h*j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*a^2*b^6*c*e*g*j*m^2*z + 27*a^2*b^6*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*z - 216*a^4*b*c^4*f*g*j*k^2*z + 216*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d^2*e*l*m*z - 108*a^4*b*c^4*e*h*j*k^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^3*b*c^5*d^2*h*j*l*z + 108*a^3*b*c^5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z + 27*a*b^5*c^3*d^2*g*k*l*z + 27*a*b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j*l^2*z + 216*a^3*b*c^5*d*e^2*k*m*z - 108*a^4*b*c^4*d*g*j*l^2*z - 108*a^3*b*c^5*e^2*g*j*k*z + 27*a*b^5*c^3*d*e^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^2*z + 216*a^3*b*c^5*e^2*f*h*m*z - 108*a^4*b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5*e^2*g*h*l*z + 108*a^3*b*c^5*e*f^2*j*k*z + 108*a^3*b*c^5*d*f^2*j*l*z + 27*a*b^6*c^2*d*e*j^2*m*z - 216*a^3*b*c^5*e*f^2*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*z - 27*a*b^4*c^4*d^2*e*j*l*z + 216*a^3*b*c^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*g^2*h*j*z + 54*a*b^4*c^4*d^2*f*g*m*z - 27*a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*c^4*d^2*e*h*m*z - 27*a*b^4*c^4*d*e^2*j*k*z - 108*a^3*b*c^5*d*g*h^2*j*z + 54*a*b^4*c^4*d*e^2*h*l*z + 27*a*b^6*c^2*d*e*h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*z - 27*a*b^4*c^4*d*e^2*g*m*z - 27*a*b^4*c^4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2*f*g*j*z - 108*a^3*b*c^5*d*e*g*k^2*z - 108*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*b*c^6*d^2*e*g*k*z - 54*a*b^3*c^5*d^2*f*g*j*z - 27*a*b^5*c^3*d*e*g*k^2*z + 27*a*b^4*c^4*d*e*g^2*k*z + 27*a*b^3*c^5*d^2*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k*z - 108*a^2*b*c^6*d*e^2*g*j*z + 27*a*b^3*c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d*e*f^2*j*z + 27*a*b^3*c^5*d*e*f^2*j*z - 432*a^5*c^4*e*h*j*l*m*z + 432*a^4*c^5*d*e*j*k*l*z + 432*a^4*c^5*e*f*h*j*l*z - 432*a^4*c^5*d*f*g*k*m*z - 27*a*b^7*c*d*e*j*m^2*z - 54*a^5*b^2*c^2*j^2*k*l*m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z + 108*a^5*b^2*c^2*g*k*l^2*m*z - 54*a^5*b^2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c^3*f^2*k*l*m*z - 270*a^5*b^2*c^2*f*k*l*m^2*z - 189*a^3*b^4*c^2*f^2*k*l*m*z - 108*a^5*b^2*c^2*h*j*k*m^2*z - 108*a^5*b^2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^2*h*j^2*k*m*z - 54*a^4*b^3*c^2*g*j^2*l*m*z - 162*a^4*b^3*c^2*e*k^2*l*m*z + 54*a^4*b^2*c^3*g^2*j*k*m*z + 27*a^4*b^3*c^2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d*k*l^2*m*z + 108*a^4*b^2*c^3*g^2*h*l*m*z - 54*a^3*b^3*c^3*e^2*j*l*m*z + 27*a^4*b^3*c^2*g*j*k*l^2*z - 27*a^3*b^4*c^2*g^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^2*k*l*z + 189*a^4*b^3*c^2*e*j*k*m^2*z + 189*a^4*b^3*c^2*d*j*l*m^2*z - 162*a^4*b^2*c^3*e*j^2*k*m*z - 162*a^4*b^2*c^3*d*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*j*k*l*z + 108*a^4*b^2*c^3*g*h^2*k*m*z + 54*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4*b^2*c^3*f*h^2*l*m*z + 54*a^3*b^4*c^2*f*j^2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*m*z + 27*a^3*b^4*c^2*e*j^2*k*m*z + 27*a^3*b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*c^2*f^2*j*k*l*z - 270*a^3*b^2*c^4*d^2*j*k*m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z - 162*a^4*b^2*c^3*g*h*j^2*m*z + 162*a^4*b^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*c^3*f^2*h*k*m*z + 162*a^3*b^3*c^3*f^2*g*l*m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z - 54*a^4*b^3*c^2*f*g*l*m^2*z - 54*a^4*b^3*c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*d*j*k^2*m*z + 54*a^2*b^4*c^3*d^2*j*k*m*z + 27*a^3*b^4*c^2*g*h*j^2*m*z - 27*a^3*b^4*c^2*e*j*k^2*l*z - 27*a^2*b^5*c^2*f^2*h*k*m*z - 27*a^2*b

^5*c^2*f^2*g*l*m*z + 162*a^4*b^2*c^3*d*j*k*l^2*z - 162*a^3*b^3*c^3*e*g^2*l*
m*z + 108*a^4*b^2*c^3*e*h*k^2*m*z + 108*a^3*b^2*c^4*d^2*h*l*m*z - 54*a^4*b^
2*c^3*f*g*k^2*m*z - 27*a^3*b^4*c^2*e*h*k^2*m*z - 27*a^3*b^4*c^2*d*j*k*l^2*z
 + 27*a^3*b^3*c^3*g^2*h*j*l*z + 27*a^2*b^5*c^2*e*g^2*l*m*z - 27*a^2*b^4*c^3
*d^2*h*l*m*z + 270*a^4*b^2*c^3*f*h*j*l^2*z - 270*a^3*b^2*c^4*e^2*h*j*m*z -
162*a^4*b^2*c^3*e*h*k*l^2*z - 162*a^3*b^3*c^3*d*h^2*k*m*z + 162*a^3*b^2*c^4
*e^2*h*k*l*z + 108*a^4*b^2*c^3*d*g*l^2*m*z + 108*a^3*b^2*c^4*e^2*g*k*m*z -
54*a^4*b^2*c^3*e*f*l^2*m*z - 54*a^3*b^4*c^2*f*h*j*l^2*z + 54*a^3*b^3*c^3*f*
h^2*j*l*z - 54*a^3*b^3*c^3*e*h^2*j*m*z + 54*a^3*b^2*c^4*e^2*f*l*m*z + 54*a^
2*b^4*c^3*e^2*h*j*m*z + 27*a^3*b^4*c^2*e*h*k*l^2*z - 27*a^3*b^4*c^2*d*g*l^2
*m*z + 27*a^3*b^3*c^3*g*h^2*j*k*z + 27*a^2*b^5*c^2*d*h^2*k*m*z - 27*a^2*b^4
*c^3*e^2*h*k*l*z - 27*a^2*b^4*c^3*e^2*g*k*m*z + 432*a^4*b^2*c^3*e*g*j*m^2*z
 + 432*a^4*b^2*c^3*d*h*j*m^2*z - 270*a^4*b^2*c^3*d*g*k*m^2*z - 216*a^3*b^4*
c^2*e*g*j*m^2*z - 216*a^3*b^4*c^2*d*h*j*m^2*z + 216*a^3*b^3*c^3*e*g*j^2*m*z
 + 216*a^3*b^3*c^3*d*h*j^2*m*z - 162*a^3*b^2*c^4*e*f^2*k*m*z - 162*a^3*b^2*
c^4*d*f^2*l*m*z - 108*a^3*b^2*c^4*f^2*h*j*k*z - 108*a^3*b^2*c^4*f^2*g*j*l*z
 + 54*a^4*b^2*c^3*e*f*k*m^2*z + 54*a^4*b^2*c^3*d*f*l*m^2*z + 54*a^3*b^4*c^2
*d*g*k*m^2*z - 54*a^3*b^3*c^3*f*h*j^2*k*z - 54*a^3*b^3*c^3*f*g*j^2*l*z - 27
*a^2*b^5*c^2*e*g*j^2*m*z - 27*a^2*b^5*c^2*d*h*j^2*m*z + 27*a^2*b^4*c^3*f^2*
h*j*k*z + 27*a^2*b^4*c^3*f^2*g*j*l*z + 27*a^2*b^4*c^3*e*f^2*k*m*z + 27*a^2*
b^4*c^3*d*f^2*l*m*z + 324*a^2*b^3*c^4*d^2*g*j*m*z - 270*a^3*b^2*c^4*d*g^2*j
*m*z - 162*a^3*b^2*c^4*f^2*g*h*m*z + 162*a^3*b^2*c^4*e*g^2*j*l*z - 162*a^2*
b^3*c^4*d^2*e*l*m*z - 135*a^2*b^3*c^4*d^2*g*k*l*z + 108*a^3*b^2*c^4*d*g^2*k
*l*z + 54*a^4*b^2*c^3*f*g*h*m^2*z + 54*a^3*b^3*c^3*f*g*j*k^2*z - 54*a^3*b^2
*c^4*f*g^2*j*k*z + 54*a^2*b^4*c^3*d*g^2*j*m*z - 54*a^2*b^3*c^4*d^2*f*k*m*z
+ 27*a^3*b^3*c^3*e*h*j*k^2*z + 27*a^3*b^3*c^3*d*g*k^2*l*z + 27*a^2*b^4*c^3*
f^2*g*h*m*z - 27*a^2*b^4*c^3*e*g^2*j*l*z - 27*a^2*b^4*c^3*d*g^2*k*l*z + 27*
a^2*b^3*c^4*d^2*h*j*l*z + 162*a^3*b^2*c^4*d*h^2*j*k*z - 162*a^2*b^3*c^4*d*e
^2*k*m*z + 108*a^3*b^2*c^4*e*g^2*h*m*z + 54*a^3*b^3*c^3*e*f*j*l^2*z + 27*a^
3*b^3*c^3*d*g*j*l^2*z - 27*a^2*b^4*c^3*e*g^2*h*m*z - 27*a^2*b^4*c^3*d*h^2*j
*k*z + 27*a^2*b^3*c^4*e^2*g*j*k*z - 621*a^3*b^3*c^3*d*e*j*m^2*z + 594*a^3*b
^2*c^4*d*e*j^2*m*z + 243*a^2*b^5*c^2*d*e*j*m^2*z - 243*a^2*b^4*c^3*d*e*j^2*
m*z + 135*a^3*b^3*c^3*e*g*h*l^2*z - 108*a^3*b^2*c^4*e*g*h^2*l*z + 108*a^3*b
^2*c^4*d*g*h^2*m*z + 54*a^3*b^2*c^4*e*f*j^2*k*z + 54*a^3*b^2*c^4*e*f*h^2*m*
z + 54*a^3*b^2*c^4*d*g*j^2*k*z + 54*a^3*b^2*c^4*d*f*j^2*l*z - 54*a^2*b^3*c^
4*e^2*f*h*m*z - 27*a^2*b^5*c^2*e*g*h*l^2*z + 27*a^2*b^4*c^3*e*g*h^2*l*z - 2
7*a^2*b^4*c^3*d*g*h^2*m*z - 27*a^2*b^3*c^4*e^2*g*h*l*z - 27*a^2*b^3*c^4*e*f
^2*j*k*z - 27*a^2*b^3*c^4*d*f^2*j*l*z + 162*a^2*b^2*c^5*d^2*e*j*l*z + 54*a^
3*b^2*c^4*f*g*h*j^2*z - 54*a^3*b^2*c^4*d*f*j*k^2*z + 54*a^2*b^3*c^4*e*f^2*h
*l*z + 54*a^2*b^2*c^5*d^2*f*j*k*z - 27*a^2*b^3*c^4*f^2*g*h*j*z - 270*a^2*b^
2*c^5*d^2*f*g*m*z - 162*a^3*b^2*c^4*d*g*h*k^2*z + 162*a^2*b^2*c^5*d^2*g*h*k
*z + 162*a^2*b^2*c^5*d*e^2*j*k*z + 108*a^2*b^2*c^5*d^2*e*h*m*z - 54*a^2*b^3
*c^4*d*f*g^2*m*z + 27*a^2*b^4*c^3*d*g*h*k^2*z + 27*a^2*b^3*c^4*e*g^2*h*j*z
+ 270*a^3*b^2*c^4*d*e*h*l^2*z - 270*a^2*b^2*c^5*d*e^2*h*l*z - 162*a^2*b^4*c
^3*d*e*h*l^2*z + 108*a^2*b^3*c^4*d*e*h^2*l*z + 108*a^2*b^2*c^5*d*e^2*g*m*z

+ 54*a^2*b^2*c^5*e^2*f*h*j*z + 27*a^2*b^3*c^4*d*g*h^2*j*z + 162*a^2*b^2*c^5
*d*e*f^2*m*z - 54*a^3*b^2*c^4*d*e*f*m^2*z - 54*a^2*b^2*c^5*d*f^2*g*k*z + 13
5*a^2*b^3*c^4*d*e*g*k^2*z - 108*a^2*b^2*c^5*d*e*g^2*k*z + 54*a^2*b^2*c^5*d*
f*g^2*j*z - 54*a^2*b^2*c^5*d*e*f*j^2*z - 9*a*b^7*c*d*e*l^3*z - 36*a*b*c^7*d
^3*e*g*z - 108*a^6*b*c^2*k^2*l^2*m*z + 27*a^5*b^3*c*k^2*l^2*m*z - 18*a^5*b^
2*c^2*j*k^3*m*z - 27*a^4*b^3*c^2*j^3*k*l*z - 108*a^5*b*c^3*h^2*k^2*m*z - 10
8*a^5*b*c^3*g^2*l^2*m*z + 108*a^5*b*c^3*h^2*k*l^2*z + 108*a^5*b*c^3*g^2*k*m
^2*z + 90*a^5*b^2*c^2*f*l^3*m*z - 18*a^5*b^2*c^2*h*k*l^3*z + 18*a^4*b^2*c^3
*h^3*k*l*z + 18*a^4*b^2*c^3*h^3*j*m*z - 108*a^5*b*c^3*h*j^2*l^2*z + 18*a^4*
b^3*c^2*f*k^3*m*z - 18*a^3*b^3*c^3*g^3*j*m*z - 9*a^4*b^3*c^2*g*k^3*l*z + 9*
a^3*b^3*c^3*g^3*k*l*z + 252*a^4*b^2*c^3*f*j^3*m*z + 216*a^5*b*c^3*f*j^2*m^2
*z + 180*a^3*b^2*c^4*f^3*j*m*z - 108*a^4*b*c^4*e^2*k^2*m*z - 108*a^4*b*c^4*
d^2*l^2*m*z + 90*a^5*b^2*c^2*e*k*m^3*z + 90*a^5*b^2*c^2*d*l*m^3*z - 90*a^3*
b^2*c^4*f^3*k*l*z + 54*a^3*b^5*c*f*j^2*m^2*z - 54*a^3*b^4*c^2*f*j^3*m*z + 3
6*a^5*b^2*c^2*f*j*m^3*z + 36*a^4*b^2*c^3*h*j^3*k*z + 36*a^4*b^2*c^3*g*j^3*l
*z - 36*a^2*b^4*c^3*f^3*j*m*z - 27*a^2*b^6*c*f^2*j*m^2*z + 18*a^2*b^4*c^3*f
^3*k*l*z - 216*a^4*b*c^4*d^2*k*m^2*z + 108*a^5*b*c^3*d*k^2*m^2*z - 108*a^4*
b^3*c^2*f*j*l^3*z - 108*a^4*b*c^4*g^2*h^2*m*z + 108*a^2*b^3*c^4*e^3*j*m*z +
 90*a^5*b^2*c^2*g*h*m^3*z + 54*a^4*b^3*c^2*e*k*l^3*z - 54*a^2*b^3*c^4*e^3*k
*l*z + 234*a^2*b^2*c^5*d^3*j*m*z - 144*a^2*b^2*c^5*d^3*k*l*z + 90*a^4*b^2*c
^3*f*j*k^3*z - 72*a^4*b^2*c^3*d*k^3*l*z + 27*a^4*b^3*c^2*g*h*l^3*z - 27*a^3
*b^3*c^3*g*h^3*l*z - 18*a^3*b^4*c^2*f*j*k^3*z + 9*a^3*b^4*c^2*d*k^3*l*z + 2
16*a^4*b*c^4*f^2*h*l^2*z - 216*a^4*b*c^4*e^2*h*m^2*z + 108*a^4*b*c^4*g^2*h*
k^2*z - 18*a^4*b^2*c^3*g*h*k^3*z + 18*a^3*b^2*c^4*g^3*h*k*z + 18*a^3*b^2*c^
4*f*g^3*m*z + 9*a^3*b^4*c^2*g*h*k^3*z - 9*a^3*b^3*c^3*e*j^3*k*z - 9*a^3*b^3
*c^3*d*j^3*l*z - 144*a^4*b^3*c^2*e*g*m^3*z - 144*a^4*b^3*c^2*d*h*m^3*z - 10
8*a^3*b*c^5*e^2*g^2*m*z + 108*a^3*b*c^5*d^2*j^2*k*z - 108*a^3*b*c^5*d^2*h^2
*m*z - 18*a^2*b^3*c^4*f^3*h*k*z - 18*a^2*b^3*c^4*f^3*g*l*z - 9*a^3*b^3*c^3*
g*h*j^3*z - 216*a^4*b*c^4*d*g^2*m^2*z + 144*a^4*b^2*c^3*e*g*l^3*z - 126*a^3
*b^2*c^4*d*h^3*l*z - 108*a^4*b*c^4*d*h^2*l^2*z - 108*a^3*b*c^5*f^2*g^2*k*z
 - 108*a^3*b*c^5*e^2*h^2*k*z - 90*a^2*b^2*c^5*e^3*f*m*z + 72*a^2*b^2*c^5*e^3
*g*l*z - 63*a^3*b^4*c^2*e*g*l^3*z - 36*a^3*b^4*c^2*d*h*l^3*z + 27*a^2*b^4*c
^3*d*h^3*l*z + 27*a*b^6*c^2*d^2*g*m^2*z - 18*a^4*b^2*c^3*d*h*l^3*z - 18*a^3
*b^2*c^4*f*h^3*j*z - 18*a^3*b^2*c^4*e*h^3*k*z + 18*a^2*b^2*c^5*e^3*h*k*z +
108*a^3*b*c^5*e^2*h*j^2*z + 54*a^3*b^3*c^3*d*h*k^3*z + 27*a^3*b^3*c^3*e*g*k
^3*z - 27*a^2*b^3*c^4*e*g^3*k*z + 27*a^2*b^3*c^4*d*g^3*l*z - 27*a*b^4*c^4*d
^2*g^2*l*z - 9*a^2*b^5*c^2*e*g*k^3*z - 9*a^2*b^5*c^2*d*h*k^3*z + 207*a^3*b^
4*c^2*d*e*m^3*z - 108*a^2*b*c^6*d^2*e^2*m*z - 90*a^4*b^2*c^3*d*e*m^3*z - 72
*a^3*b^2*c^4*e*g*j^3*z - 72*a^3*b^2*c^4*d*h*j^3*z + 27*a*b^3*c^5*d^2*e^2*m*
z + 18*a^2*b^2*c^5*e*f^3*k*z + 18*a^2*b^2*c^5*d*f^3*l*z + 9*a^2*b^4*c^3*e*g
*j^3*z + 9*a^2*b^4*c^3*d*h*j^3*z - 216*a^3*b*c^5*d*e^2*l^2*z - 198*a^3*b^3*
c^3*d*e*l^3*z + 108*a^3*b*c^5*d*g^2*j^2*z - 108*a^3*b*c^5*d*f^2*k^2*z + 72*
a^2*b^5*c^2*d*e*l^3*z - 27*a*b^5*c^3*d*e^2*l^2*z + 27*a*b^4*c^4*d^2*g*j^2*z
 + 18*a^2*b^2*c^5*f^3*g*h*z + 144*a^3*b^2*c^4*d*e*k^3*z - 63*a^2*b^4*c^3*d*
e*k^3*z + 27*a*b^4*c^4*d^2*e*k^2*z - 9*a^2*b^3*c^4*e*g*h^3*z - 108*a^2*b*c^

6*d^2*g^2*h*z + 81*a^2*b^3*c^4*d*e*j^3*z + 27*a*b^3*c^5*d^2*g^2*h*z - 27*a*
b^2*c^6*d^2*e^2*j*z - 18*a^2*b^2*c^5*d*g^3*h*z + 108*a^2*b*c^6*d*e^2*h^2*z
- 27*a*b^3*c^5*d*e^2*h^2*z + 27*a*b^2*c^6*d^2*f^2*g*z - 18*a^2*b^2*c^5*d*e*
h^3*z - 216*a^6*c^3*j^2*k*l*m*z + 216*a^6*c^3*h*j*l^2*m*z + 216*a^6*c^3*f*k
*l*m^2*z - 216*a^5*c^4*f^2*k*l*m*z - 216*a^5*c^4*g^2*j*k*m*z + 216*a^5*c^4*
f*j^2*k*l*z + 216*a^5*c^4*f*h^2*l*m*z + 216*a^5*c^4*e*j^2*k*m*z + 216*a^5*c
^4*d*j^2*l*m*z + 216*a^5*c^4*g*h*j^2*m*z - 216*a^5*c^4*e*j*k^2*l*z - 216*a^
5*c^4*d*j*k^2*m*z + 216*a^4*c^5*d^2*j*k*m*z - 18*a^6*b^2*c*k*l*m^3*z + 216*
a^5*c^4*f*g*k^2*m*z - 216*a^5*c^4*d*j*k*l^2*z - 72*a^6*b*c^2*j*l^3*m*z + 18
*a^5*b^3*c*j*l^3*m*z - 216*a^5*c^4*f*h*j*l^2*z + 216*a^5*c^4*e*h*k*l^2*z +
216*a^5*c^4*e*f*l^2*m*z - 216*a^4*c^5*e^2*h*k*l*z + 216*a^4*c^5*e^2*h*j*m*z
 - 216*a^4*c^5*e^2*f*l*m*z - 216*a^5*c^4*e*f*k*m^2*z + 216*a^5*c^4*d*g*k*m^
2*z - 216*a^5*c^4*d*f*l*m^2*z + 216*a^4*c^5*e*f^2*k*m*z + 216*a^4*c^5*d*f^2
*l*m*z + 108*a^5*b*c^3*j^3*k*l*z - 216*a^5*c^4*f*g*h*m^2*z + 216*a^4*c^5*f^
2*g*h*m*z + 216*a^4*c^5*f*g^2*j*k*z - 216*a^4*c^5*e*g^2*j*l*z + 216*a^4*c^5
*d*g^2*j*m*z - 72*a^6*b*c^2*h*k*m^3*z - 72*a^6*b*c^2*g*l*m^3*z + 54*a^5*b^3
*c*h*k*m^3*z + 54*a^5*b^3*c*g*l*m^3*z - 216*a^4*c^5*d*h^2*j*k*z - 18*a^4*b^
4*c*f*l^3*m*z + 9*a^4*b^4*c*h*k*l^3*z - 216*a^4*c^5*e*f*j^2*k*z - 216*a^4*c
^5*e*f*h^2*m*z - 216*a^4*c^5*d*g*j^2*k*z - 216*a^4*c^5*d*f*j^2*l*z - 216*a^
4*c^5*d*e*j^2*m*z - 72*a^5*b*c^3*f*k^3*m*z + 72*a^4*b*c^4*g^3*j*m*z + 36*a^
5*b*c^3*g*k^3*l*z - 36*a^4*b*c^4*g^3*k*l*z - 216*a^4*c^5*f*g*h*j^2*z + 216*
a^4*c^5*d*f*j*k^2*z - 216*a^3*c^6*d^2*f*j*k*z - 216*a^3*c^6*d^2*e*j*l*z + 7
2*a^4*b^4*c*f*j*m^3*z - 63*a^4*b^4*c*e*k*m^3*z - 63*a^4*b^4*c*d*l*m^3*z + 2
16*a^4*c^5*d*g*h*k^2*z - 216*a^3*c^6*d^2*g*h*k*z + 216*a^3*c^6*d^2*f*g*m*z
- 216*a^3*c^6*d*e^2*j*k*z + 144*a^5*b*c^3*f*j*l^3*z - 144*a^3*b*c^5*e^3*j*m
*z - 72*a^5*b*c^3*e*k*l^3*z + 72*a^3*b*c^5*e^3*k*l*z - 63*a^4*b^4*c*g*h*m^3
*z + 18*a^3*b^5*c*f*j*l^3*z - 18*a*b^5*c^3*e^3*j*m*z - 9*a^3*b^5*c*e*k*l^3*
z + 9*a*b^5*c^3*e^3*k*l*z - 216*a^4*c^5*d*e*h*l^2*z - 216*a^3*c^6*e^2*f*h*j
*z + 216*a^3*c^6*d*e^2*h*l*z - 126*a*b^4*c^4*d^3*j*m*z + 108*a^4*b*c^4*g*h^
3*l*z + 63*a*b^4*c^4*d^3*k*l*z + 36*a^5*b*c^3*g*h*l^3*z - 9*a^3*b^5*c*g*h*l
^3*z + 216*a^4*c^5*d*e*f*m^2*z + 216*a^3*c^6*d*f^2*g*k*z - 216*a^3*c^6*d*e*
f^2*m*z + 36*a^4*b*c^4*e*j^3*k*z + 36*a^4*b*c^4*d*j^3*l*z - 216*a^3*c^6*d*f
*g^2*j*z + 72*a^3*b^5*c*e*g*m^3*z + 72*a^3*b^5*c*d*h*m^3*z + 72*a^3*b*c^5*f
^3*h*k*z + 72*a^3*b*c^5*f^3*g*l*z + 36*a^4*b*c^4*g*h*j^3*z + 18*a*b^4*c^4*e
^3*f*m*z + 9*a^2*b^6*c*e*g*l^3*z + 9*a^2*b^6*c*d*h*l^3*z - 9*a*b^4*c^4*e^3*
h*k*z - 9*a*b^4*c^4*e^3*g*l*z + 216*a^3*c^6*d*e*f*j^2*z - 144*a^2*b*c^6*d^3
*f*m*z + 108*a^3*b*c^5*e*g^3*k*z - 108*a^3*b*c^5*d*g^3*l*z + 108*a*b^3*c^5*
d^3*f*m*z - 72*a^4*b*c^4*d*h*k^3*z + 72*a^2*b*c^6*d^3*h*k*z - 54*a*b^3*c^5*
d^3*h*k*z + 36*a^4*b*c^4*e*g*k^3*z - 36*a^2*b*c^6*d^3*g*l*z - 27*a*b^3*c^5*
d^3*g*l*z - 81*a^2*b^6*c*d*e*m^3*z + 216*a^4*b*c^4*d*e*l^3*z + 72*a^2*b*c^6
*e^3*f*j*z + 72*a^2*b*c^6*d*e^3*l*z - 18*a*b^3*c^5*e^3*f*j*z - 18*a*b^3*c^5
*d*e^3*l*z - 90*a*b^2*c^6*d^3*f*j*z + 72*a*b^2*c^6*d^3*e*k*z + 36*a^3*b*c^5
*e*g*h^3*z - 36*a^2*b*c^6*e^3*g*h*z + 9*a*b^6*c^2*d*e*k^3*z + 9*a*b^3*c^5*e
^3*g*h*z - 180*a^3*b*c^5*d*e*j^3*z + 18*a*b^2*c^6*d^3*g*h*z - 9*a*b^5*c^3*d
*e*j^3*z + 18*a*b^2*c^6*d*e^3*h*z + 9*a*b^4*c^4*d*e*h^3*z + 36*a^2*b*c^6*d*

e*g^3*z - 9*a*b^3*c^5*d*e*g^3*z - 18*a*b^2*c^6*d*e*f^3*z + 27*a^5*b^2*c^2*h
^2*l*m^2*z - 27*a^5*b^2*c^2*j*k^2*l^2*z + 27*a^4*b^3*c^2*h^2*k^2*m*z + 27*a
^4*b^3*c^2*g^2*l^2*m*z + 27*a^5*b^2*c^2*g*k^2*m^2*z - 27*a^4*b^3*c^2*h^2*k*
l^2*z - 27*a^4*b^3*c^2*g^2*k*m^2*z - 135*a^4*b^2*c^3*e^2*l*m^2*z + 27*a^5*b
^2*c^2*e*l^2*m^2*z + 27*a^4*b^3*c^2*h*j^2*l^2*z - 27*a^4*b^2*c^3*h^2*j^2*l*
z + 27*a^3*b^4*c^2*e^2*l*m^2*z - 270*a^4*b^3*c^2*f*j^2*m^2*z - 270*a^4*b^2*
c^3*f^2*j*m^2*z + 162*a^3*b^4*c^2*f^2*j*m^2*z - 108*a^3*b^3*c^3*f^2*j^2*m*z
 - 27*a^4*b^2*c^3*h^2*j*k^2*z - 27*a^4*b^2*c^3*g^2*j*l^2*z + 27*a^3*b^3*c^3
*e^2*k^2*m*z + 27*a^3*b^3*c^3*d^2*l^2*m*z + 27*a^2*b^5*c^2*f^2*j^2*m*z + 16
2*a^3*b^3*c^3*d^2*k*m^2*z - 27*a^4*b^3*c^2*d*k^2*m^2*z - 27*a^4*b^2*c^3*g*j
^2*k^2*z + 27*a^3*b^3*c^3*g^2*h^2*m*z - 27*a^2*b^5*c^2*d^2*k*m^2*z + 162*a^
3*b^2*c^4*d^2*k^2*l*z - 108*a^4*b^2*c^3*g*h^2*l^2*z - 27*a^4*b^2*c^3*e*j^2*
l^2*z + 27*a^3*b^4*c^2*g*h^2*l^2*z + 27*a^3*b^2*c^4*e^2*j^2*l*z - 27*a^2*b^
4*c^3*d^2*k^2*l*z - 162*a^3*b^3*c^3*f^2*h*l^2*z + 162*a^3*b^3*c^3*e^2*h*m^2
*z - 135*a^4*b^2*c^3*e*h^2*m^2*z + 135*a^3*b^2*c^4*f^2*h^2*l*z + 27*a^3*b^4
*c^2*e*h^2*m^2*z - 27*a^3*b^3*c^3*g^2*h*k^2*z - 27*a^3*b^2*c^4*e^2*j*k^2*z
 - 27*a^3*b^2*c^4*d^2*j*l^2*z + 27*a^2*b^5*c^2*f^2*h*l^2*z - 27*a^2*b^5*c^2*
e^2*h*m^2*z - 27*a^2*b^4*c^3*f^2*h^2*l*z - 27*a^3*b^2*c^4*g^2*h^2*j*z + 27*
a^2*b^3*c^4*e^2*g^2*m*z - 27*a^2*b^3*c^4*d^2*j^2*k*z + 27*a^2*b^3*c^4*d^2*h
^2*m*z + 351*a^3*b^2*c^4*d^2*g*m^2*z - 189*a^2*b^4*c^3*d^2*g*m^2*z + 162*a^
3*b^3*c^3*d*g^2*m^2*z - 162*a^3*b^2*c^4*e^2*g*l^2*z + 135*a^3*b^3*c^3*d*h^2
*l^2*z + 135*a^3*b^2*c^4*f^2*g*k^2*z - 27*a^2*b^5*c^2*d*h^2*l^2*z - 27*a^2*
b^5*c^2*d*g^2*m^2*z - 27*a^2*b^4*c^3*f^2*g*k^2*z + 27*a^2*b^4*c^3*e^2*g*l^2
*z + 27*a^2*b^3*c^4*f^2*g^2*k*z + 27*a^2*b^3*c^4*e^2*h^2*k*z + 135*a^3*b^2*
c^4*e*f^2*l^2*z - 108*a^3*b^2*c^4*e*g^2*k^2*z + 108*a^2*b^2*c^5*d^2*g^2*l*z
 + 27*a^3*b^2*c^4*e*h^2*j^2*z + 27*a^2*b^4*c^3*e*g^2*k^2*z - 27*a^2*b^4*c^3
*e*f^2*l^2*z - 27*a^2*b^3*c^4*e^2*h*j^2*z - 27*a^2*b^2*c^5*e^2*f^2*l*z - 27
*a^2*b^2*c^5*e^2*g^2*j*z - 27*a^2*b^2*c^5*d^2*h^2*j*z + 162*a^2*b^3*c^4*d*e
^2*l^2*z - 135*a^2*b^2*c^5*d^2*g*j^2*z - 27*a^2*b^3*c^4*d*g^2*j^2*z + 27*a^
2*b^3*c^4*d*f^2*k^2*z - 162*a^2*b^2*c^5*d^2*e*k^2*z - 27*a^2*b^2*c^5*e*f^2*
h^2*z - 72*a^7*c^2*k*l*m^3*z + 9*a^5*b^4*k*l*m^3*z + 72*a^6*c^3*j*k^3*m*z -
 72*a^6*c^3*h*k*l^3*z - 72*a^6*c^3*f*l^3*m*z - 72*a^5*c^4*h^3*k*l*z - 72*a^
5*c^4*h^3*j*m*z - 9*a^4*b^5*h*k*m^3*z - 9*a^4*b^5*g*l*m^3*z - 144*a^6*c^3*f
*j*m^3*z - 144*a^5*c^4*h*j^3*k*z - 144*a^5*c^4*g*j^3*l*z - 144*a^5*c^4*f*j^
3*m*z - 144*a^4*c^5*f^3*j*m*z + 72*a^6*c^3*e*k*m^3*z + 72*a^6*c^3*d*l*m^3*z
 + 72*a^4*c^5*f^3*k*l*z + 72*a^6*c^3*g*h*m^3*z + 18*b^6*c^3*d^3*j*m*z - 18*
a^3*b^6*f*j*m^3*z - 9*b^6*c^3*d^3*k*l*z + 9*a^3*b^6*e*k*m^3*z + 9*a^3*b^6*d
*l*m^3*z + 144*a^5*c^4*d*k^3*l*z + 144*a^3*c^6*d^3*k*l*z - 72*a^5*c^4*f*j*k
^3*z - 72*a^3*c^6*d^3*j*m*z + 9*a^3*b^6*g*h*m^3*z - 72*a^5*c^4*g*h*k^3*z -
72*a^4*c^5*g^3*h*k*z - 72*a^4*c^5*f*g^3*m*z - 108*a^5*b*c^3*j^4*m*z + 63*a^
6*b^2*c*j*m^4*z + 36*a^6*b*c^2*k*l^4*z - 9*a^5*b^3*c*k*l^4*z - 144*a^5*c^4*
e*g*l^3*z - 144*a^3*c^6*e^3*g*l*z + 72*a^5*c^4*d*h*l^3*z + 72*a^4*c^5*f*h^3
*j*z + 72*a^4*c^5*e*h^3*k*z + 72*a^4*c^5*d*h^3*l*z + 72*a^3*c^6*e^3*h*k*z +
 72*a^3*c^6*e^3*f*m*z - 18*b^5*c^4*d^3*f*m*z + 9*b^5*c^4*d^3*h*k*z + 9*b^5*
c^4*d^3*g*l*z - 9*a^2*b^7*e*g*m^3*z - 9*a^2*b^7*d*h*m^3*z + 144*a^4*c^5*e*g

```
*j^3*z + 144*a^4*c^5*d*h*j^3*z - 72*a^5*c^4*d*e*m^3*z - 72*a^3*c^6*e*f^3*k*
z - 72*a^3*c^6*d*f^3*l*z + 144*a^6*b*c^2*f*m^4*z - 108*a^5*b^3*c*f*m^4*z -
72*a^3*c^6*f^3*g*h*z + 36*a^5*b*c^3*h*k^4*z - 36*a^3*b*c^5*f^4*m*z + 18*b^4
*c^5*d^3*f*j*z - 9*b^4*c^5*d^3*e*k*z + 9*a^4*b^4*c*g*l^4*z - 144*a^4*c^5*d*
e*k^3*z - 144*a^2*c^7*d^3*e*k*z + 72*a^2*c^7*d^3*f*j*z - 9*b^4*c^5*d^3*g*h*
z + 72*a^3*c^6*d*g^3*h*z + 72*a^2*c^7*d^3*g*h*z - 72*a^5*b*c^3*d*l^4*z - 72
*a^4*b*c^4*f*j^4*z + 45*a*b^2*c^6*d^4*l*z - 36*a^2*b*c^6*e^4*k*z - 9*a^3*b^
5*c*d*l^4*z + 9*a*b^3*c^5*e^4*k*z - 72*a^3*c^6*d*e*h^3*z - 72*a^2*c^7*d*e^3
*h*z + 9*b^3*c^6*d^3*e*g*z + 72*a^2*c^7*d*e*f^3*z + 36*a^3*b*c^5*d*h^4*z -
9*a*b^2*c^6*e^4*g*z + 36*a*b*c^7*d^3*f^2*z + 90*a^5*b^2*c^2*j^3*m^2*z + 45*
a^5*b^2*c^2*j^2*l^3*z + 9*a^4*b^3*c^2*j^2*k^3*z - 9*a^4*b^3*c^2*h^3*m^2*z -
 45*a^4*b^2*c^3*g^3*m^2*z + 9*a^3*b^4*c^2*g^3*m^2*z + 198*a^4*b^3*c^2*f^2*m
^3*z - 108*a^3*b^3*c^3*f^3*m^2*z + 18*a^2*b^5*c^2*f^3*m^2*z - 117*a^4*b^2*c
^3*f^2*l^3*z + 117*a^3*b^2*c^4*e^3*m^2*z + 63*a^3*b^4*c^2*f^2*l^3*z - 63*a^
2*b^4*c^3*e^3*m^2*z - 171*a^2*b^3*c^4*d^3*m^2*z - 54*a^3*b^3*c^3*f^2*k^3*z
+ 9*a^3*b^2*c^4*g^3*j^2*z + 9*a^2*b^5*c^2*f^2*k^3*z + 18*a^3*b^2*c^4*f^2*j^
3*z + 18*a^2*b^3*c^4*f^3*j^2*z - 9*a^2*b^4*c^3*f^2*j^3*z - 45*a^2*b^2*c^5*e
^3*j^2*z + 9*a^2*b^3*c^4*f^2*h^3*z - 9*a^2*b^2*c^5*f^2*g^3*z + 9*a*b^8*d*e*
m^3*z - 36*a*b*c^7*d^4*h*z - 108*a^6*c^3*h^2*l*m^2*z + 108*a^6*c^3*j*k^2*l^
2*z - 108*a^6*c^3*g*k^2*m^2*z - 108*a^6*c^3*e*l^2*m^2*z + 108*a^5*c^4*h^2*j
^2*l*z + 108*a^5*c^4*e^2*l*m^2*z + 216*a^5*c^4*f^2*j*m^2*z + 108*a^5*c^4*h^
2*j*k^2*z + 108*a^5*c^4*g^2*j*l^2*z + 108*a^5*c^4*g*j^2*k^2*z - 216*a^4*c^5
*d^2*k^2*l*z + 108*a^5*c^4*e*j^2*l^2*z - 108*a^4*c^5*e^2*j^2*l*z - 9*a^6*b^
2*c*l^3*m^2*z + 108*a^5*c^4*e*h^2*m^2*z - 108*a^4*c^5*f^2*h^2*l*z + 108*a^4
*c^5*e^2*j*k^2*z + 108*a^4*c^5*d^2*j*l^2*z - 144*a^6*b*c^2*j^2*m^3*z + 108*
a^4*c^5*g^2*h^2*j*z - 27*a^4*b^4*c*j^3*m^2*z + 27*a^4*b^3*c^2*j^4*m*z + 9*a
^5*b^2*c^2*k^4*l*z + 216*a^4*c^5*e^2*g*l^2*z - 108*a^4*c^5*f^2*g*k^2*z - 10
8*a^4*c^5*d^2*g*m^2*z - 9*a^4*b^4*c*j^2*l^3*z - 108*a^4*c^5*e*h^2*j^2*z - 1
08*a^4*c^5*e*f^2*l^2*z + 108*a^3*c^6*e^2*f^2*l*z - 36*a^5*b*c^3*j^2*k^3*z +
 36*a^5*b*c^3*h^3*m^2*z + 108*a^3*c^6*e^2*g^2*j*z + 108*a^3*c^6*d^2*h^2*j*z
 - 216*a^5*b*c^3*f^2*m^3*z + 144*a^4*b*c^4*f^3*m^2*z + 108*a^3*c^6*d^2*g*j^
2*z - 72*a^3*b^5*c*f^2*m^3*z - 45*a^5*b^2*c^2*g*l^4*z - 9*a^4*b^3*c^2*h*k^4
*z - 9*a^3*b^2*c^4*g^4*l*z + 9*a^2*b^3*c^4*f^4*m*z + 216*a^3*c^6*d^2*e*k^2*
z - 9*a^2*b^6*c*f^2*l^3*z + 9*a*b^6*c^2*e^3*m^2*z + 108*a^3*c^6*e*f^2*h^2*z
 + 108*a^3*b*c^5*d^3*m^2*z + 108*a^2*c^7*d^2*e^2*j*z + 72*a^4*b*c^4*f^2*k^3
*z + 72*a*b^5*c^3*d^3*m^2*z - 72*a^3*b*c^5*f^3*j^2*z + 54*a^4*b^3*c^2*d*l^4
*z - 45*a^4*b^2*c^3*e*k^4*z + 18*a^3*b^3*c^3*f*j^4*z + 9*a^3*b^4*c^2*e*k^4*
z - 9*a^2*b^2*c^5*f^4*j*z - 108*a^2*c^7*d^2*f^2*g*z + 9*a^3*b^2*c^4*g*h^4*z
 + 9*a*b^4*c^4*e^3*j^2*z - 72*a^2*b*c^6*d^3*j^2*z + 54*a*b^3*c^5*d^3*j^2*z
- 36*a^3*b*c^5*f^2*h^3*z - 9*a^2*b^3*c^4*d*h^4*z + 9*a^2*b^2*c^5*e*g^4*z +
9*a*b^2*c^6*e^3*f^2*z + 36*a^7*c^2*l^3*m^2*z + 72*a^6*c^3*j^3*m^2*z - 36*a^
6*c^3*j^2*l^3*z + 9*a^4*b^5*j^2*m^3*z + 36*a^5*c^4*g^3*m^2*z + 36*a^5*c^4*f
^2*l^3*z - 36*a^4*c^5*e^3*m^2*z - 9*b^7*c^2*d^3*m^2*z + 9*a^2*b^7*f^2*m^3*z
 - 36*a^4*c^5*g^3*j^2*z + 72*a^4*c^5*f^2*j^3*z + 36*a^3*c^6*e^3*j^2*z - 9*b
^5*c^4*d^3*j^2*z + 36*a^3*c^6*f^2*g^3*z - 9*a^4*b^2*c^3*j^5*z - 36*a^2*c^7*
```

e^3*f^2*z - 9*b^3*c^6*d^3*f^2*z + 36*a^7*c^2*j*m^4*z - 36*a^6*c^3*k^4*l*z -
18*a^5*b^4*j*m^4*z + 36*a^6*c^3*g*l^4*z + 36*a^4*c^5*g^4*l*z + 18*a^4*b^5*
f*m^4*z - 9*b^4*c^5*d^4*l*z + 36*a^5*c^4*e*k^4*z + 36*a^3*c^6*f^4*j*z - 36*
a^2*c^7*d^4*l*z - 36*a^4*c^5*g*h^4*z + 9*b^3*c^6*d^4*h*z - 36*a^3*c^6*e*g^4
*z + 36*a^2*c^7*e^4*g*z - 9*b^2*c^7*d^4*e*z - 36*a^7*b*c*m^5*z + 36*a*c^8*d
^4*e*z + 9*a^6*b^3*m^5*z + 36*a^5*c^4*j^5*z + 9*a^4*b^3*c*g*h*j*k*l*m - 9*a
^3*b^4*c*e*g*j*k*l*m - 9*a^3*b^4*c*d*h*j*k*l*m - 9*a^3*b^4*c*f*g*h*k*l*m +
36*a^4*b*c^3*d*e*j*k*l*m + 9*a^2*b^5*c*d*e*j*k*l*m + 36*a^4*b*c^3*e*f*h*j*l
*m + 36*a^4*b*c^3*e*f*g*k*l*m + 36*a^4*b*c^3*d*f*h*k*l*m + 9*a^2*b^5*c*e*f*
g*k*l*m + 9*a^2*b^5*c*d*f*h*k*l*m + 36*a^3*b*c^4*d*e*f*j*k*l + 9*a*b^5*c^2*
d*e*f*j*k*l + 36*a^3*b*c^4*d*e*g*h*k*l + 36*a^3*b*c^4*d*e*f*h*k*m + 36*a^3*
b*c^4*d*e*f*g*l*m + 9*a*b^5*c^2*d*e*f*h*k*m + 9*a*b^5*c^2*d*e*f*g*l*m - 9*a
*b^4*c^3*d*e*f*h*j*k - 9*a*b^4*c^3*d*e*f*g*j*l - 9*a*b^4*c^3*d*e*f*g*h*m +
9*a*b^3*c^4*d*e*f*g*h*j - 9*a*b^6*c*d*e*f*k*l*m + 18*a^4*b^2*c^2*e*g*j*k*l*
m + 18*a^4*b^2*c^2*d*h*j*k*l*m + 18*a^4*b^2*c^2*f*g*h*k*l*m - 36*a^3*b^3*c^
2*d*e*j*k*l*m - 36*a^3*b^3*c^2*e*f*g*k*l*m - 36*a^3*b^3*c^2*d*f*h*k*l*m + 9
*a^3*b^3*c^2*f*g*h*j*k*l + 9*a^3*b^3*c^2*e*g*h*j*k*m + 9*a^3*b^3*c^2*d*g*h*
j*l*m - 108*a^3*b^2*c^3*d*e*f*k*l*m + 54*a^2*b^4*c^2*d*e*f*k*l*m - 36*a^3*b
^2*c^3*d*f*g*j*k*m + 18*a^3*b^2*c^3*e*f*g*j*k*l + 18*a^3*b^2*c^3*d*f*h*j*k*
l + 18*a^3*b^2*c^3*d*e*h*j*k*m + 18*a^3*b^2*c^3*d*e*g*j*l*m - 9*a^2*b^4*c^2
*e*f*g*j*k*l - 9*a^2*b^4*c^2*d*f*h*j*k*l - 9*a^2*b^4*c^2*d*e*h*j*k*m - 9*a^
2*b^4*c^2*d*e*g*j*l*m + 18*a^3*b^2*c^3*e*f*g*h*k*m + 18*a^3*b^2*c^3*d*f*g*h
*l*m - 9*a^2*b^4*c^2*e*f*g*h*k*m - 9*a^2*b^4*c^2*d*f*g*h*l*m - 36*a^2*b^3*c
^3*d*e*f*j*k*l - 36*a^2*b^3*c^3*d*e*f*h*k*m - 36*a^2*b^3*c^3*d*e*f*g*l*m +
9*a^2*b^3*c^3*e*f*g*h*j*k + 9*a^2*b^3*c^3*d*f*g*h*j*l + 9*a^2*b^3*c^3*d*e*g
*h*j*m + 18*a^2*b^2*c^4*d*e*f*h*j*k + 18*a^2*b^2*c^4*d*e*f*g*j*l + 18*a^2*b
^2*c^4*d*e*f*g*h*m - 9*a^5*b^2*c*h*j*k^2*l*m - 9*a^5*b^2*c*g*j*k*l^2*m + 27
*a^5*b^2*c*f*j*k*l*m^2 - 9*a^4*b^3*c*f*j^2*k*l*m + 9*a^3*b^4*c*f^2*j*k*l*m
- 18*a^5*b*c^2*e*j*k^2*l*m - 9*a^5*b^2*c*g*h*k*l*m^2 + 9*a^4*b^3*c*e*j*k^2*
l*m - 18*a^5*b*c^2*f*h*k^2*l*m - 18*a^5*b*c^2*d*j*k*l^2*m + 9*a^4*b^3*c*f*h
*k^2*l*m + 9*a^4*b^3*c*d*j*k*l^2*m + 36*a^5*b*c^2*e*h*k*l^2*m - 36*a^4*b*c^
3*e^2*h*k*l*m + 18*a^5*b*c^2*f*h*j*l^2*m - 18*a^5*b*c^2*f*g*k*l^2*m - 18*a^
4*b^3*c*e*h*k*l^2*m + 9*a^4*b^3*c*f*g*k*l^2*m + 9*a^3*b^4*c*e*h^2*k*l*m - 9
*a^2*b^5*c*e^2*h*k*l*m - 54*a^5*b*c^2*e*h*j*l*m^2 - 18*a^5*b*c^2*e*g*k*l*m^
2 - 18*a^5*b*c^2*d*h*k*l*m^2 + 18*a^4*b^3*c*e*h*j*l*m^2 - 9*a^4*b^3*c*f*h*j
*k*m^2 - 9*a^4*b^3*c*f*g*j*l*m^2 + 9*a^4*b^3*c*e*g*k*l*m^2 + 9*a^4*b^3*c*d*
h*k*l*m^2 + 18*a^4*b*c^3*f*g^2*j*k*m - 18*a^4*b*c^3*e*g^2*j*l*m + 18*a^3*b^
4*c*d*g*k^2*l*m - 9*a^3*b^4*c*e*f*k^2*l*m - 9*a^2*b^5*c*d*g^2*k*l*m - 18*a^
4*b*c^3*f*g^2*h*l*m - 18*a^4*b*c^3*d*h^2*j*k*m - 9*a^3*b^4*c*d*f*k*l^2*m -
54*a^4*b*c^3*d*g*j^2*k*m - 18*a^4*b*c^3*f*g*h^2*k*m - 18*a^4*b*c^3*e*g*j^2*
k*l - 18*a^4*b*c^3*d*h*j^2*k*l - 18*a^3*b^4*c*d*g*j*k*m^2 + 9*a^3*b^4*c*e*f
*j*k*m^2 + 9*a^3*b^4*c*d*f*j*l*m^2 - 9*a^3*b^4*c*d*e*k*l*m^2 - 54*a^3*b*c^4
*d^2*f*j*k*m + 36*a^4*b*c^3*d*g*j*k^2*l - 36*a^3*b*c^4*d^2*g*j*k*l - 18*a^4
*b*c^3*e*f*j*k^2*l + 18*a^4*b*c^3*d*f*j*k^2*m - 18*a^3*b*c^4*d^2*e*j*l*m +
9*a^3*b^4*c*f*g*h*j*m^2 - 9*a*b^5*c^2*d^2*g*j*k*l + 36*a^4*b*c^3*d*g*h*k^2*

m - 36*a^3*b*c^4*d^2*g*h*k*m + 18*a^4*b*c^3*e*g*h*k^2*l - 18*a^4*b*c^3*e*f*
h*k^2*m - 18*a^4*b*c^3*d*f*j*k*l^2 - 18*a^3*b*c^4*d^2*f*h*l*m - 18*a^3*b*c^
4*d*e^2*j*k*m - 9*a*b^5*c^2*d^2*g*h*k*m - 54*a^4*b*c^3*d*g*h*k*l^2 - 54*a^3
*b*c^4*e^2*f*h*j*m - 18*a^4*b*c^3*d*f*g*l^2*m - 18*a^3*b*c^4*e^2*f*g*k*m -
54*a^4*b*c^3*d*f*g*k*m^2 - 36*a^4*b*c^3*e*f*g*j*m^2 - 36*a^4*b*c^3*d*f*h*j*
m^2 + 36*a^3*b*c^4*e*f^2*g*j*m + 36*a^3*b*c^4*d*f^2*h*j*m - 18*a^4*b*c^3*d*
e*h*k*m^2 - 18*a^4*b*c^3*d*e*g*l*m^2 + 18*a^3*b*c^4*e*f^2*h*j*l - 18*a^3*b*
c^4*e*f^2*g*k*l - 18*a^3*b*c^4*d*f^2*h*k*l + 18*a^3*b*c^4*d*f^2*g*k*m - 9*a
^2*b^5*c*e*f*g*j*m^2 - 9*a^2*b^5*c*d*f*h*j*m^2 - 54*a^3*b*c^4*d*f*g^2*j*m -
18*a^3*b*c^4*e*f*g^2*j*l - 18*a*b^4*c^3*d^2*f*g*j*m + 9*a*b^4*c^3*d^2*g*h*
j*k + 9*a*b^4*c^3*d^2*f*g*k*l + 9*a*b^4*c^3*d^2*e*g*k*m - 9*a*b^4*c^3*d^2*e
*f*l*m - 18*a^3*b*c^4*e*f*g^2*h*m - 18*a^3*b*c^4*d*f*h^2*j*k - 9*a*b^4*c^3*
d*e^2*f*k*m + 18*a^3*b*c^4*d*f*g*j^2*k - 18*a^3*b*c^4*d*f*g*h^2*m - 18*a^3*
b*c^4*d*e*h*j^2*k - 18*a^3*b*c^4*d*e*g*j^2*l + 18*a*b^4*c^3*d*e*f^2*j*m - 9
*a*b^5*c^2*d*e*f*j^2*m - 9*a*b^4*c^3*d*e*f^2*k*l - 18*a^2*b*c^5*d^2*e*f*j*l
- 9*a*b^3*c^4*d^2*e*g*j*k + 9*a*b^3*c^4*d^2*e*f*j*l - 54*a^2*b*c^5*d^2*e*g
*h*l - 18*a^2*b*c^5*d^2*e*f*h*m - 18*a^2*b*c^5*d*e^2*f*j*k + 18*a*b^3*c^4*d
^2*e*g*h*l - 9*a*b^3*c^4*d^2*f*g*h*k + 9*a*b^3*c^4*d^2*e*f*h*m + 9*a*b^3*c^
4*d*e^2*f*j*k - 36*a^3*b*c^4*d*e*f*h*l^2 + 36*a^2*b*c^5*d*e^2*f*h*l + 18*a^
2*b*c^5*d*e^2*g*h*k - 18*a^2*b*c^5*d*e^2*f*g*m - 18*a*b^3*c^4*d*e^2*f*h*l -
9*a*b^5*c^2*d*e*f*h*l^2 + 9*a*b^4*c^3*d*e*f*h^2*l + 9*a*b^3*c^4*d*e^2*f*g*
m - 18*a^2*b*c^5*d*e*f^2*h*k - 18*a^2*b*c^5*d*e*f^2*g*l + 9*a*b^3*c^4*d*e*f
^2*h*k + 9*a*b^3*c^4*d*e*f^2*g*l + 27*a*b^2*c^5*d^2*e*f*g*k + 9*a*b^4*c^3*d
*e*f*g*k^2 - 9*a*b^3*c^4*d*e*f*g^2*k - 9*a*b^2*c^5*d^2*e*f*h*j - 9*a*b^2*c^
5*d*e^2*f*g*j - 9*a*b^2*c^5*d*e*f^2*g*h + 72*a^4*c^4*d*f*g*j*k*m + 72*a^4*c
^4*d*e*f*k*l*m + 9*a*b^6*c*d^2*g*k*l*m + 9*a*b^6*c*d*e*f*j*m^2 - 27*a^4*b^2
*c^2*f^2*j*k*l*m - 9*a^4*b^2*c^2*g^2*h*j*l*m + 36*a^3*b^3*c^2*e^2*h*k*l*m -
18*a^4*b^2*c^2*e*h^2*k*l*m - 9*a^4*b^2*c^2*g*h^2*j*k*m + 18*a^4*b^2*c^2*f*
h*j^2*k*m + 18*a^4*b^2*c^2*f*g*j^2*l*m - 18*a^4*b^2*c^2*e*h*j^2*l*m - 9*a^4
*b^2*c^2*g*h*j^2*k*l - 9*a^3*b^3*c^2*f^2*h*j*k*m - 9*a^3*b^3*c^2*f^2*g*j*l*
m - 63*a^4*b^2*c^2*d*g*k^2*l*m + 63*a^3*b^2*c^3*d^2*g*k*l*m - 45*a^2*b^4*c^
2*d^2*g*k*l*m + 36*a^4*b^2*c^2*e*f*k^2*l*m + 27*a^3*b^3*c^2*d*g^2*k*l*m - 9
*a^4*b^2*c^2*f*h*j*k^2*l - 9*a^4*b^2*c^2*e*h*j*k^2*m + 9*a^3*b^3*c^2*e*g^2*
j*l*m - 9*a^3*b^2*c^3*d^2*h*j*l*m + 36*a^4*b^2*c^2*d*f*k*l^2*m + 27*a^4*b^2
*c^2*e*h*j*k*l^2 - 27*a^3*b^2*c^3*e^2*h*j*k*l - 18*a^3*b^2*c^3*e^2*f*j*l*m
- 9*a^4*b^2*c^2*f*g*j*k*l^2 - 9*a^4*b^2*c^2*d*g*j*l^2*m + 9*a^3*b^3*c^2*f*g
^2*h*l*m - 9*a^3*b^3*c^2*e*h^2*j*k*l + 9*a^3*b^3*c^2*d*h^2*j*k*m - 9*a^3*b^
2*c^3*e^2*g*j*k*m + 9*a^2*b^4*c^2*e^2*h*j*k*l + 72*a^4*b^2*c^2*d*g*j*k*m^2
+ 36*a^4*b^2*c^2*d*e*k*l*m^2 + 27*a^4*b^2*c^2*e*g*h*l^2*m - 27*a^4*b^2*c^2*
e*f*j*k*m^2 - 27*a^4*b^2*c^2*d*f*j*l*m^2 - 27*a^3*b^2*c^3*e^2*g*h*l*m + 27*
a^3*b^2*c^3*e*f^2*j*k*m + 27*a^3*b^2*c^3*d*f^2*j*l*m + 18*a^3*b^3*c^2*d*g*j
^2*k*m + 9*a^3*b^3*c^2*f*g*h^2*k*m + 9*a^3*b^3*c^2*e*g*j^2*k*l - 9*a^3*b^3*
c^2*e*g*h^2*l*m - 9*a^3*b^3*c^2*e*f*j^2*k*m + 9*a^3*b^3*c^2*d*h*j^2*k*l - 9
*a^3*b^3*c^2*d*f*j^2*l*m + 9*a^2*b^4*c^2*e^2*g*h*l*m + 36*a^2*b^3*c^3*d^2*g
*j*k*l - 27*a^4*b^2*c^2*f*g*h*j*m^2 + 27*a^3*b^2*c^3*f^2*g*h*j*m - 18*a^4*b

```
^2*c^2*e*f*h*l*m^2 - 18*a^3*b^3*c^2*d*g*j*k^2*l - 18*a^3*b^2*c^3*d*g^2*j*k*
l + 18*a^2*b^3*c^3*d^2*f*j*k*m - 9*a^4*b^2*c^2*e*g*h*k*m^2 - 9*a^4*b^2*c^2*
d*g*h*l*m^2 - 9*a^3*b^3*c^2*f*g*h*j^2*m + 9*a^3*b^3*c^2*e*f*j*k^2*l - 9*a^3
*b^2*c^3*f^2*g*h*k*l + 9*a^2*b^4*c^2*d*g^2*j*k*l + 9*a^2*b^3*c^3*d^2*e*j*l*
m + 36*a^3*b^2*c^3*e*f*g^2*l*m + 36*a^2*b^3*c^3*d^2*g*h*k*m - 18*a^3*b^3*c^
2*d*g*h*k^2*m - 18*a^3*b^2*c^3*d*g^2*h*k*m + 9*a^3*b^3*c^2*e*f*h*k^2*m + 9*
a^3*b^3*c^2*d*f*j*k*l^2 - 9*a^3*b^2*c^3*f*g^2*h*j*l - 9*a^3*b^2*c^3*e*g^2*h
*j*m - 9*a^2*b^4*c^2*e*f*g^2*l*m + 9*a^2*b^4*c^2*d*g^2*h*k*m + 9*a^2*b^3*c^
3*d^2*f*h*l*m + 9*a^2*b^3*c^3*d*e^2*j*k*m + 36*a^3*b^2*c^3*d*f*h^2*k*m + 36
*a^3*b^2*c^3*d*e*j^2*k*l + 18*a^3*b^3*c^2*d*g*h*k*l^2 + 18*a^3*b^2*c^3*e*g*
h^2*j*l + 18*a^3*b^2*c^3*e*f*h^2*k*l - 18*a^3*b^2*c^3*e*f*h^2*j*m - 18*a^3*
b^2*c^3*d*g*h^2*k*l + 18*a^3*b^2*c^3*d*e*h^2*l*m + 18*a^2*b^3*c^3*e^2*f*h*j
*m - 9*a^3*b^3*c^2*e*g*h*j*l^2 - 9*a^3*b^3*c^2*e*f*h*k*l^2 + 9*a^3*b^3*c^2*
d*f*g*l^2*m - 9*a^3*b^3*c^2*d*e*h*l^2*m - 9*a^3*b^2*c^3*f*g*h^2*j*k - 9*a^3
*b^2*c^3*d*g*h^2*j*m - 9*a^2*b^4*c^2*d*f*h^2*k*m - 9*a^2*b^4*c^2*d*e*j^2*k*
l - 9*a^2*b^3*c^3*e^2*g*h*j*l - 9*a^2*b^3*c^3*e^2*f*h*k*l + 9*a^2*b^3*c^3*e
^2*f*g*k*m - 9*a^2*b^3*c^3*d*e^2*h*l*m + 36*a^3*b^3*c^2*e*f*g*j*m^2 + 36*a^
3*b^3*c^2*d*f*h*j*m^2 + 18*a^3*b^3*c^2*d*f*g*k*m^2 - 18*a^3*b^2*c^3*e*f*g*j
^2*m - 18*a^3*b^2*c^3*d*f*h*j^2*m - 18*a^2*b^3*c^3*e*f^2*g*j*m - 18*a^2*b^3
*c^3*d*f^2*h*j*m + 9*a^3*b^3*c^2*d*e*h*k*m^2 + 9*a^3*b^3*c^2*d*e*g*l*m^2 -
9*a^3*b^2*c^3*e*g*h*j^2*k - 9*a^3*b^2*c^3*d*g*h*j^2*l + 9*a^2*b^4*c^2*e*f*g
*j^2*m + 9*a^2*b^4*c^2*d*f*h*j^2*m + 9*a^2*b^3*c^3*e*f^2*g*k*l + 9*a^2*b^3*
c^3*d*f^2*h*k*l + 72*a^2*b^2*c^4*d^2*f*g*j*m + 36*a^2*b^2*c^4*d^2*e*f*l*m +
 27*a^3*b^2*c^3*d*g*h*j*k^2 + 27*a^3*b^2*c^3*d*f*g*k^2*l + 27*a^3*b^2*c^3*d
*e*g*k^2*m - 27*a^2*b^2*c^4*d^2*g*h*j*k - 27*a^2*b^2*c^4*d^2*f*g*k*l - 27*a
^2*b^2*c^4*d^2*e*g*k*m + 18*a^2*b^3*c^3*d*f*g^2*j*m - 18*a^2*b^2*c^4*d^2*e*
h*k*l - 9*a^3*b^2*c^3*e*f*h*j*k^2 + 9*a^2*b^3*c^3*e*f*g^2*j*l - 9*a^2*b^3*c
^3*d*g^2*h*j*k - 9*a^2*b^3*c^3*d*f*g^2*k*l - 9*a^2*b^3*c^3*d*e*g^2*k*m - 9*
a^2*b^2*c^4*d^2*f*h*j*l - 9*a^2*b^2*c^4*d^2*e*h*j*m + 36*a^2*b^2*c^4*d*e^2*
f*k*m - 27*a^3*b^2*c^3*d*e*h*j*l^2 + 27*a^2*b^2*c^4*d*e^2*h*j*l - 18*a^3*b^
2*c^3*d*e*g*k*l^2 - 9*a^3*b^2*c^3*d*f*g*j*l^2 + 9*a^2*b^4*c^2*d*e*h*j*l^2 +
 9*a^2*b^3*c^3*e*f*g^2*h*m + 9*a^2*b^3*c^3*d*f*h^2*j*k - 9*a^2*b^3*c^3*d*e*
h^2*j*l - 9*a^2*b^2*c^4*e^2*f*g*j*k - 9*a^2*b^2*c^4*d*e^2*g*j*m + 63*a^3*b^
2*c^3*d*e*f*j*m^2 - 63*a^2*b^2*c^4*d*e*f^2*j*m - 45*a^2*b^4*c^2*d*e*f*j*m^2
 + 36*a^2*b^2*c^4*d*e*f^2*k*l - 27*a^3*b^2*c^3*e*f*g*h*l^2 + 27*a^2*b^3*c^3
*d*e*f*j^2*m + 27*a^2*b^2*c^4*e^2*f*g*h*l + 9*a^2*b^4*c^2*e*f*g*h*l^2 - 9*a
^2*b^3*c^3*e*f*g*h^2*l + 9*a^2*b^3*c^3*d*f*g*h^2*m + 9*a^2*b^3*c^3*d*e*h*j^
2*k + 9*a^2*b^3*c^3*d*e*g*j^2*l + 18*a^2*b^2*c^4*d*e*g^2*j*k - 9*a^3*b^2*c^
3*d*e*g*h*m^2 - 9*a^2*b^3*c^3*d*e*g*j*k^2 - 9*a^2*b^2*c^4*e*f^2*g*h*k - 9*a
^2*b^2*c^4*d*f^2*g*h*l + 18*a^2*b^2*c^4*d*f*g^2*h*k - 18*a^2*b^2*c^4*d*e*g^
2*h*l - 9*a^2*b^3*c^3*d*f*g*h*k^2 - 9*a^2*b^2*c^4*e*f*g^2*h*j + 36*a^2*b^3*
c^3*d*e*f*h*l^2 - 18*a^2*b^2*c^4*d*e*f*h^2*l - 9*a^2*b^2*c^4*d*f*g*h^2*j -
9*a^2*b^2*c^4*d*e*g*h*j^2 - 27*a^2*b^2*c^4*d*e*f*g*k^2 + 18*a^2*b^2*c^4*d^2
*f*h*k^2 - 9*a^2*b^3*c^3*e*f*g^2*k^2 - 9*a^2*b^2*c^4*e^2*f*h*j^2 - 9*a^2*b^
2*c^4*d*f^2*h^2*k + 45*a^2*b^3*c^3*d*e*f^2*m^2 + 36*a^2*b^2*c^4*d^2*e*g*l^2
```

+ 9*a^2*b^3*c^3*d*e*g^2*l^2 + 9*a^2*b^2*c^4*e*f^2*g*j^2 + 9*a^2*b^2*c^4*d*
f^2*h*j^2 - 9*a^2*b^2*c^4*d*e^2*h*k^2 - 36*a^2*b^2*c^4*d*e^2*f*l^2 - 9*a^2*
b^2*c^4*d*f*g^2*j^2 - 12*a^6*b*c*h*k*l^3*m + 3*a*b^6*c*e^3*k*l*m + 3*a*b^6*
c*d*e*f*l^3 - 12*a*b*c^6*d*e^3*f*h + 9*a^5*b^2*c*h^2*k*l^2*m + 18*a^5*b*c^2
*g^2*k^2*l*m - 9*a^5*b^2*c*h^2*j*l*m^2 + 9*a^5*b*c^2*h^2*j^2*l*m - 9*a^4*b^
3*c*g^2*k^2*l*m - 3*a^4*b^2*c^2*g^3*k*l*m + 18*a^5*b*c^2*f^2*k*l*m^2 + 15*a
^3*b^3*c^2*f^3*k*l*m + 9*a^5*b^2*c*h*j^2*k*m^2 + 9*a^5*b^2*c*g*j^2*l*m^2 -
9*a^5*b^2*c*f*k^2*l^2*m + 9*a^5*b*c^2*h^2*j*k^2*m + 9*a^5*b*c^2*g^2*j*l^2*m
- 9*a^4*b^3*c*f^2*k*l*m^2 + 36*a^3*b^2*c^3*e^3*k*l*m - 27*a^5*b*c^2*g^2*j*
k*m^2 - 18*a^5*b*c^2*h^2*j*k*l^2 - 18*a^2*b^4*c^2*e^3*k*l*m - 9*a^5*b^2*c*g
*j*k^2*m^2 - 9*a^5*b^2*c*e*k^2*l*m^2 + 9*a^5*b*c^2*h*j^2*k^2*l + 9*a^5*b*c^
2*g*j^2*k^2*m + 9*a^4*b^3*c*g^2*j*k*m^2 + 9*a^3*b^4*c*e^2*k*l^2*m + 3*a^4*b
^2*c^2*h^3*j*k*l - 54*a^4*b*c^3*d^2*k^2*l*m - 51*a^2*b^3*c^3*d^3*k*l*m - 27
*a^4*b*c^3*e^2*j^2*l*m - 18*a^5*b*c^2*g*h^2*l^2*m - 9*a^5*b^2*c*e*j*l^2*m^2
- 9*a^5*b^2*c*d*k*l^2*m^2 + 9*a^5*b*c^2*g^2*h*l*m^2 + 9*a^5*b*c^2*g*j^2*k*
l^2 + 9*a^5*b*c^2*e*j^2*l^2*m - 9*a^3*b^4*c*e^2*j*l*m^2 - 9*a^2*b^5*c*d^2*k
^2*l*m + 3*a^4*b^2*c^2*g*h^3*l*m - 3*a^3*b^3*c^2*g^3*j*k*l + 18*a^5*b*c^2*e
*j^2*k*m^2 + 18*a^5*b*c^2*d*j^2*l*m^2 + 18*a^4*b*c^3*f^2*j^2*k*l + 9*a^5*b*
c^2*g*h^2*k*m^2 + 9*a^5*b*c^2*f*h^2*l*m^2 + 9*a^5*b*c^2*f*j*k^2*l^2 - 9*a^4
*b^3*c*e*j^2*k*m^2 - 9*a^4*b^3*c*d*j^2*l*m^2 + 9*a^4*b^2*c^2*f*j^3*k*l + 9*
a^4*b^2*c^2*e*j^3*k*m + 9*a^4*b^2*c^2*d*j^3*l*m + 9*a^4*b*c^3*f^2*h^2*l*m +
9*a^4*b*c^3*e^2*j*k^2*m + 9*a^4*b*c^3*d^2*j*l^2*m - 3*a^3*b^3*c^2*g^3*h*k*
m - 3*a^3*b^2*c^3*f^3*j*k*l + 3*a^2*b^4*c^2*f^3*j*k*l + 45*a^4*b*c^3*d^2*j*
k*m^2 - 27*a^5*b*c^2*d*j*k^2*m^2 + 18*a^5*b*c^2*g*h*j^2*m^2 + 18*a^4*b*c^3*
e^2*j*k*l^2 + 15*a^2*b^3*c^3*e^3*j*k*l - 12*a^3*b^2*c^3*f^3*h*k*m - 12*a^3*
b^2*c^3*f^3*g*l*m + 9*a^5*b*c^2*g*h*k^2*l^2 - 9*a^4*b^3*c*g*h*j^2*m^2 + 9*a
^4*b^3*c*d*j*k^2*m^2 + 9*a^4*b^2*c^2*g*h*j^3*m + 9*a^4*b*c^3*g^2*h^2*k*l +
9*a^4*b*c^3*g^2*h^2*j*m + 9*a^2*b^5*c*d^2*j*k*m^2 + 3*a^2*b^4*c^2*f^3*h*k*m
+ 3*a^2*b^4*c^2*f^3*g*l*m + 36*a^2*b^2*c^4*d^3*j*k*l + 18*a^4*b*c^3*e^2*g*
l^2*m + 15*a^2*b^3*c^3*e^3*g*l*m + 12*a^4*b^2*c^2*d*j*k^3*l + 9*a^5*b*c^2*f
*g*k^2*m^2 + 9*a^5*b*c^2*e*h*k^2*m^2 + 9*a^4*b*c^3*g^2*h*j^2*l + 9*a^4*b*c^
3*f^2*h*k^2*l + 9*a^4*b*c^3*f^2*g*k^2*m + 9*a^4*b*c^3*d^2*h*l*m^2 - 9*a^3*b
^3*c^2*e*h^3*k*m + 6*a^2*b^3*c^3*e^3*h*k*m + 45*a^4*b*c^3*e^2*h*j*m^2 + 36*
a^2*b^2*c^4*d^3*h*k*m - 33*a^3*b^2*c^3*d*g^3*l*m - 27*a^4*b*c^3*f^2*h*j*l^2
- 27*a^4*b*c^3*e^2*f*l*m^2 - 27*a^4*b*c^3*e*h^2*j^2*m - 18*a^4*b*c^3*g^2*h
*j*k^2 - 18*a^4*b*c^3*f*g^2*k^2*l - 18*a^4*b*c^3*e*g^2*k^2*m - 18*a^3*b*c^4
*d^2*g^2*l*m + 12*a^4*b^2*c^2*d*h*k^3*m + 9*a^5*b*c^2*e*f*l^2*m^2 + 9*a^5*b
*c^2*d*g*l^2*m^2 + 9*a^4*b*c^3*f^2*g*k*l^2 + 9*a^4*b*c^3*e^2*g*k*m^2 + 9*a^
4*b*c^3*g*h^2*j^2*k + 9*a^4*b*c^3*f*h^2*j^2*l + 9*a^4*b*c^3*e*f^2*l^2*m - 9
*a^3*b^4*c*e*h^2*j*m^2 + 9*a^3*b*c^4*e^2*f^2*l*m + 9*a^2*b^5*c*e^2*h*j*m^2
+ 9*a^2*b^4*c^2*d*g^3*l*m - 9*a^2*b^2*c^4*d^3*g*l*m - 9*a*b^5*c^2*d^2*g^2*l
*m - 6*a^4*b^2*c^2*e*h*k^3*l - 6*a^3*b^2*c^3*f*g^3*j*m + 3*a^4*b^2*c^2*g*h*
j*k^3 + 3*a^4*b^2*c^2*f*g*k^3*l + 3*a^4*b^2*c^2*e*g*k^3*m + 3*a^3*b^2*c^3*g
^3*h*j*k + 3*a^3*b^2*c^3*f*g^3*k*l + 3*a^3*b^2*c^3*e*g^3*k*m - 27*a^3*b*c^4
*d^2*h^2*k*l + 18*a^4*b*c^3*e*f^2*k*m^2 + 18*a^4*b*c^3*d*f^2*l*m^2 + 9*a^4*

b*c^3*f*h^2*j*k^2 + 9*a^4*b*c^3*f*g^2*j*l^2 + 9*a^4*b*c^3*e*g^2*k*l^2 + 9*a
^4*b*c^3*d*h^2*k^2*l + 9*a^3*b^4*c*e*g*j^2*m^2 + 9*a^3*b^4*c*d*h*j^2*m^2 -
9*a^3*b^3*c^2*e*g*j^3*m - 9*a^3*b^3*c^2*d*h*j^3*m + 9*a^3*b*c^4*e^2*g^2*k*l
 + 9*a^3*b*c^4*e^2*g^2*j*m + 9*a^3*b*c^4*d^2*h^2*j*m - 3*a^2*b^3*c^3*f^3*h*
j*k - 3*a^2*b^3*c^3*f^3*g*j*l - 3*a^2*b^3*c^3*e*f^3*k*m - 3*a^2*b^3*c^3*d*f
^3*l*m + 45*a^4*b*c^3*d*g^2*j*m^2 + 45*a^3*b*c^4*d^2*g*j^2*m + 24*a^4*b^2*c
^2*d*g*k*l^3 + 24*a^2*b^2*c^4*e^3*f*j*m + 18*a^4*b*c^3*f^2*g*h*m^2 + 18*a^4
*b*c^3*d*h^2*j*l^2 + 18*a^3*b*c^4*e^2*h^2*j*k - 12*a^4*b^2*c^2*e*g*j*l^3 -
12*a^4*b^2*c^2*e*f*k*l^3 - 12*a^4*b^2*c^2*d*e*l^3*m - 12*a^2*b^2*c^4*e^3*g*
j*l - 12*a^2*b^2*c^4*e^3*f*k*l - 12*a^2*b^2*c^4*d*e^3*l*m + 9*a^4*b*c^3*f*g
*j^2*k^2 + 9*a^4*b*c^3*e*h*j^2*k^2 + 9*a^3*b^2*c^3*e*h^3*j*k + 9*a^3*b^2*c^
3*d*h^3*j*l + 9*a^3*b*c^4*f^2*g^2*j*k + 9*a^3*b*c^4*d^2*h*j^2*l + 9*a^2*b^5
*c*d*g^2*j*m^2 + 9*a*b^5*c^2*d^2*g*j^2*m - 3*a^4*b^2*c^2*d*h*j*l^3 - 3*a^2*
b^3*c^3*f^3*g*h*m - 3*a^2*b^2*c^4*e^3*h*j*k + 18*a^4*b*c^3*f*g*h^2*l^2 + 18
*a^3*b*c^4*e^2*g*h^2*m + 18*a^3*b*c^4*d^2*h*j*k^2 + 18*a^3*b*c^4*d^2*f*k^2*
l + 18*a^3*b*c^4*d^2*e*k^2*m + 9*a^4*b*c^3*e*g^2*h*m^2 + 9*a^4*b*c^3*e*f*j^
2*l^2 + 9*a^4*b*c^3*d*g*j^2*l^2 + 9*a^3*b^2*c^3*f*g*h^3*l + 9*a^3*b^2*c^3*e
*g*h^3*m + 9*a^3*b*c^4*f^2*g^2*h*l + 9*a^3*b*c^4*e^2*g*j^2*k + 9*a^3*b*c^4*
e^2*f*j^2*l - 9*a^2*b^3*c^3*d*g^3*j*l + 9*a*b^4*c^3*d^2*g^2*j*l - 3*a^4*b^2
*c^2*f*g*h*l^3 - 3*a^3*b^3*c^2*e*g*j*k^3 - 3*a^3*b^3*c^2*d*h*j*k^3 - 3*a^3*
b^3*c^2*d*f*k^3*l - 3*a^3*b^3*c^2*d*e*k^3*m - 3*a^2*b^2*c^4*e^3*g*h*m - 33*
a^3*b^2*c^3*d*e*j^3*m - 27*a^4*b*c^3*e*f*h^2*m^2 - 27*a^3*b*c^4*d^2*e*k*l^2
 - 18*a^4*b*c^3*d*e*j^2*m^2 - 18*a^3*b*c^4*e*f^2*j^2*k - 18*a^3*b*c^4*d*f^2
*j^2*l - 9*a^4*b^2*c^2*d*e*j*m^3 + 9*a^4*b*c^3*d*g*h^2*m^2 + 9*a^4*b*c^3*d*
e*k^2*l^2 + 9*a^3*b*c^4*f^2*g*h^2*k + 9*a^3*b*c^4*e^2*f*j*k^2 + 9*a^3*b*c^4
*d^2*f*j*l^2 + 9*a^3*b*c^4*e*f^2*h^2*m + 9*a^3*b*c^4*d*e^2*k^2*l - 9*a^2*b^
5*c*d*e*j^2*m^2 + 9*a^2*b^4*c^2*d*e*j^3*m - 9*a^2*b^3*c^3*d*g^3*h*m + 9*a^2
*b*c^5*d^2*e^2*k*l + 9*a^2*b*c^5*d^2*e^2*j*m + 9*a*b^4*c^3*d^2*g^2*h*m - 6*
a^3*b^2*c^3*d*g*j^3*k - 3*a^3*b^3*c^2*f*g*h*k^3 + 3*a^3*b^2*c^3*e*f*j^3*k +
 3*a^3*b^2*c^3*d*f*j^3*l + 3*a^2*b^2*c^4*e*f^3*j*k + 3*a^2*b^2*c^4*d*f^3*j*
l + 45*a^3*b*c^4*d^2*g*h*l^2 + 36*a^4*b^2*c^2*e*f*g*m^3 + 36*a^4*b^2*c^2*d*
f*h*m^3 - 27*a^3*b*c^4*e^2*g*h*k^2 - 27*a^3*b*c^4*d*g^2*h^2*l - 18*a^3*b*c^
4*f^2*g*h*j^2 + 18*a^3*b*c^4*d*e^2*j*l^2 + 15*a^3*b^3*c^2*d*e*j*l^3 + 12*a^
2*b^2*c^4*e*f^3*g*m + 12*a^2*b^2*c^4*d*f^3*h*m + 9*a^3*b*c^4*f*g^2*h^2*j +
9*a^3*b*c^4*e*g^2*h^2*k + 9*a^3*b*c^4*d*f^2*j*k^2 + 9*a^2*b*c^5*d^2*f^2*j*k
 + 9*a*b^5*c^2*d^2*g*h*l^2 - 9*a*b^4*c^3*d^2*g*h^2*l - 6*a^2*b^2*c^4*e*f^3*
h*l + 3*a^3*b^2*c^3*f*g*h*j^3 + 3*a^2*b^2*c^4*f^3*g*h*j + 45*a^3*b*c^4*d^2*
f*g*m^2 - 27*a^2*b*c^5*d^2*f^2*g*m + 18*a^3*b*c^4*e^2*f*g*l^2 + 15*a^3*b^3*
c^2*e*f*g*l^3 - 12*a^3*b^2*c^3*d*e*j*k^3 + 9*a^3*b*c^4*d^2*e*h*m^2 + 9*a^3*
b*c^4*e*g^2*h*j^2 + 9*a^3*b*c^4*e*f^2*h*k^2 - 9*a^2*b^3*c^3*d*f*h^3*l + 9*a
^2*b*c^5*d^2*f^2*h*l + 9*a*b^5*c^2*d^2*f*g*m^2 + 9*a*b^3*c^4*d^2*f^2*g*m +
6*a^3*b^3*c^2*d*f*h*l^3 + 3*a^2*b^4*c^2*d*e*j*k^3 + 18*a^3*b*c^4*e*f*g^2*k^
2 + 18*a^2*b*c^5*d^2*g^2*h*j + 18*a^2*b*c^5*d^2*f*g^2*l + 18*a^2*b*c^5*d^2*
e*g^2*m - 12*a^3*b^2*c^3*d*f*h*k^3 + 9*a^3*b*c^4*e*f*h^2*j^2 + 9*a^3*b*c^4*
d*f^2*g*l^2 + 9*a^3*b*c^4*d*e^2*g*m^2 + 9*a^3*b*c^4*d*g*h^2*j^2 + 9*a^2*b^2

```
*c^4*e*f*g^3*k + 9*a^2*b^2*c^4*d*g^3*h*j + 9*a^2*b^2*c^4*d*f*g^3*l + 9*a^2*
b^2*c^4*d*e*g^3*m + 9*a^2*b*c^5*e^2*f^2*h*j + 9*a^2*b*c^5*e^2*f^2*g*k - 9*a
*b^3*c^4*d^2*g^2*h*j - 9*a*b^3*c^4*d^2*f*g^2*l - 9*a*b^3*c^4*d^2*e*g^2*m -
3*a^3*b^2*c^3*e*f*g*k^3 + 3*a^2*b^4*c^2*e*f*g*k^3 + 3*a^2*b^4*c^2*d*f*h*k^3
 - 54*a^3*b*c^4*d*e*f^2*m^2 - 51*a^3*b^3*c^2*d*e*f*m^3 - 27*a^3*b*c^4*d*e*g
^2*l^2 + 9*a^3*b*c^4*d*e*h^2*k^2 + 9*a^2*b*c^5*e^2*f*g^2*j + 9*a^2*b*c^5*d^
2*f*h^2*j + 9*a^2*b*c^5*d^2*e*h^2*k + 9*a^2*b*c^5*d*e^2*g^2*l - 9*a*b^5*c^2
*d*e*f^2*m^2 - 9*a*b^4*c^3*d^2*e*g*l^2 - 9*a*b^2*c^5*d^2*e^2*g*l - 9*a*b^2*
c^5*d^2*e^2*f*m - 3*a^2*b^3*c^3*e*f*g*j^3 - 3*a^2*b^3*c^3*d*f*h*j^3 + 36*a^
3*b^2*c^3*d*e*f*l^3 - 27*a^2*b*c^5*d^2*f*g*j^2 - 18*a^2*b^4*c^2*d*e*f*l^3 -
 18*a^2*b*c^5*d*e^2*h^2*j + 9*a^2*b*c^5*d^2*e*h*j^2 + 9*a^2*b*c^5*d*f^2*g^2
*j + 9*a*b^4*c^3*d*e^2*f*l^2 + 9*a*b^3*c^4*d^2*f*g*j^2 - 9*a*b^2*c^5*d^2*f^
2*g*j - 9*a*b^2*c^5*d^2*e*f^2*l + 3*a^2*b^2*c^4*d*e*h^3*j - 18*a^2*b*c^5*e^
2*f*g*h^2 + 18*a^2*b*c^5*d^2*e*f*k^2 + 15*a^2*b^3*c^3*d*e*f*k^3 + 9*a^2*b*c
^5*e*f^2*g^2*h + 9*a^2*b*c^5*d*e^2*g*j^2 - 9*a*b^3*c^4*d^2*e*f*k^2 + 9*a*b^
2*c^5*d^2*e*g^2*j - 9*a*b^2*c^5*d*e^2*f^2*k + 3*a^2*b^2*c^4*e*f*g*h^3 + 18*
a^2*b*c^5*d*e*f^2*j^2 + 9*a^2*b*c^5*d*f^2*g*h^2 - 9*a*b^3*c^4*d*e*f^2*j^2 +
 9*a*b^2*c^5*d^2*f*g^2*h - 3*a^2*b^2*c^4*d*e*f*j^3 + 9*a^2*b*c^5*d*e*g^2*h^
2 - 9*a*b^2*c^5*d^2*e*g*h^2 + 9*a*b^2*c^5*d*e^2*f*h^2 - 36*a^6*c^2*f*j*k*l*
m^2 + 36*a^5*c^3*f^2*j*k*l*m - 36*a^5*c^3*f*h^2*j*l*m + 36*a^5*c^3*e*h*j^2*
l*m - 18*a^6*b*c*j^2*k*l*m^2 + 9*a^6*b*c*j*k^2*l^2*m + 3*a^5*b^2*c*j^3*k*l*
m - 36*a^5*c^3*f*g*j*k^2*m - 36*a^5*c^3*e*f*k^2*l*m + 36*a^5*c^3*d*g*k^2*l*
m - 36*a^4*c^4*d^2*g*k*l*m - 36*a^5*c^3*e*h*j*k*l^2 - 36*a^5*c^3*e*f*j*l^2*
m - 36*a^5*c^3*d*f*k*l^2*m + 36*a^4*c^4*e^2*h*j*k*l + 36*a^4*c^4*e^2*f*j*l*
m + 9*a^6*b*c*h*k^2*l*m^2 - 3*a^4*b^3*c*h^3*k*l*m - 36*a^5*c^3*e*g*h*l^2*m
+ 36*a^5*c^3*e*f*j*k*m^2 - 36*a^5*c^3*d*g*j*k*m^2 + 36*a^5*c^3*d*f*j*l*m^2
- 36*a^5*c^3*d*e*k*l*m^2 + 36*a^4*c^4*e^2*g*h*l*m - 36*a^4*c^4*e*f^2*j*k*m
- 36*a^4*c^4*d*f^2*j*l*m + 9*a^6*b*c*h*j*l^2*m^2 + 9*a^6*b*c*g*k*l^2*m^2 +
9*a^5*b^2*c*g*k^3*l*m + 3*a^3*b^4*c*g^3*k*l*m + 36*a^5*c^3*f*g*h*j*m^2 + 36
*a^5*c^3*e*f*h*l*m^2 - 36*a^4*c^4*f^2*g*h*j*m - 36*a^4*c^4*e*f^2*h*l*m - 24
*a^4*b*c^3*f^3*k*l*m - 12*a^5*b*c^2*h*j^3*k*m - 12*a^5*b*c^2*g*j^3*l*m - 3*
a^2*b^5*c*f^3*k*l*m - 36*a^4*c^4*e*g^2*h*k*l - 36*a^4*c^4*e*f*g^2*l*m + 12*
a^5*b^2*c*e*k*l^3*m - 6*a^5*b^2*c*f*j*l^3*m + 3*a^5*b^2*c*h*j*k*l^3 + 48*a^
3*b*c^4*d^3*k*l*m + 36*a^4*c^4*e*f*h^2*j*m + 36*a^4*c^4*d*g*h^2*k*l - 36*a^
4*c^4*d*f*h^2*k*m - 36*a^4*c^4*d*e*j^2*k*l + 24*a^5*b*c^2*d*k^3*l*m + 21*a*
b^5*c^2*d^3*k*l*m - 12*a^5*b*c^2*g*j*k^3*l - 9*a^4*b^3*c*d*k^3*l*m + 6*a^5*
b*c^2*f*j*k^3*m + 3*a^5*b^2*c*g*h*l^3*m - 36*a^4*c^4*e*f*h*j^2*l - 12*a^5*b
*c^2*g*h*k^3*m - 3*a^5*b^2*c*e*j*k*m^3 - 3*a^5*b^2*c*d*j*l*m^3 - 36*a^4*c^4
*d*g*h*j*k^2 - 36*a^4*c^4*d*f*g*k^2*l - 36*a^4*c^4*d*e*h*k^2*l - 36*a^4*c^4
*d*e*g*k^2*m + 36*a^3*c^5*d^2*g*h*j*k + 36*a^3*c^5*d^2*f*g*k*l - 36*a^3*c^5
*d^2*f*g*j*m + 36*a^3*c^5*d^2*e*h*k*l + 36*a^3*c^5*d^2*e*g*k*m - 36*a^3*c^5
*d^2*e*f*l*m + 24*a^5*b^2*c*e*h*l*m^3 - 24*a^3*b*c^4*e^3*j*k*l - 12*a^5*b^2
*c*f*h*k*m^3 - 12*a^5*b^2*c*f*g*l*m^3 - 3*a^5*b^2*c*g*h*j*m^3 - 3*a^4*b^3*c
*e*j*k*l^3 - 3*a*b^5*c^2*e^3*j*k*l + 36*a^4*c^4*d*e*h*j*l^2 + 36*a^4*c^4*d*
e*g*k*l^2 - 36*a^3*c^5*d*e^2*h*j*l - 36*a^3*c^5*d*e^2*g*k*l - 36*a^3*c^5*d*
```

e^2*f*k*m + 24*a^4*b*c^3*e*h^3*k*m - 24*a^3*b*c^4*e^3*g*l*m - 18*a*b^4*c^3*
d^3*j*k*l - 12*a^4*b*c^3*g*h^3*j*l - 12*a^4*b*c^3*f*h^3*k*l - 12*a^4*b*c^3*
d*h^3*l*m + 12*a^3*b*c^4*e^3*h*k*m + 6*a^4*b*c^3*f*h^3*j*m - 3*a^4*b^3*c*g*
h*j*l^3 - 3*a^4*b^3*c*f*h*k*l^3 - 3*a^4*b^3*c*e*g*l^3*m - 3*a^4*b^3*c*d*h*l
^3*m - 3*a*b^5*c^2*e^3*h*k*m - 3*a*b^5*c^2*e^3*g*l*m + 36*a^4*c^4*e*f*g*h*l
^2 - 36*a^4*c^4*d*e*f*j*m^2 - 36*a^3*c^5*e^2*f*g*h*l - 36*a^3*c^5*d*f^2*g*j
*k - 36*a^3*c^5*d*e*f^2*k*l + 36*a^3*c^5*d*e*f^2*j*m - 18*a*b^4*c^3*d^3*h*k
*m - 9*a*b^4*c^3*d^3*g*l*m + 30*a^5*b*c^2*d*g*k*m^3 - 30*a^4*b^3*c*d*g*k*m^
3 - 24*a^5*b*c^2*e*f*k*m^3 - 24*a^5*b*c^2*d*f*l*m^3 + 24*a^4*b*c^3*e*g*j^3*
m + 24*a^4*b*c^3*d*h*j^3*m + 15*a^4*b^3*c*e*f*k*m^3 + 15*a^4*b^3*c*d*f*l*m^
3 + 12*a^5*b*c^2*e*g*j*m^3 + 12*a^5*b*c^2*d*h*j*m^3 - 12*a^4*b*c^3*f*h*j^3*
k - 12*a^4*b*c^3*f*g*j^3*l + 6*a^4*b^3*c*e*g*j*m^3 + 6*a^4*b^3*c*d*h*j*m^3
+ 6*a^4*b*c^3*e*h*j^3*l + 36*a^3*c^5*d*e*g^2*h*l - 24*a^5*b*c^2*f*g*h*m^3 +
15*a^4*b^3*c*f*g*h*m^3 - 9*a*b^6*c*d^2*g*j*m^2 - 6*a^3*b^4*c*d*g*k*l^3 - 6
*a*b^4*c^3*e^3*f*j*m + 3*a^3*b^4*c*e*g*j*l^3 + 3*a^3*b^4*c*e*f*k*l^3 + 3*a^
3*b^4*c*d*h*j*l^3 + 3*a^3*b^4*c*d*e*l^3*m + 3*a*b^4*c^3*e^3*h*j*k + 3*a*b^4
*c^3*e^3*g*j*l + 3*a*b^4*c^3*e^3*f*k*l + 3*a*b^4*c^3*d*e^3*l*m - 36*a^3*c^5
*d*e*g*h^2*k + 30*a^2*b*c^5*d^3*f*j*m - 30*a*b^3*c^4*d^3*f*j*m + 24*a^3*b*c
^4*d*g^3*j*l - 24*a^2*b*c^5*d^3*h*j*k - 24*a^2*b*c^5*d^3*f*k*l - 24*a^2*b*c
^5*d^3*e*k*m + 15*a*b^3*c^4*d^3*h*j*k + 15*a*b^3*c^4*d^3*f*k*l + 15*a*b^3*c
^4*d^3*e*k*m - 12*a^3*b*c^4*e*g^3*j*k + 12*a^2*b*c^5*d^3*g*j*l + 6*a*b^3*c^
4*d^3*g*j*l + 3*a^3*b^4*c*f*g*h*l^3 + 3*a*b^4*c^3*e^3*g*h*m + 24*a^3*b*c^4*
d*g^3*h*m - 12*a^3*b*c^4*f*g^3*h*k + 12*a^2*b*c^5*d^3*g*h*m - 9*a^3*b^4*c*d
*e*j*m^3 + 6*a^3*b*c^4*e*g^3*h*l + 6*a*b^3*c^4*d^3*g*h*m + 36*a^3*c^5*d*e*f
*g*k^2 - 36*a^2*c^6*d^2*e*f*g*k - 24*a^4*b*c^3*d*e*j*l^3 - 18*a^3*b^4*c*e*f
*g*m^3 - 18*a^3*b^4*c*d*f*h*m^3 - 3*a^2*b^5*c*d*e*j*l^3 - 3*a*b^3*c^4*d*e^3
*j*l - 24*a^4*b*c^3*e*f*g*l^3 + 24*a^3*b*c^4*d*f*h^3*l + 12*a^4*b*c^3*d*f*h
*l^3 - 12*a^3*b*c^4*e*g*h^3*j - 12*a^3*b*c^4*e*f*h^3*k - 12*a^3*b*c^4*d*e*h
^3*m - 12*a*b^2*c^5*d^3*e*j*k + 6*a^3*b*c^4*d*g*h^3*k - 3*a^2*b^5*c*e*f*g*l
^3 - 3*a^2*b^5*c*d*f*h*l^3 - 3*a*b^3*c^4*e^3*g*h*j - 3*a*b^3*c^4*e^3*f*h*k
- 3*a*b^3*c^4*e^3*f*g*l - 3*a*b^3*c^4*d*e^3*h*m + 24*a*b^2*c^5*d^3*e*h*l -
12*a*b^2*c^5*d^3*f*h*k - 3*a*b^2*c^5*d^3*g*h*j - 3*a*b^2*c^5*d^3*f*g*l - 3*
a*b^2*c^5*d^3*e*g*m + 48*a^4*b*c^3*d*e*f*m^3 + 24*a^2*b*c^5*d*e*f^3*m + 21*
a^2*b^5*c*d*e*f*m^3 - 12*a^2*b*c^5*e*f^3*g*j - 12*a^2*b*c^5*d*f^3*h*j - 9*a
*b^3*c^4*d*e*f^3*m + 6*a^2*b*c^5*d*f^3*g*k + 12*a*b^2*c^5*d*e^3*f*l - 6*a*b
^2*c^5*d*e^3*g*k + 3*a*b^2*c^5*d*e^3*h*j - 24*a^3*b*c^4*d*e*f*k^3 - 12*a^2*
b*c^5*d*e*g^3*j - 3*a*b^5*c^2*d*e*f*k^3 + 3*a*b^2*c^5*e^3*f*g*h - 12*a^2*b*
c^5*d*f*g^3*h + 9*a*b^2*c^5*d*e*f^3*j + 9*a*b*c^6*d^2*e^2*f*j + 3*a*b^4*c^3
*d*e*f*j^3 + 9*a*b*c^6*d^2*e^2*g*h + 9*a*b*c^6*d^2*e*f^2*h - 3*a*b^3*c^4*d*
e*f*h^3 - 18*a*b*c^6*d^2*e*f*g^2 + 9*a*b*c^6*d*e^2*f^2*g + 3*a*b^2*c^5*d*e*
f*g^3 - 36*a^4*b^2*c^2*e^2*k*l^2*m - 9*a^4*b^2*c^2*g^2*j^2*k*m + 45*a^3*b^3
*c^2*d^2*k^2*l*m + 36*a^4*b^2*c^2*e^2*j*l*m^2 + 9*a^4*b^2*c^2*g^2*j*k^2*l +
9*a^3*b^3*c^2*e^2*j^2*l*m + 9*a^4*b^2*c^2*g^2*h*k^2*m - 9*a^4*b^2*c^2*f^2*
h*l^2*m - 9*a^3*b^3*c^2*f^2*j^2*k*l - 45*a^3*b^3*c^2*d^2*j*k*m^2 + 36*a^3*b
^2*c^3*d^2*j^2*k*m + 18*a^4*b^2*c^2*f^2*h*k*m^2 + 18*a^4*b^2*c^2*f^2*g*l*m^

2 - 9*a^4*b^2*c^2*g^2*h*k*l^2 - 9*a^4*b^2*c^2*f*h^2*k^2*m - 9*a^4*b^2*c^2*f
*g^2*l^2*m - 9*a^4*b^2*c^2*e*j^2*k^2*l - 9*a^4*b^2*c^2*d*j^2*k^2*m - 9*a^3*
b^3*c^2*e^2*j*k*l^2 - 9*a^2*b^4*c^2*d^2*j^2*k*m - 36*a^3*b^2*c^3*d^2*j*k^2*
l - 27*a^3*b^2*c^3*e^2*h^2*k*m + 9*a^4*b^2*c^2*g*h^2*j*l^2 + 9*a^4*b^2*c^2*
f*h^2*k*l^2 - 9*a^4*b^2*c^2*f*g^2*k*m^2 - 9*a^4*b^2*c^2*e*g^2*l*m^2 - 9*a^4
*b^2*c^2*d*j^2*k*l^2 + 9*a^4*b^2*c^2*d*h^2*l^2*m - 9*a^3*b^3*c^2*e^2*g*l^2*
m + 9*a^2*b^4*c^2*e^2*h^2*k*m + 9*a^2*b^4*c^2*d^2*j*k^2*l - 45*a^3*b^3*c^2*
e^2*h*j*m^2 + 36*a^4*b^2*c^2*e*h^2*j*m^2 + 36*a^3*b^2*c^3*e^2*h*j^2*m - 36*
a^3*b^2*c^3*d^2*h*k^2*m + 36*a^2*b^3*c^3*d^2*g^2*l*m - 9*a^4*b^2*c^2*f*h*j^
2*l^2 - 9*a^4*b^2*c^2*d*h^2*k*m^2 + 9*a^3*b^3*c^2*f^2*h*j*l^2 + 9*a^3*b^3*c
^2*e^2*f*l*m^2 + 9*a^3*b^3*c^2*e*h^2*j^2*m - 9*a^3*b^2*c^3*f^2*h^2*j*l - 9*
a^2*b^4*c^2*e^2*h*j^2*m + 9*a^2*b^4*c^2*d^2*h*k^2*m + 36*a^3*b^2*c^3*d^2*h*
k*l^2 - 27*a^4*b^2*c^2*e*g*j^2*m^2 - 27*a^4*b^2*c^2*d*h*j^2*m^2 - 9*a^4*b^2
*c^2*d*h*k^2*l^2 - 9*a^3*b^3*c^2*e*f^2*k*m^2 - 9*a^3*b^3*c^2*d*f^2*l*m^2 +
9*a^3*b^2*c^3*f^2*h*j^2*k + 9*a^3*b^2*c^3*f^2*g*j^2*l - 9*a^3*b^2*c^3*e^2*g
*k^2*l - 9*a^3*b^2*c^3*e^2*f*k^2*m - 9*a^3*b^2*c^3*d^2*f*l^2*m - 9*a^2*b^4*
c^2*d^2*h*k*l^2 + 9*a^2*b^3*c^3*d^2*h^2*k*l - 81*a^3*b^2*c^3*d^2*g*j*m^2 +
54*a^2*b^4*c^2*d^2*g*j*m^2 - 45*a^3*b^3*c^2*d*g^2*j*m^2 - 45*a^2*b^3*c^3*d^
2*g*j^2*m + 36*a^3*b^2*c^3*d^2*f*k*m^2 + 36*a^3*b^2*c^3*d*g^2*j^2*m + 18*a^
3*b^2*c^3*e^2*g*j*l^2 + 18*a^3*b^2*c^3*e^2*f*k*l^2 + 18*a^3*b^2*c^3*d*e^2*l
^2*m - 9*a^4*b^2*c^2*d*f*k^2*m^2 - 9*a^3*b^3*c^2*f^2*g*h*m^2 - 9*a^3*b^3*c^
2*d*h^2*j*l^2 - 9*a^3*b^2*c^3*f^2*g*j*k^2 - 9*a^3*b^2*c^3*d^2*e*l*m^2 - 9*a
^3*b^2*c^3*f*g^2*h^2*m - 9*a^3*b^2*c^3*e*g^2*j^2*l - 9*a^3*b^2*c^3*e*f^2*k^
2*l - 9*a^2*b^4*c^2*d^2*f*k*m^2 - 9*a^2*b^4*c^2*d*g^2*j^2*m - 9*a^2*b^3*c^3
*e^2*h^2*j*k - 9*a^2*b^2*c^4*d^2*f^2*k*m - 27*a^2*b^2*c^4*d^2*g^2*j*l - 9*a
^3*b^3*c^2*f*g*h^2*l^2 + 9*a^3*b^2*c^3*e*g^2*j*k^2 - 9*a^3*b^2*c^3*e*f^2*j*
l^2 - 9*a^3*b^2*c^3*d*h^2*j^2*k - 9*a^3*b^2*c^3*d*f^2*k*l^2 - 9*a^3*b^2*c^3
*d*e^2*k*m^2 - 9*a^2*b^3*c^3*e^2*g*h^2*m - 9*a^2*b^3*c^3*d^2*h*j*k^2 - 9*a^
2*b^3*c^3*d^2*f*k^2*l - 9*a^2*b^3*c^3*d^2*e*k^2*m + 36*a^3*b^3*c^2*d*e*j^2*
m^2 + 36*a^3*b^2*c^3*e^2*f*h*m^2 - 27*a^2*b^2*c^4*d^2*g^2*h*m + 9*a^3*b^3*c
^2*e*f*h^2*m^2 + 9*a^3*b^2*c^3*f*g^2*h*k^2 - 9*a^2*b^4*c^2*e^2*f*h*m^2 + 9*
a^2*b^3*c^3*d^2*e*k*l^2 - 9*a^2*b^2*c^4*e^2*f^2*h*m - 45*a^2*b^3*c^3*d^2*g*
h*l^2 - 36*a^3*b^2*c^3*e*f^2*g*m^2 + 36*a^3*b^2*c^3*d*g^2*h*l^2 - 36*a^3*b^
2*c^3*d*f^2*h*m^2 + 36*a^2*b^2*c^4*d^2*g*h^2*l - 9*a^3*b^2*c^3*e*g*h^2*k^2
+ 9*a^2*b^4*c^2*e*f^2*g*m^2 - 9*a^2*b^4*c^2*d*g^2*h*l^2 + 9*a^2*b^4*c^2*d*f
^2*h*m^2 + 9*a^2*b^3*c^3*e^2*g*h*k^2 + 9*a^2*b^3*c^3*d*g^2*h^2*l - 9*a^2*b^
3*c^3*d*e^2*j*l^2 - 9*a^2*b^2*c^4*e^2*g^2*h*k - 9*a^2*b^2*c^4*e^2*f*g^2*m -
9*a^2*b^2*c^4*d^2*f*j^2*k - 9*a^2*b^2*c^4*d^2*f*h^2*m - 9*a^2*b^2*c^4*d^2*
e*j^2*l - 45*a^2*b^3*c^3*d^2*f*g*m^2 + 36*a^3*b^2*c^3*d*f*g^2*m^2 - 27*a^3*
b^2*c^3*d*f*h^2*l^2 + 18*a^2*b^2*c^4*d^2*e*j*k^2 + 9*a^2*b^4*c^2*d*f*h^2*l^
2 - 9*a^2*b^4*c^2*d*f*g^2*m^2 - 9*a^2*b^3*c^3*e^2*f*g*l^2 + 9*a^2*b^2*c^4*e
^2*g*h^2*j + 9*a^2*b^2*c^4*e^2*f*h^2*k - 9*a^2*b^2*c^4*e*f^2*g^2*l - 9*a^2*
b^2*c^4*d*f^2*g^2*m - 9*a^2*b^2*c^4*d*e^2*j^2*k + 9*a^2*b^2*c^4*d*e^2*h^2*m
+ 18*a^4*b^2*c^2*f^2*j^2*m^2 + 18*a^3*b^2*c^3*e^2*h^2*l^2 - 9*a^2*b^4*c^2*
e^2*h^2*l^2 + 18*a^2*b^2*c^4*d^2*g^2*k^2 + 12*a^6*c^2*j^3*k*l*m + 3*a^6*b^2

*j*k*l*m^3 - 12*a^6*c^2*g*k^3*l*m - 12*a^5*c^3*g^3*k*l*m - 24*a^6*c^2*e*k*l
^3*m - 24*a^4*c^4*e^3*k*l*m + 12*a^6*c^2*h*j*k*l^3 + 12*a^6*c^2*f*j*l^3*m +
12*a^5*c^3*h^3*j*k*l - 3*a^5*b^3*h*j*k*m^3 - 3*a^5*b^3*g*j*l*m^3 - 3*a^5*b
^3*f*k*l*m^3 + 12*a^6*c^2*g*h*l^3*m + 12*a^5*c^3*g*h^3*l*m - 12*a^6*c^2*e*j
*k*m^3 - 12*a^6*c^2*d*j*l*m^3 - 12*a^5*c^3*f*j^3*k*l - 12*a^5*c^3*e*j^3*k*m
- 12*a^5*c^3*d*j^3*l*m - 12*a^4*c^4*f^3*j*k*l + 24*a^6*c^2*f*h*k*m^3 + 24*
a^6*c^2*f*g*l*m^3 + 24*a^4*c^4*f^3*h*k*m + 24*a^4*c^4*f^3*g*l*m - 12*a^6*c^
2*g*h*j*m^3 - 12*a^6*c^2*e*h*l*m^3 - 12*a^5*c^3*g*h*j^3*m + 3*b^6*c^2*d^3*j
*k*l + 3*a^4*b^4*e*j*k*m^3 + 3*a^4*b^4*d*j*l*m^3 - 24*a^5*c^3*d*j*k^3*l - 2
4*a^3*c^5*d^3*j*k*l - 6*a^4*b^4*e*h*l*m^3 + 3*b^6*c^2*d^3*h*k*m + 3*b^6*c^2
*d^3*g*l*m + 3*a^6*b*c*j^2*l^3*m + 3*a^4*b^4*g*h*j*m^3 + 3*a^4*b^4*f*h*k*m^
3 + 3*a^4*b^4*f*g*l*m^3 - 24*a^5*c^3*d*h*k^3*m - 24*a^3*c^5*d^3*h*k*m + 12*
a^5*c^3*g*h*j*k^3 + 12*a^5*c^3*f*g*k^3*l + 12*a^5*c^3*e*h*k^3*l + 12*a^5*c^
3*e*g*k^3*m + 12*a^4*c^4*g^3*h*j*k + 12*a^4*c^4*f*g^3*k*l + 12*a^4*c^4*f*g^
3*j*m + 12*a^4*c^4*e*g^3*k*m + 12*a^4*c^4*d*g^3*l*m + 12*a^3*c^5*d^3*g*l*m
+ 3*a^6*b*c*j*k^3*m^2 - 9*a^6*b*c*h^2*l*m^3 - 3*a^5*b*c^2*j^4*k*l + 24*a^5*
c^3*e*g*j*l^3 + 24*a^5*c^3*e*f*k*l^3 + 24*a^5*c^3*d*e*l^3*m + 24*a^3*c^5*e^
3*g*j*l + 24*a^3*c^5*e^3*f*k*l + 24*a^3*c^5*d*e^3*l*m - 12*a^5*c^3*d*h*j*l^
3 - 12*a^5*c^3*d*g*k*l^3 - 12*a^4*c^4*e*h^3*j*k - 12*a^4*c^4*d*h^3*j*l - 12
*a^3*c^5*e^3*h*j*k - 12*a^3*c^5*e^3*f*j*m + 9*a^4*b*c^3*g^4*l*m + 6*b^5*c^3
*d^3*f*j*m + 6*a^3*b^5*d*g*k*m^3 - 3*b^5*c^3*d^3*h*j*k - 3*b^5*c^3*d^3*g*j*
l - 3*b^5*c^3*d^3*f*k*l - 3*b^5*c^3*d^3*e*k*m - 3*a^3*b^5*e*g*j*m^3 - 3*a^3
*b^5*e*f*k*m^3 - 3*a^3*b^5*d*h*j*m^3 - 3*a^3*b^5*d*f*l*m^3 - 12*a^5*c^3*f*g
*h*l^3 - 12*a^4*c^4*f*g*h^3*l - 12*a^4*c^4*e*g*h^3*m - 12*a^3*c^5*e^3*g*h*m
- 9*a^6*b*c*g*k^2*m^3 - 3*b^5*c^3*d^3*g*h*m + 3*a^6*b*c*f*l^3*m^2 - 3*a^3*
b^5*f*g*h*m^3 + 12*a^5*c^3*d*e*j*m^3 + 12*a^4*c^4*e*f*j^3*k + 12*a^4*c^4*d*
g*j^3*k + 12*a^4*c^4*d*f*j^3*l + 12*a^4*c^4*d*e*j^3*m + 12*a^3*c^5*e*f^3*j*
k + 12*a^3*c^5*d*f^3*j*l - 9*a^6*b*c*e*l^2*m^3 - 24*a^5*c^3*e*f*g*m^3 - 24*
a^5*c^3*d*f*h*m^3 - 24*a^3*c^5*e*f^3*g*m - 24*a^3*c^5*d*f^3*h*m - 15*a^2*b*
c^5*d^4*l*m + 15*a*b^3*c^4*d^4*l*m + 12*a^4*c^4*f*g*h*j^3 + 12*a^3*c^5*f^3*
g*h*j + 12*a^3*c^5*e*f^3*h*l + 9*a^3*b*c^4*f^4*k*l - 9*a^3*b*c^4*f^4*j*m +
3*b^4*c^4*d^3*e*j*k + 3*a^5*b^2*c*g*j*l^4 + 3*a^5*b^2*c*f*k*l^4 + 3*a^5*b^2
*c*d*l^4*m - 3*a^5*b*c^2*h*j*k^4 - 3*a^5*b*c^2*f*k^4*l - 3*a^5*b*c^2*e*k^4*
m - 3*a^4*b*c^3*h^4*j*k + 3*a^2*b^6*d*e*j*m^3 + 3*a*b^4*c^3*e^4*k*m + 24*a^
4*c^4*d*e*j*k^3 + 24*a^2*c^6*d^3*e*j*k - 6*b^4*c^4*d^3*e*h*l + 3*b^4*c^4*d^
3*g*h*j + 3*b^4*c^4*d^3*f*h*k + 3*b^4*c^4*d^3*f*g*l + 3*b^4*c^4*d^3*e*g*m -
3*a^4*b*c^3*g*h^4*m + 3*a^2*b^6*e*f*g*m^3 + 3*a^2*b^6*d*f*h*m^3 - 3*a*b^6*
c*e^3*j*m^2 + 24*a^4*c^4*d*f*h*k^3 + 24*a^2*c^6*d^3*f*h*k - 12*a^4*c^4*e*f*
g*k^3 - 12*a^3*c^5*e*f*g^3*k - 12*a^3*c^5*d*g^3*h*j - 12*a^3*c^5*d*f*g^3*l
- 12*a^3*c^5*d*e*g^3*m - 12*a^2*c^6*d^3*g*h*j - 12*a^2*c^6*d^3*f*g*l - 12*a
^2*c^6*d^3*e*h*l - 12*a^2*c^6*d^3*e*g*m - 12*a*b^2*c^5*d^4*j*l + 9*a^5*b*c^
2*d*j*l^4 + 9*a^2*b*c^5*e^4*j*k - 3*a^4*b^3*c*d*j*l^4 - 3*a^4*b*c^3*e*j^4*k
- 3*a^4*b*c^3*d*j^4*l - 3*a*b^3*c^4*e^4*j*k - 24*a^4*c^4*d*e*f*l^3 - 24*a^
2*c^6*d*e^3*f*l - 12*a^5*b^2*c*e*g*m^4 - 12*a^5*b^2*c*d*h*m^4 + 12*a^3*c^5*
d*e*h^3*j + 12*a^2*c^6*d*e^3*h*j + 12*a^2*c^6*d*e^3*g*k - 12*a*b^2*c^5*d^4*

h*m + 9*a^5*b*c^2*f*g*l^4 - 9*a^5*b*c^2*e*h*l^4 - 9*a^2*b*c^5*e^4*h*l + 9*a
^2*b*c^5*e^4*g*m + 6*a^4*b^3*c*e*h*l^4 + 6*a*b^3*c^4*e^4*h*l - 3*b^3*c^5*d^
3*e*g*j - 3*b^3*c^5*d^3*e*f*k - 3*a^4*b^3*c*f*g*l^4 - 3*a^4*b*c^3*g*h*j^4 -
 3*a^3*b*c^4*g^4*h*j - 3*a^3*b*c^4*f*g^4*l - 3*a^3*b*c^4*e*g^4*m - 3*a*b^3*
c^4*e^4*g*m + 12*a^3*c^5*e*f*g*h^3 + 12*a^2*c^6*e^3*f*g*h - 3*b^3*c^5*d^3*f
*g*h - 12*a^3*c^5*d*e*f*j^3 - 12*a^2*c^6*d*e*f^3*j - 3*a*b^6*c*d^2*g*l^3 -
15*a^5*b*c^2*d*e*m^4 + 15*a^4*b^3*c*d*e*m^4 + 9*a^4*b*c^3*e*f*k^4 - 9*a^4*b
*c^3*d*g*k^4 + 3*a^3*b^4*c*d*f*l^4 - 3*a^3*b*c^4*d*h^4*j - 3*a^2*b*c^5*e*f^
4*k - 3*a^2*b*c^5*d*f^4*l + 3*a*b^2*c^5*e^4*g*j + 3*a*b^2*c^5*e^4*f*k + 3*a
*b^2*c^5*d*e^4*m - 9*a*b*c^6*d^3*e^2*l + 3*b^2*c^6*d^3*e*f*g - 3*a^3*b*c^4*
f*g*h^4 - 3*a^2*b*c^5*f^4*g*h + 12*a^2*c^6*d*e*f*g^3 - 9*a*b*c^6*d^3*f^2*j
+ 3*a*b*c^6*d^2*e^3*k + 9*a^3*b*c^4*d*e*j^4 - 3*a^2*b*c^5*e*f*g^4 - 9*a*b*c
^6*d^3*e*h^2 + 3*a*b*c^6*d^2*f^3*g + 3*a*b*c^6*d*e^3*g^2 - 3*a^4*b^2*c^2*h^
3*j^2*m + 12*a^4*b^2*c^2*g^3*j*m^2 - 3*a^4*b^2*c^2*f^2*k^3*m + 3*a^3*b^3*c^
2*g^3*j^2*m - 9*a^3*b^4*c*f^2*j^2*m^2 + 9*a^3*b^3*c^2*f^2*j^3*m - 6*a^3*b^3
*c^2*f^3*j*m^2 - 6*a^3*b^2*c^3*f^3*j^2*m - 3*a^2*b^4*c^2*f^3*j^2*m - 27*a^4
*b^2*c^2*d^2*k*m^3 - 27*a^3*b^2*c^3*e^3*j*m^2 + 18*a^2*b^4*c^2*e^3*j*m^2 -
15*a^2*b^3*c^3*e^3*j^2*m + 12*a^4*b^2*c^2*f^2*j*l^3 + 3*a^3*b^3*c^2*e^2*k^3
*l + 42*a^2*b^3*c^3*d^3*j*m^2 - 27*a^2*b^2*c^4*d^3*j^2*m - 15*a^3*b^3*c^2*d
^2*k*l^3 - 3*a^4*b^2*c^2*f*j^2*k^3 - 3*a^4*b^2*c^2*f*h^3*m^2 + 3*a^3*b^3*c^
2*g^3*h*l^2 + 3*a^3*b^3*c^2*f^2*j*k^3 - 3*a^3*b^2*c^3*g^3*h^2*l - 3*a^3*b^2
*c^3*e^2*j^3*l - 27*a^4*b^2*c^2*e^2*h*m^3 + 12*a^3*b^2*c^3*f^3*h*l^2 + 3*a^
3*b^3*c^2*f*g^3*m^2 - 3*a^2*b^4*c^2*f^3*h*l^2 + 3*a^2*b^3*c^3*f^3*h^2*l + 9
*a^3*b^3*c^2*e*h^3*l^2 + 9*a^2*b^3*c^3*e^2*h^3*l - 6*a^4*b^2*c^2*e*h^2*l^3
- 6*a^3*b^3*c^2*e^2*h*l^3 - 6*a^2*b^3*c^3*e^3*h*l^2 - 6*a^2*b^2*c^4*e^3*h^2
*l + 3*a^2*b^3*c^3*d^2*j^3*k + 42*a^3*b^3*c^2*d^2*g*m^3 - 27*a^4*b^2*c^2*d*
g^2*m^3 - 27*a^2*b^2*c^4*d^3*h*l^2 - 15*a^2*b^3*c^3*e^3*f*m^2 + 12*a^3*b^2*
c^3*e^2*h*k^3 + 3*a^3*b^3*c^2*e*h^2*k^3 - 3*a^3*b^2*c^3*e*g^3*l^2 - 3*a^2*b
^4*c^2*e^2*h*k^3 + 3*a^2*b^3*c^3*f^3*g*k^2 - 3*a^2*b^2*c^4*f^3*g^2*k - 27*a
^3*b^2*c^3*d^2*g*l^3 - 27*a^2*b^2*c^4*d^3*f*m^2 + 18*a^2*b^4*c^2*d^2*g*l^3
- 15*a^3*b^3*c^2*d*g^2*l^3 + 12*a^2*b^2*c^4*e^3*g*k^2 - 3*a^3*b^2*c^3*e*h^2
*j^3 + 3*a^2*b^3*c^3*e^2*h*j^3 + 3*a^2*b^3*c^3*e*f^3*l^2 - 3*a^2*b^2*c^4*d^
2*h^3*k + 9*a^2*b^3*c^3*d*g^3*k^2 - 9*a*b^4*c^3*d^2*g^2*k^2 - 6*a^3*b^2*c^3
*d*g^2*k^3 - 6*a^2*b^3*c^3*d^2*g*k^3 - 3*a^2*b^4*c^2*d*g^2*k^3 + 12*a^2*b^2
*c^4*d^2*g*j^3 + 3*a^2*b^3*c^3*d*g^2*j^3 - 3*a^2*b^2*c^4*d*f^3*k^2 - 3*a^2*
b^2*c^4*d*g^2*h^3 + 12*a^7*c*j*k*l*m^3 - 3*b^7*c*d^3*k*l*m - 3*a^6*b*c*k^4*
l*m - 3*a^6*b*c*j*k*l^4 - 3*a^6*b*c*g*l^4*m - 9*a^6*b*c*f*j*m^4 + 9*a^6*b*c
*e*k*m^4 + 9*a^6*b*c*d*l*m^4 + 9*a^6*b*c*g*h*m^4 - 3*a*b^7*d*e*f*m^3 + 9*a*
b*c^6*d^4*h*j - 9*a*b*c^6*d^4*g*k + 9*a*b*c^6*d^4*f*l + 9*a*b*c^6*d^4*e*m +
 12*a*c^7*d^3*e*f*g - 3*a*b*c^6*d*e^4*j - 3*a*b*c^6*e^4*f*g - 3*a*b*c^6*d*e
*f^4 + 18*a^6*c^2*h^2*j*l*m^2 - 18*a^6*c^2*h*j^2*l^2*m + 18*a^6*c^2*f*k^2*l
^2*m + 36*a^5*c^3*e^2*k*l^2*m + 18*a^6*c^2*g*j*k^2*m^2 + 18*a^6*c^2*e*k^2*l
*m^2 + 18*a^5*c^3*g^2*j^2*k*m + 18*a^6*c^2*e*j*l^2*m^2 + 18*a^6*c^2*d*k*l^2
*m^2 - 18*a^5*c^3*e^2*j*l*m^2 - 18*a^6*c^2*f*h*l^2*m^2 + 18*a^5*c^3*f^2*h*l
^2*m - 36*a^5*c^3*f^2*h*k*m^2 - 36*a^5*c^3*f^2*g*l*m^2 + 18*a^5*c^3*g^2*h*k

*l^2 - 18*a^5*c^3*g*h^2*k^2*l + 18*a^5*c^3*f*h^2*k^2*m + 18*a^5*c^3*f*g^2*l
^2*m + 18*a^5*c^3*e*j^2*k^2*l + 18*a^5*c^3*d*j^2*k^2*m - 18*a^4*c^4*d^2*j^2
*k*m + 36*a^4*c^4*d^2*j*k^2*l + 18*a^5*c^3*f*g^2*k*m^2 + 18*a^5*c^3*e*g^2*l
*m^2 + 18*a^5*c^3*d*j^2*k*l^2 - 18*a^4*c^4*f^2*g^2*k*m + 36*a^4*c^4*d^2*h*k
^2*m + 18*a^5*c^3*f*h*j^2*l^2 - 18*a^5*c^3*e*h^2*j*m^2 + 18*a^5*c^3*d*h^2*k
*m^2 + 18*a^4*c^4*f^2*h^2*j*l - 18*a^4*c^4*e^2*h*j^2*m - 18*a^5*c^3*e*g*k^2
*l^2 + 18*a^5*c^3*d*h*k^2*l^2 + 18*a^4*c^4*e^2*g*k^2*l + 18*a^4*c^4*e^2*f*k
^2*m - 18*a^4*c^4*d^2*h*k*l^2 + 18*a^4*c^4*d^2*f*l^2*m - 36*a^4*c^4*e^2*g*j
*l^2 - 36*a^4*c^4*e^2*f*k*l^2 - 36*a^4*c^4*d*e^2*l^2*m + 18*a^5*c^3*d*f*k^2
*m^2 + 18*a^4*c^4*f^2*g*j*k^2 + 18*a^4*c^4*d^2*g*j*m^2 - 18*a^4*c^4*d^2*f*k
*m^2 + 18*a^4*c^4*d^2*e*l*m^2 - 18*a^4*c^4*f*g^2*j^2*k + 18*a^4*c^4*f*g^2*h
^2*m + 18*a^4*c^4*e*g^2*j^2*l + 18*a^4*c^4*e*f^2*k^2*l - 18*a^4*c^4*d*g^2*j
^2*m - 18*a^4*c^4*d*f^2*k^2*m + 18*a^3*c^5*d^2*f^2*k*m + 3*a^4*b^2*c^2*h^4*
k*m - 3*a^3*b^3*c^2*g^4*l*m + 18*a^4*c^4*e*f^2*j*l^2 + 18*a^4*c^4*d*h^2*j^2
*k + 18*a^4*c^4*d*f^2*k*l^2 + 18*a^4*c^4*d*e^2*k*m^2 - 18*a^3*c^5*e^2*f^2*j
*l + 12*a^5*b^2*c*g^2*k*m^3 - 9*a^5*b*c^2*h^3*j*m^2 - 9*a^5*b*c^2*f^2*l^3*m
 + 3*a^5*b*c^2*h^2*k^3*l + 3*a^4*b^3*c*h^3*j*m^2 + 3*a^4*b^3*c*f^2*l^3*m -
18*a^4*c^4*e^2*f*h*m^2 + 18*a^3*c^5*e^2*f^2*h*m + 15*a^5*b*c^2*e^2*l*m^3 -
15*a^4*b^3*c*e^2*l*m^3 - 9*a^5*b*c^2*g^2*k*l^3 - 9*a^4*b*c^3*g^3*j^2*m - 3*
a^5*b^2*c*g*k^2*l^3 + 3*a^5*b*c^2*h*j^3*l^2 + 3*a^4*b^3*c*g^2*k*l^3 - 3*a^3
*b^4*c*g^3*j*m^2 + 36*a^4*c^4*e*f^2*g*m^2 + 36*a^4*c^4*d*f^2*h*m^2 + 18*a^4
*c^4*e*g*h^2*k^2 - 18*a^4*c^4*d*g^2*h*l^2 - 18*a^4*c^4*d*f*j^2*k^2 + 18*a^3
*c^5*e^2*g^2*h*k + 18*a^3*c^5*e^2*f*g^2*m - 18*a^3*c^5*d^2*g*h^2*l + 18*a^3
*c^5*d^2*f*j^2*k + 18*a^3*c^5*d^2*f*h^2*m + 18*a^3*c^5*d^2*e*j^2*l - 12*a^2
*b^2*c^4*e^4*k*m + 9*a^4*b^3*c*f*j^3*m^2 - 9*a^4*b^2*c^2*f*j^4*m - 6*a^5*b^
2*c*f*j^2*m^3 + 6*a^5*b*c^2*f^2*j*m^3 - 6*a^5*b*c^2*f*j^3*m^2 - 6*a^4*b^3*c
*f^2*j*m^3 + 6*a^4*b*c^3*f^3*j*m^2 - 6*a^4*b*c^3*f^2*j^3*m + 6*a^2*b^3*c^3*
f^4*j*m + 3*a^3*b^2*c^3*g^4*j*l + 3*a^2*b^5*c*f^3*j*m^2 - 3*a^2*b^3*c^3*f^4
*k*l - 36*a^3*c^5*d^2*e*j*k^2 - 18*a^4*c^4*d*f*g^2*m^2 + 18*a^3*c^5*e*f^2*g
^2*l + 18*a^3*c^5*d*f^2*g^2*m + 18*a^3*c^5*d*e^2*j^2*k + 18*a^3*b^4*c*d^2*k
*m^3 + 15*a^3*b*c^4*e^3*j^2*m + 12*a^5*b^2*c*d*k^2*m^3 - 9*a^5*b*c^2*f*j^2*
l^3 - 9*a^4*b*c^3*e^2*k^3*l + 3*a^5*b*c^2*e*k^3*l^2 + 3*a^4*b^3*c*f*j^2*l^3
 + 3*a^4*b*c^3*g^2*j^3*k - 3*a^3*b^4*c*f^2*j*l^3 + 3*a^3*b^2*c^3*g^4*h*m +
3*a*b^5*c^2*e^3*j^2*m - 36*a^3*c^5*d^2*f*h*k^2 - 21*a^3*b*c^4*d^3*j*m^2 - 2
1*a*b^5*c^2*d^3*j*m^2 + 18*a^3*c^5*e^2*f*h*j^2 - 18*a^3*c^5*e*f^2*h^2*j + 1
8*a^3*c^5*d*f^2*h^2*k + 18*a*b^4*c^3*d^3*j^2*m + 15*a^4*b*c^3*d^2*k*l^3 - 9
*a^5*b*c^2*d*k^2*l^3 - 9*a^4*b*c^3*g^3*h*l^2 - 9*a^4*b*c^3*f^2*j*k^3 + 3*a^
4*b^3*c*d*k^2*l^3 + 3*a^2*b^5*c*d^2*k*l^3 - 18*a^3*c^5*d^2*e*g*l^2 + 18*a^3
*c^5*d*e^2*h*k^2 + 18*a^3*b^4*c*e^2*h*m^3 - 18*a^2*c^6*d^2*e^2*h*k + 18*a^2
*c^6*d^2*e^2*g*l + 18*a^2*c^6*d^2*e^2*f*m + 15*a^5*b*c^2*e*h^2*m^3 - 15*a^4
*b^3*c*e*h^2*m^3 - 9*a^4*b*c^3*f*g^3*m^2 - 9*a^3*b*c^4*f^3*h^2*l + 3*a^4*b^
2*c^2*e*j*k^4 + 3*a^4*b*c^3*g*h^3*k^2 + 3*a^3*b*c^4*f^2*g^3*m + 36*a^3*c^5*
d*e^2*f*l^2 + 18*a^3*c^5*d*f*g^2*j^2 + 18*a^2*c^6*d^2*f^2*g*j + 18*a^2*c^6*
d^2*e*f^2*l - 9*a^3*b^2*c^3*e*h^4*l - 9*a^3*b*c^4*d^2*j^3*k + 6*a^4*b*c^3*e
^2*h*l^3 - 6*a^4*b*c^3*e*h^3*l^2 + 6*a^3*b*c^4*e^3*h*l^2 - 6*a^3*b*c^4*e^2*

h^3*l + 3*a^4*b^2*c^2*f*h*k^4 + 3*a^4*b*c^3*d*j^3*k^2 - 3*a^3*b^4*c*e*h^2*l
^3 + 3*a^2*b^5*c*e^2*h*l^3 + 3*a^2*b^2*c^4*f^4*h*k + 3*a^2*b^2*c^4*f^4*g*l
+ 3*a*b^5*c^2*e^3*h*l^2 - 3*a*b^4*c^3*e^3*h^2*l - 21*a^4*b*c^3*d^2*g*m^3 -
21*a^2*b^5*c*d^2*g*m^3 + 18*a^3*b^4*c*d*g^2*m^3 + 18*a^2*c^6*d*e^2*f^2*k +
18*a*b^4*c^3*d^3*h*l^2 + 15*a^3*b*c^4*e^3*f*m^2 + 15*a^2*b*c^5*d^3*h^2*l -
15*a*b^3*c^4*d^3*h^2*l - 9*a^4*b*c^3*e*h^2*k^3 - 9*a^3*b*c^4*f^3*g*k^2 - 9*
a^2*b*c^5*e^3*f^2*m + 3*a^3*b*c^4*f^2*h^3*j + 3*a*b^5*c^2*e^3*f*m^2 + 3*a*b
^3*c^4*e^3*f^2*m + 18*a*b^4*c^3*d^3*f*m^2 + 15*a^4*b*c^3*d*g^2*l^3 + 12*a*b
^2*c^5*d^3*f^2*m - 9*a^3*b*c^4*e^2*h*j^3 - 9*a^3*b*c^4*e*f^3*l^2 - 9*a^2*b*
c^5*e^3*g^2*k + 3*a^3*b*c^4*f*g^3*j^2 + 3*a^2*b^5*c*d*g^2*l^3 + 3*a^2*b*c^5
*e^2*f^3*l - 3*a*b^4*c^3*e^3*g*k^2 + 3*a*b^3*c^4*e^3*g^2*k + 18*a^2*c^6*d^2
*e*g*h^2 - 18*a^2*c^6*d*e^2*g^2*h - 12*a^4*b^2*c^2*d*f*l^4 - 9*a^2*b^2*c^4*
d*g^4*k + 9*a*b^3*c^4*d^2*g^3*k + 6*a^3*b^3*c^2*d*g*k^4 + 6*a^3*b*c^4*d^2*g
*k^3 - 6*a^3*b*c^4*d*g^3*k^2 + 6*a^2*b*c^5*d^3*g*k^2 - 6*a^2*b*c^5*d^2*g^3*
k - 6*a*b^3*c^4*d^3*g*k^2 - 6*a*b^2*c^5*d^3*g^2*k - 3*a^3*b^3*c^2*e*f*k^4 +
3*a^3*b^2*c^3*e*g*j^4 + 3*a^3*b^2*c^3*d*h*j^4 + 3*a*b^5*c^2*d^2*g*k^3 + 15
*a^2*b*c^5*d^3*e*l^2 - 15*a*b^3*c^4*d^3*e*l^2 - 9*a^3*b*c^4*d*g^2*j^3 - 9*a
^2*b*c^5*e^3*f*j^2 - 3*a*b^4*c^3*d^2*g*j^3 + 3*a*b^3*c^4*e^3*f*j^2 - 3*a*b^
2*c^5*e^3*f^2*j + 12*a*b^2*c^5*d^3*f*j^2 - 9*a^2*b*c^5*d*e^3*k^2 + 3*a^2*b*
c^5*e^2*g^3*h + 3*a*b^3*c^4*d*e^3*k^2 - 9*a^2*b*c^5*d^2*g*h^3 - 3*a^2*b^3*c
^3*d*e*j^4 + 3*a^2*b*c^5*e*f^3*h^2 + 3*a*b^3*c^4*d^2*g*h^3 + 3*a^2*b^2*c^4*
d*f*h^4 - 9*a^7*c*k^2*l^2*m^2 - 6*a^6*c^2*j^2*k^3*m - 3*a^6*b^2*h*l^2*m^3 +
3*a^5*b^3*h^2*l*m^3 - 6*a^6*c^2*g^2*k*m^3 - 6*a^6*c^2*h*k^3*l^2 + 6*a^5*c^
3*h^3*j^2*m + 6*a^6*c^2*g*k^2*l^3 - 6*a^6*c^2*f*k^3*m^2 - 6*a^5*c^3*h^2*j^3
*l - 6*a^5*c^3*g^3*j*m^2 + 6*a^5*c^3*f^2*k^3*m + 3*a^5*b^3*g*k^2*m^3 - 3*a^
4*b^4*g^2*k*m^3 + 12*a^6*c^2*f*j^2*m^3 + 12*a^4*c^4*f^3*j^2*m + 3*a^5*b^3*e
*l^2*m^3 + 3*a^3*b^5*e^2*l*m^3 - 6*a^6*c^2*d*k^2*m^3 - 6*a^5*c^3*f^2*j*l^3
+ 6*a^5*c^3*d^2*k*m^3 - 6*a^5*c^3*g*j^3*k^2 + 6*a^4*c^4*e^3*j*m^2 - 3*b^6*c
^2*d^3*j^2*m - 3*a^4*b^4*f*j^2*m^3 + 3*a^3*b^5*f^2*j*m^3 + 6*a^5*c^3*f*j^2*
k^3 + 6*a^5*c^3*f*h^3*m^2 - 6*a^5*c^3*e*j^3*l^2 + 6*a^4*c^4*g^3*h^2*l - 6*a
^4*c^4*f^2*h^3*m + 6*a^4*c^4*e^2*j^3*l + 6*a^3*c^5*d^3*j^2*m - 3*a^4*b^4*d*
k^2*m^3 - 3*a^2*b^6*d^2*k*m^3 + 6*a^5*c^3*e^2*h*m^3 - 6*a^4*c^4*g^2*h^3*k -
6*a^4*c^4*f^3*h*l^2 + 12*a^5*c^3*e*h^2*l^3 + 12*a^3*c^5*e^3*h^2*l - 3*b^6*
c^2*d^3*h*l^2 + 3*b^5*c^3*d^3*h^2*l - 3*a^5*b^2*c*j^4*m^2 + 3*a^3*b^5*e*h^2
*m^3 - 3*a^2*b^6*e^2*h*m^3 + 6*a^5*c^3*d*g^2*m^3 - 6*a^4*c^4*e^2*h*k^3 - 6*
a^4*c^4*f*h^3*j^2 + 6*a^4*c^4*e*g^3*l^2 + 6*a^3*c^5*f^3*g^2*k - 6*a^3*c^5*e
^2*g^3*l + 6*a^3*c^5*d^3*h*l^2 - 3*b^6*c^2*d^3*f*m^2 - 3*b^4*c^4*d^3*f^2*m
+ 6*a^4*c^4*d^2*g*l^3 + 6*a^4*c^4*e*h^2*j^3 - 6*a^4*c^4*d*h^3*k^2 - 6*a^3*c
^5*f^2*g^3*j - 6*a^3*c^5*e^3*g*k^2 + 6*a^3*c^5*d^3*f*m^2 + 6*a^3*c^5*d^2*h^
3*k - 6*a^2*c^6*d^3*f^2*m + 4*a^5*b^2*c*h^3*m^3 + 3*b^5*c^3*d^3*g*k^2 - 3*b
^4*c^4*d^3*g^2*k - 3*a^2*b^6*d*g^2*m^3 + a^5*b*c^2*j^3*k^3 + 12*a^4*c^4*d*g
^2*k^3 + 12*a^2*c^6*d^3*g^2*k + 6*a^5*b*c^2*h^3*l^3 + 5*a^5*b*c^2*g^3*m^3 -
5*a^4*b^3*c*g^3*m^3 + 3*b^5*c^3*d^3*e*l^2 + 3*b^3*c^5*d^3*e^2*l - 3*a^5*b^
2*c*h^2*l^4 + a^4*b^3*c*h^3*l^3 + 12*a^5*b^2*c*f^2*m^4 - 6*a^3*c^5*d^2*g*j^
3 + 6*a^3*c^5*d*f^3*k^2 + 6*a^3*b^4*c*f^3*m^3 + 6*a^2*c^6*e^3*f^2*j - 6*a^2

*c^6*d^2*f^3*k - 3*b^4*c^4*d^3*f*j^2 + 3*b^3*c^5*d^3*f^2*j - 3*a^2*b^2*c^4*
f^5*m - 7*a^4*b*c^3*e^3*m^3 - 7*a^2*b^5*c*e^3*m^3 + 6*a^4*b*c^3*g^3*k^3 - 6
*a^3*c^5*e*g^3*h^2 - 6*a^2*c^6*d^3*f*j^2 + 5*a^4*b*c^3*f^3*l^3 + a^4*b*c^3*
h^3*j^3 + a^2*b^5*c*f^3*l^3 + 6*a^3*c^5*d*g^2*h^3 - 6*a^2*c^6*e^2*f^3*h - 3
*a^3*b^4*c*e^2*l^4 - 3*a*b^4*c^3*e^4*l^2 - 7*a^3*b*c^4*d^3*l^3 - 7*a*b^5*c^
2*d^3*l^3 + 6*a^3*b*c^4*f^3*j^3 + 5*a^3*b*c^4*e^3*k^3 + 3*b^3*c^5*d^3*e*h^2
 - 3*b^2*c^6*d^3*e^2*h + a*b^5*c^2*e^3*k^3 + 12*a*b^2*c^5*d^4*k^2 - 6*a^2*c
^6*d*f^3*g^2 + 6*a*b^4*c^3*d^3*k^3 - 3*a^4*b^2*c^2*d*k^5 + a^3*b*c^4*g^3*h^
3 + 5*a^2*b*c^5*d^3*j^3 - 5*a*b^3*c^4*d^3*j^3 - 9*a*c^7*d^2*e^2*f^2 + 6*a^2
*b*c^5*e^3*h^3 - 3*a*b^2*c^5*e^4*h^2 + a^2*b*c^5*f^3*g^3 + a*b^3*c^4*e^3*h^
3 + 4*a*b^2*c^5*d^3*h^3 - 3*a*b^2*c^5*d^2*g^4 - 6*a^7*c*j*l^3*m^2 + 6*a^7*c
*h*l^2*m^3 + 6*a^6*c^2*j*k^4*l + 6*a^6*c^2*h*k^4*m - 6*a^5*c^3*h^4*k*m + 3*
a^6*b^2*h*k*m^4 + 3*a^6*b^2*g*l*m^4 - 3*b^5*c^3*d^4*l*m - 6*a^6*c^2*g*j*l^4
 - 6*a^6*c^2*f*k*l^4 - 6*a^6*c^2*d*l^4*m + 6*a^5*c^3*h*j^4*k + 6*a^5*c^3*g*
j^4*l + 6*a^5*c^3*f*j^4*m - 6*a^4*c^4*g^4*j*l + 6*a^3*c^5*e^4*k*m + 6*a^5*b
^3*f*j*m^4 - 6*a^4*c^4*g^4*h*m + 3*b^7*c*d^3*j*m^2 - 3*a^5*b^3*e*k*m^4 - 3*
a^5*b^3*d*l*m^4 + 3*b^4*c^4*d^4*j*l - 3*a^5*b^3*g*h*m^4 - 6*a^5*c^3*e*j*k^4
 + 6*a^2*c^6*d^4*j*l + 3*b^4*c^4*d^4*h*m + 6*a^6*c^2*e*g*m^4 + 6*a^6*c^2*d*
h*m^4 + 6*a^6*b*c*j^3*m^3 - 6*a^5*c^3*f*h*k^4 + 6*a^4*c^4*g*h^4*j + 6*a^4*c
^4*f*h^4*k + 6*a^4*c^4*e*h^4*l + 6*a^4*c^4*d*h^4*m - 6*a^3*c^5*f^4*h*k - 6*
a^3*c^5*f^4*g*l + 6*a^2*c^6*d^4*h*m + 3*a^5*b*c^2*j^5*m + a^6*b*c*k^3*l^3 +
 3*a^4*b^4*e*g*m^4 + 3*a^4*b^4*d*h*m^4 + 6*b^3*c^5*d^4*g*k - 3*b^3*c^5*d^4*
h*j - 3*b^3*c^5*d^4*f*l - 3*b^3*c^5*d^4*e*m + 3*a*b^7*d^2*g*m^3 + 6*a^5*c^3
*d*f*l^4 - 6*a^4*c^4*e*g*j^4 - 6*a^4*c^4*d*h*j^4 + 6*a^3*c^5*e*g^4*j + 6*a^
3*c^5*d*g^4*k - 6*a^2*c^6*e^4*g*j - 6*a^2*c^6*e^4*f*k - 6*a^2*c^6*d*e^4*m +
 3*a^4*b*c^3*h^5*l + 6*a^3*c^5*f*g^4*h - 3*a^3*b^5*d*e*m^4 + 3*b^2*c^6*d^4*
e*j + 3*a^5*b*c^2*g*k^5 + 3*a^3*b*c^4*g^5*k + 8*a*b^6*c*d^3*m^3 + 3*b^2*c^6
*d^4*f*h - 3*a^5*b^2*c*e*l^5 - 3*a*b^2*c^5*e^5*l - 6*a^3*c^5*d*f*h^4 + 6*a^
2*c^6*e*f^4*g + 6*a^2*c^6*d*f^4*h + 3*a^4*b*c^3*f*j^5 + 3*a^2*b*c^5*f^5*j +
 6*a*c^7*d^3*e^2*h - 6*a*c^7*d^2*e^3*g + 3*a^3*b*c^4*e*h^5 + 6*a*b*c^6*d^3*
g^3 + 3*a^2*b*c^5*d*g^5 + a*b*c^6*e^3*f^3 - 9*a^6*c^2*j^2*k^2*l^2 - 9*a^6*c
^2*h^2*k^2*m^2 - 9*a^6*c^2*g^2*l^2*m^2 - 18*a^5*c^3*f^2*j^2*m^2 - 9*a^5*c^3
*h^2*j^2*k^2 - 9*a^5*c^3*g^2*j^2*l^2 - 9*a^5*c^3*f^2*k^2*l^2 - 9*a^5*c^3*e^
2*k^2*m^2 - 9*a^5*c^3*d^2*l^2*m^2 - 9*a^5*c^3*g^2*h^2*m^2 - 9*a^4*c^4*e^2*j
^2*k^2 - 9*a^4*c^4*d^2*j^2*l^2 - 18*a^4*c^4*e^2*h^2*l^2 - 9*a^4*c^4*g^2*h^2
*j^2 - 9*a^4*c^4*f^2*h^2*k^2 - 9*a^4*c^4*f^2*g^2*l^2 - 9*a^4*c^4*e^2*g^2*m^
2 - 9*a^4*c^4*d^2*h^2*m^2 - 18*a^3*c^5*d^2*g^2*k^2 - 9*a^3*c^5*e^2*g^2*j^2
 - 9*a^3*c^5*e^2*f^2*k^2 - 9*a^3*c^5*d^2*h^2*j^2 - 9*a^3*c^5*d^2*f^2*l^2 - 9
*a^3*c^5*d^2*e^2*m^2 - 3*a^4*b^2*c^2*h^4*l^2 - 18*a^4*b^2*c^2*f^3*m^3 + 12*
a^3*b^2*c^3*f^4*m^2 - 9*a^3*c^5*f^2*g^2*h^2 + 4*a^4*b^2*c^2*g^3*l^3 - 3*a^2
*b^4*c^2*f^4*m^2 + 14*a^3*b^3*c^2*e^3*m^3 - 5*a^3*b^3*c^2*f^3*l^3 - 3*a^4*b
^2*c^2*g^2*k^4 - 3*a^3*b^2*c^3*g^4*k^2 + a^3*b^3*c^2*g^3*k^3 - 20*a^2*b^4*c
^2*d^3*m^3 - 18*a^3*b^2*c^3*e^3*l^3 + 16*a^3*b^2*c^3*d^3*m^3 + 12*a^4*b^2*c
^2*e^2*l^4 + 12*a^2*b^2*c^4*e^4*l^2 - 9*a^2*c^6*d^2*e^2*j^2 + 6*a^2*b^4*c^2
*e^3*l^3 + 4*a^3*b^2*c^3*f^3*k^3 + 14*a^2*b^3*c^3*d^3*l^3 - 9*a^2*c^6*e^2*f

^2*g^2 - 9*a^2*c^6*d^2*f^2*h^2 - 5*a^2*b^3*c^3*e^3*k^3 - 3*a^3*b^2*c^3*f^2*
j^4 - 3*a^2*b^2*c^4*f^4*j^2 + a^2*b^3*c^3*f^3*j^3 - 18*a^2*b^2*c^4*d^3*k^3
+ 12*a^3*b^2*c^3*d^2*k^4 + 4*a^2*b^2*c^4*e^3*j^3 - 3*a^2*b^4*c^2*d^2*k^4 -
3*a^2*b^2*c^4*e^2*h^4 + 6*a^7*c*k*l^4*m - 3*a^7*b*k*l*m^4 - 6*a^7*c*h*k*m^4
 - 6*a^7*c*g*l*m^4 + 3*a^6*b*c*h*l^5 - 6*a*c^7*d^4*e*j - 6*a*c^7*d^4*f*h -
3*b*c^7*d^4*e*f + 6*a*c^7*d*e^4*f + 3*a*b*c^6*e^5*h - a^5*b^2*c*j^3*l^3 - a
^3*b^4*c*g^3*l^3 - a*b^4*c^3*e^3*j^3 - a*b^2*c^5*e^3*g^3 + 3*a^7*b*j*m^5 +
6*a^7*c*f*m^5 + 6*a*c^7*d^5*k + 3*b*c^7*d^5*g - 3*a^6*c^2*j^4*m^2 - 3*a^6*b
^2*j^2*m^4 + 2*a^6*c^2*j^3*l^3 + a^5*b^3*j^3*m^3 - 2*a^6*c^2*h^3*m^3 - 3*a^
6*c^2*h^2*l^4 - 3*a^5*c^3*h^4*l^2 - a*b^6*c*e^3*l^3 + 20*a^5*c^3*f^3*m^3 -
15*a^6*c^2*f^2*m^4 - 15*a^4*c^4*f^4*m^2 + 2*a^5*c^3*h^3*k^3 - 2*a^5*c^3*g^3
*l^3 + a^3*b^5*g^3*m^3 - 3*a^5*c^3*g^2*k^4 - 3*a^4*c^4*g^4*k^2 - 3*a^4*b^4*
f^2*m^4 + 20*a^4*c^4*e^3*l^3 - 15*a^5*c^3*e^2*l^4 - 15*a^3*c^5*e^4*l^2 + 2*
a^4*c^4*g^3*j^3 - 2*a^4*c^4*f^3*k^3 - 2*a^4*c^4*d^3*m^3 - 3*b^4*c^4*d^4*k^2
 - 3*a^4*c^4*f^2*j^4 - 3*a^3*c^5*f^4*j^2 + 20*a^3*c^5*d^3*k^3 - 15*a^4*c^4*
d^2*k^4 - 15*a^2*c^6*d^4*k^2 - 2*a^3*c^5*e^3*j^3 + b^5*c^3*d^3*j^3 + 2*a^3*
c^5*f^3*h^3 - 3*a^3*c^5*e^2*h^4 - 3*a^2*c^6*e^4*h^2 - 3*b^2*c^6*d^4*g^2 + 2
*a^2*c^6*e^3*g^3 - 2*a^2*c^6*d^3*h^3 + b^3*c^5*d^3*g^3 - 3*a^2*c^6*d^2*g^4
- a^4*b^2*c^2*h^3*k^3 - a^3*b^2*c^3*g^3*j^3 - a^2*b^4*c^2*f^3*k^3 - a^2*b^2
*c^4*f^3*h^3 + 2*a^7*c*k^3*m^3 + a^7*b*l^3*m^3 - 3*a^7*c*j^2*m^4 + 6*a^3*c^
5*f^5*m - 3*a^6*b^2*f*m^5 + 6*a^6*c^2*e*l^5 + 6*a^2*c^6*e^5*l + b^7*c*d^3*l
^3 + a*b^7*e^3*m^3 - 3*b^2*c^6*d^5*k + 6*a^5*c^3*d*k^5 - 3*a*c^7*d^4*g^2 +
2*a*c^7*d^3*f^3 + b*c^7*d^3*e^3 - a^6*b^2*k^3*m^3 - a^4*b^4*h^3*m^3 - a^2*b
^6*f^3*m^3 - b^6*c^2*d^3*k^3 - b^4*c^4*d^3*h^3 - b^2*c^6*d^3*f^3 - b^8*d^3*
m^3 - a^6*c^2*k^6 - a^5*c^3*j^6 - a^4*c^4*h^6 - a^3*c^5*g^6 - a^2*c^6*f^6 -
 a^7*c*l^6 - a*c^7*e^6 - a^8*m^6 - c^8*d^6, z, k1)*(root(34992*a^4*b^2*c^8*
z^6 - 8748*a^3*b^4*c^7*z^6 + 729*a^2*b^6*c^6*z^6 - 46656*a^5*c^9*z^6 + 3499
2*a^4*b^3*c^6*m*z^5 - 8748*a^3*b^5*c^5*m*z^5 + 729*a^2*b^7*c^4*m*z^5 - 3499
2*a^4*b^2*c^7*j*z^5 + 8748*a^3*b^4*c^6*j*z^5 - 729*a^2*b^6*c^5*j*z^5 - 4665
6*a^5*b*c^7*m*z^5 + 46656*a^5*c^8*j*z^5 + 34992*a^5*b*c^6*j*m*z^4 - 11664*a
^5*b*c^6*k*l*z^4 + 3888*a^4*b*c^7*f*j*z^4 + 3888*a^4*b*c^7*e*k*z^4 + 3888*a
^4*b*c^7*d*l*z^4 + 3888*a^4*b*c^7*g*h*z^4 + 3888*a^3*b*c^8*d*e*z^4 + 243*a*
b^5*c^6*d*e*z^4 - 25272*a^4*b^3*c^5*j*m*z^4 + 9720*a^4*b^3*c^5*k*l*z^4 + 60
75*a^3*b^5*c^4*j*m*z^4 - 2673*a^3*b^5*c^4*k*l*z^4 - 486*a^2*b^7*c^3*j*m*z^4
 + 243*a^2*b^7*c^3*k*l*z^4 - 7776*a^4*b^2*c^6*h*k*z^4 - 7776*a^4*b^2*c^6*g*
l*z^4 - 7776*a^4*b^2*c^6*f*m*z^4 + 2430*a^3*b^4*c^5*h*k*z^4 + 2430*a^3*b^4*
c^5*g*l*z^4 + 2430*a^3*b^4*c^5*f*m*z^4 - 243*a^2*b^6*c^4*h*k*z^4 - 243*a^2*
b^6*c^4*g*l*z^4 - 243*a^2*b^6*c^4*f*m*z^4 - 1944*a^3*b^3*c^6*f*j*z^4 - 1944
*a^3*b^3*c^6*e*k*z^4 - 1944*a^3*b^3*c^6*d*l*z^4 + 243*a^2*b^5*c^5*f*j*z^4 +
 243*a^2*b^5*c^5*e*k*z^4 + 243*a^2*b^5*c^5*d*l*z^4 - 1944*a^3*b^3*c^6*g*h*z
^4 + 243*a^2*b^5*c^5*g*h*z^4 + 3888*a^3*b^2*c^7*e*g*z^4 + 3888*a^3*b^2*c^7*
d*h*z^4 - 486*a^2*b^4*c^6*e*g*z^4 - 486*a^2*b^4*c^6*d*h*z^4 - 1944*a^2*b^3*
c^7*d*e*z^4 + 7776*a^5*c^7*h*k*z^4 + 7776*a^5*c^7*g*l*z^4 + 7776*a^5*c^7*f*
m*z^4 - 7776*a^4*c^8*e*g*z^4 - 7776*a^4*c^8*d*h*z^4 - 13608*a^5*b^2*c^5*m^2
*z^4 + 11421*a^4*b^4*c^4*m^2*z^4 - 2916*a^3*b^6*c^3*m^2*z^4 + 243*a^2*b^8*c

^2*m^2*z^4 + 13608*a^4*b^2*c^6*j^2*z^4 - 3159*a^3*b^4*c^5*j^2*z^4 + 243*a^2
*b^6*c^4*j^2*z^4 + 1944*a^3*b^2*c^7*f^2*z^4 - 243*a^2*b^4*c^6*f^2*z^4 - 388
8*a^6*c^6*m^2*z^4 - 19440*a^5*c^7*j^2*z^4 - 3888*a^4*c^8*f^2*z^4 + 3078*a^4
*b^4*c^3*k*l*m*z^3 - 2592*a^5*b^2*c^4*k*l*m*z^3 - 891*a^3*b^6*c^2*k*l*m*z^3
 - 4536*a^4*b^3*c^4*j*k*l*z^3 + 1053*a^3*b^5*c^3*j*k*l*z^3 - 81*a^2*b^7*c^2
*j*k*l*z^3 - 2592*a^4*b^3*c^4*h*k*m*z^3 - 2592*a^4*b^3*c^4*g*l*m*z^3 + 810*
a^3*b^5*c^3*h*k*m*z^3 + 810*a^3*b^5*c^3*g*l*m*z^3 - 81*a^2*b^7*c^2*h*k*m*z^
3 - 81*a^2*b^7*c^2*g*l*m*z^3 + 7776*a^4*b^2*c^5*f*j*m*z^3 + 3888*a^4*b^2*c^
5*h*j*k*z^3 + 3888*a^4*b^2*c^5*g*j*l*z^3 - 3888*a^4*b^2*c^5*f*k*l*z^3 - 291
6*a^3*b^4*c^4*f*j*m*z^3 + 1458*a^3*b^4*c^4*f*k*l*z^3 - 972*a^3*b^4*c^4*h*j*
k*z^3 - 972*a^3*b^4*c^4*g*j*l*z^3 - 486*a^3*b^4*c^4*e*k*m*z^3 - 486*a^3*b^4
*c^4*d*l*m*z^3 + 324*a^2*b^6*c^3*f*j*m*z^3 - 162*a^2*b^6*c^3*f*k*l*z^3 + 81
*a^2*b^6*c^3*h*j*k*z^3 + 81*a^2*b^6*c^3*g*j*l*z^3 + 81*a^2*b^6*c^3*e*k*m*z^
3 + 81*a^2*b^6*c^3*d*l*m*z^3 - 486*a^3*b^4*c^4*g*h*m*z^3 + 81*a^2*b^6*c^3*g
*h*m*z^3 + 648*a^3*b^3*c^5*e*j*k*z^3 + 648*a^3*b^3*c^5*d*j*l*z^3 - 81*a^2*b
^5*c^4*e*j*k*z^3 - 81*a^2*b^5*c^4*d*j*l*z^3 + 2592*a^3*b^3*c^5*e*g*m*z^3 +
2592*a^3*b^3*c^5*d*h*m*z^3 - 1296*a^3*b^3*c^5*f*h*k*z^3 - 1296*a^3*b^3*c^5*
f*g*l*z^3 - 1296*a^3*b^3*c^5*e*h*l*z^3 + 648*a^3*b^3*c^5*g*h*j*z^3 - 324*a^
2*b^5*c^4*e*g*m*z^3 - 324*a^2*b^5*c^4*d*h*m*z^3 + 162*a^2*b^5*c^4*f*h*k*z^3
 + 162*a^2*b^5*c^4*f*g*l*z^3 + 162*a^2*b^5*c^4*e*h*l*z^3 - 81*a^2*b^5*c^4*g
*h*j*z^3 + 5184*a^3*b^2*c^6*d*e*m*z^3 - 2592*a^3*b^2*c^6*e*g*j*z^3 - 2592*a
^3*b^2*c^6*d*h*j*z^3 - 2106*a^2*b^4*c^5*d*e*m*z^3 + 1296*a^3*b^2*c^6*e*f*k*
z^3 + 1296*a^3*b^2*c^6*d*g*k*z^3 + 1296*a^3*b^2*c^6*d*f*l*z^3 + 324*a^2*b^4
*c^5*e*g*j*z^3 + 324*a^2*b^4*c^5*d*h*j*z^3 - 162*a^2*b^4*c^5*e*f*k*z^3 - 16
2*a^2*b^4*c^5*d*g*k*z^3 - 162*a^2*b^4*c^5*d*f*l*z^3 + 1296*a^3*b^2*c^6*f*g*
h*z^3 - 162*a^2*b^4*c^5*f*g*h*z^3 + 1944*a^2*b^3*c^6*d*e*j*z^3 - 1296*a^2*b
^2*c^7*d*e*f*z^3 + 81*a^2*b^8*c*k*l*m*z^3 + 6480*a^5*b*c^5*j*k*l*z^3 + 2592
*a^5*b*c^5*h*k*m*z^3 + 2592*a^5*b*c^5*g*l*m*z^3 - 1296*a^4*b*c^6*e*j*k*z^3
- 1296*a^4*b*c^6*d*j*l*z^3 - 5184*a^4*b*c^6*e*g*m*z^3 - 5184*a^4*b*c^6*d*h*
m*z^3 + 2592*a^4*b*c^6*f*h*k*z^3 + 2592*a^4*b*c^6*f*g*l*z^3 + 2592*a^4*b*c^
6*e*h*l*z^3 - 1296*a^4*b*c^6*g*h*j*z^3 + 243*a*b^6*c^4*d*e*m*z^3 - 3888*a^3
*b*c^7*d*e*j*z^3 - 243*a*b^5*c^5*d*e*j*z^3 + 162*a*b^4*c^6*d*e*f*z^3 - 2592
*a^6*c^5*k*l*m*z^3 - 5184*a^5*c^6*h*j*k*z^3 - 5184*a^5*c^6*g*j*l*z^3 - 5184
*a^5*c^6*f*j*m*z^3 + 2592*a^5*c^6*f*k*l*z^3 + 2592*a^5*c^6*e*k*m*z^3 + 2592
*a^5*c^6*d*l*m*z^3 + 2592*a^5*c^6*g*h*m*z^3 + 5184*a^4*c^7*e*g*j*z^3 + 5184
*a^4*c^7*d*h*j*z^3 - 2592*a^4*c^7*e*f*k*z^3 - 2592*a^4*c^7*d*g*k*z^3 - 2592
*a^4*c^7*d*f*l*z^3 - 2592*a^4*c^7*d*e*m*z^3 - 2592*a^4*c^7*f*g*h*z^3 + 2592
*a^3*c^8*d*e*f*z^3 + 6480*a^5*b^2*c^4*j*m^2*z^3 + 6480*a^4*b^3*c^4*j^2*m*z^
3 - 5022*a^4*b^4*c^3*j*m^2*z^3 - 1296*a^3*b^5*c^3*j^2*m*z^3 + 1134*a^3*b^6*
c^2*j*m^2*z^3 + 81*a^2*b^7*c^2*j^2*m*z^3 + 2592*a^4*b^3*c^4*h*l^2*z^3 - 194
4*a^4*b^2*c^5*h^2*l*z^3 - 810*a^3*b^5*c^3*h*l^2*z^3 + 729*a^3*b^4*c^4*h^2*l
*z^3 + 81*a^2*b^7*c^2*h*l^2*z^3 - 81*a^2*b^6*c^3*h^2*l*z^3 - 5184*a^4*b^3*c
^4*f*m^2*z^3 + 1620*a^3*b^5*c^3*f*m^2*z^3 + 1296*a^3*b^3*c^5*f^2*m*z^3 - 16
2*a^2*b^7*c^2*f*m^2*z^3 - 162*a^2*b^5*c^4*f^2*m*z^3 - 1944*a^4*b^2*c^5*g*k^
2*z^3 + 729*a^3*b^4*c^4*g*k^2*z^3 - 648*a^3*b^3*c^5*g^2*k*z^3 - 81*a^2*b^6*

c^3*g*k^2*z^3 + 81*a^2*b^5*c^4*g^2*k*z^3 - 1944*a^4*b^2*c^5*e*l^2*z^3 + 729
*a^3*b^4*c^4*e*l^2*z^3 + 648*a^3*b^2*c^6*e^2*l*z^3 - 81*a^2*b^6*c^3*e*l^2*z
^3 - 81*a^2*b^4*c^5*e^2*l*z^3 + 1296*a^3*b^3*c^5*f*j^2*z^3 - 1296*a^3*b^2*c
^6*f^2*j*z^3 - 162*a^2*b^5*c^4*f*j^2*z^3 + 162*a^2*b^4*c^5*f^2*j*z^3 - 648*
a^3*b^3*c^5*d*k^2*z^3 + 81*a^2*b^5*c^4*d*k^2*z^3 + 648*a^3*b^2*c^6*e*h^2*z^
3 - 81*a^2*b^4*c^5*e*h^2*z^3 - 648*a^2*b^2*c^7*d^2*g*z^3 - 10368*a^5*b*c^5*
j^2*m*z^3 - 81*a^2*b^8*c*j*m^2*z^3 - 2592*a^5*b*c^5*h*l^2*z^3 + 5184*a^5*b*
c^5*f*m^2*z^3 - 2592*a^4*b*c^6*f^2*m*z^3 + 1296*a^4*b*c^6*g^2*k*z^3 - 2592*
a^4*b*c^6*f*j^2*z^3 + 1296*a^4*b*c^6*d*k^2*z^3 + 81*a*b^4*c^6*d^2*g*z^3 + 2
592*a^6*c^5*j*m^2*z^3 + 1296*a^5*c^6*h^2*l*z^3 + 1296*a^5*c^6*g*k^2*z^3 + 1
296*a^5*c^6*e*l^2*z^3 - 1296*a^4*c^7*e^2*l*z^3 + 2592*a^4*c^7*f^2*j*z^3 - 2
592*a^6*b*c^4*m^3*z^3 - 324*a^3*b^7*c*m^3*z^3 - 27*a^2*b^8*c*l^3*z^3 - 1296
*a^4*c^7*e*h^2*z^3 - 864*a^5*b*c^5*k^3*z^3 + 1296*a^3*c^8*d^2*g*z^3 + 432*a
^4*b*c^6*h^3*z^3 + 27*a*b^4*c^6*e^3*z^3 - 432*a^2*b*c^8*d^3*z^3 + 216*a*b^3
*c^7*d^3*z^3 + 1134*a^4*b^5*c^2*m^3*z^3 - 432*a^5*b^3*c^3*m^3*z^3 + 1512*a^
5*b^2*c^4*l^3*z^3 - 1107*a^4*b^4*c^3*l^3*z^3 + 297*a^3*b^6*c^2*l^3*z^3 + 86
4*a^4*b^3*c^4*k^3*z^3 - 270*a^3*b^5*c^3*k^3*z^3 + 27*a^2*b^7*c^2*k^3*z^3 -
2592*a^4*b^2*c^5*j^3*z^3 + 486*a^3*b^4*c^4*j^3*z^3 - 27*a^2*b^6*c^3*j^3*z^3
- 216*a^3*b^3*c^5*h^3*z^3 + 27*a^2*b^5*c^4*h^3*z^3 + 216*a^3*b^2*c^6*g^3*z
^3 - 27*a^2*b^4*c^5*g^3*z^3 - 216*a^2*b^2*c^7*e^3*z^3 - 432*a^6*c^5*l^3*z^3
+ 27*a^2*b^9*m^3*z^3 + 4320*a^5*c^6*j^3*z^3 - 432*a^4*c^7*g^3*z^3 + 432*a^
3*c^8*e^3*z^3 - 27*b^5*c^6*d^3*z^3 + 81*a^3*b^6*c*j*k*l*m*z^2 - 1296*a^5*b*
c^4*h*j*k*m*z^2 - 1296*a^5*b*c^4*g*j*l*m*z^2 + 1296*a^5*b*c^4*f*k*l*m*z^2 -
81*a^2*b^7*c*f*k*l*m*z^2 + 2592*a^4*b*c^5*e*g*j*m*z^2 + 2592*a^4*b*c^5*d*h
*j*m*z^2 - 1296*a^4*b*c^5*f*h*j*k*z^2 - 1296*a^4*b*c^5*f*g*j*l*z^2 - 1296*a
^4*b*c^5*e*f*k*m*z^2 - 1296*a^4*b*c^5*d*f*l*m*z^2 - 648*a^4*b*c^5*e*h*j*l*z
^2 - 648*a^4*b*c^5*e*g*k*l*z^2 - 648*a^4*b*c^5*d*h*k*l*z^2 - 648*a^4*b*c^5*
d*g*k*m*z^2 - 1296*a^4*b*c^5*f*g*h*m*z^2 - 162*a*b^6*c^3*d*e*j*m*z^2 + 81*a
*b^6*c^3*d*e*k*l*z^2 + 1296*a^3*b*c^6*d*e*f*m*z^2 - 648*a^3*b*c^6*d*f*g*k*z
^2 - 648*a^3*b*c^6*d*e*h*k*z^2 - 648*a^3*b*c^6*d*e*g*l*z^2 - 81*a*b^5*c^4*d
*e*h*k*z^2 - 81*a*b^5*c^4*d*e*g*l*z^2 + 81*a*b^5*c^4*d*e*f*m*z^2 - 81*a*b^4
*c^5*d*e*f*j*z^2 + 81*a*b^4*c^5*d*e*g*h*z^2 + 648*a^5*b^2*c^3*j*k*l*m*z^2 -
567*a^4*b^4*c^2*j*k*l*m*z^2 - 1944*a^4*b^3*c^3*f*k*l*m*z^2 + 729*a^3*b^5*c
^2*f*k*l*m*z^2 + 648*a^4*b^3*c^3*h*j*k*m*z^2 + 648*a^4*b^3*c^3*g*j*l*m*z^2
- 81*a^3*b^5*c^2*h*j*k*m*z^2 - 81*a^3*b^5*c^2*g*j*l*m*z^2 + 1944*a^4*b^2*c^
4*f*j*k*l*z^2 - 729*a^3*b^4*c^3*f*j*k*l*z^2 + 648*a^4*b^2*c^4*e*j*k*m*z^2 +
648*a^4*b^2*c^4*d*j*l*m*z^2 - 81*a^3*b^4*c^3*e*j*k*m*z^2 - 81*a^3*b^4*c^3*
d*j*l*m*z^2 + 81*a^2*b^6*c^2*f*j*k*l*z^2 + 1296*a^4*b^2*c^4*f*h*k*m*z^2 + 1
296*a^4*b^2*c^4*f*g*l*m*z^2 + 648*a^4*b^2*c^4*g*h*j*m*z^2 - 648*a^3*b^4*c^3
*f*h*k*m*z^2 - 648*a^3*b^4*c^3*f*g*l*m*z^2 - 324*a^4*b^2*c^4*g*h*k*l*z^2 -
324*a^4*b^2*c^4*e*h*l*m*z^2 + 81*a^3*b^4*c^3*g*h*k*l*z^2 - 81*a^3*b^4*c^3*g
*h*j*m*z^2 + 81*a^2*b^6*c^2*f*h*k*m*z^2 + 81*a^2*b^6*c^2*f*g*l*m*z^2 - 1296
*a^3*b^3*c^4*e*g*j*m*z^2 - 1296*a^3*b^3*c^4*d*h*j*m*z^2 + 648*a^3*b^3*c^4*f
*h*j*k*z^2 + 648*a^3*b^3*c^4*f*g*j*l*z^2 + 648*a^3*b^3*c^4*e*f*k*m*z^2 + 64
8*a^3*b^3*c^4*d*f*l*m*z^2 + 486*a^3*b^3*c^4*e*g*k*l*z^2 + 486*a^3*b^3*c^4*d

*h*k*l*z^2 + 162*a^3*b^3*c^4*e*h*j*l*z^2 + 162*a^3*b^3*c^4*d*g*k*m*z^2 + 16
2*a^2*b^5*c^3*e*g*j*m*z^2 + 162*a^2*b^5*c^3*d*h*j*m*z^2 - 81*a^2*b^5*c^3*f*
h*j*k*z^2 - 81*a^2*b^5*c^3*f*g*j*l*z^2 - 81*a^2*b^5*c^3*e*g*k*l*z^2 - 81*a^
2*b^5*c^3*e*f*k*m*z^2 - 81*a^2*b^5*c^3*d*h*k*l*z^2 - 81*a^2*b^5*c^3*d*f*l*m
*z^2 + 648*a^3*b^3*c^4*f*g*h*m*z^2 - 81*a^2*b^5*c^3*f*g*h*m*z^2 - 3240*a^3*
b^2*c^5*d*e*j*m*z^2 + 1620*a^3*b^2*c^5*d*e*k*l*z^2 + 1377*a^2*b^4*c^4*d*e*j
*m*z^2 - 648*a^3*b^2*c^5*e*f*j*k*z^2 - 648*a^3*b^2*c^5*d*f*j*l*z^2 - 648*a^
2*b^4*c^4*d*e*k*l*z^2 - 324*a^3*b^2*c^5*d*g*j*k*z^2 + 81*a^2*b^4*c^4*e*f*j*
k*z^2 + 81*a^2*b^4*c^4*d*f*j*l*z^2 + 972*a^3*b^2*c^5*e*f*h*l*z^2 - 648*a^3*
b^2*c^5*f*g*h*j*z^2 - 324*a^3*b^2*c^5*e*g*h*k*z^2 - 324*a^3*b^2*c^5*d*g*h*l
*z^2 - 162*a^2*b^4*c^4*e*f*h*l*z^2 + 81*a^2*b^4*c^4*f*g*h*j*z^2 + 81*a^2*b^
4*c^4*e*g*h*k*z^2 + 81*a^2*b^4*c^4*d*g*h*l*z^2 - 648*a^2*b^3*c^5*d*e*f*m*z^
2 + 486*a^2*b^3*c^5*d*e*h*k*z^2 + 486*a^2*b^3*c^5*d*e*g*l*z^2 + 162*a^2*b^3
*c^5*d*f*g*k*z^2 + 648*a^2*b^2*c^6*d*e*f*j*z^2 - 324*a^2*b^2*c^6*d*e*g*h*z^
2 - 1296*a^6*b*c^3*k*l*m^2*z^2 - 81*a^4*b^5*c*k*l*m^2*z^2 - 1296*a^5*b*c^4*
j^2*k*l*z^2 - 324*a^5*b*c^4*h^2*l*m*z^2 + 324*a^5*b*c^4*h*k^2*l*z^2 - 324*a
^5*b*c^4*g*k^2*m*z^2 + 972*a^5*b*c^4*h*j*l^2*z^2 + 324*a^5*b*c^4*g*k*l^2*z^
2 - 324*a^5*b*c^4*e*l^2*m*z^2 - 324*a^4*b*c^5*e^2*l*m*z^2 - 1944*a^5*b*c^4*
f*j*m^2*z^2 + 1296*a^5*b*c^4*e*k*m^2*z^2 + 1296*a^5*b*c^4*d*l*m^2*z^2 + 648
*a^4*b*c^5*f^2*j*m*z^2 + 81*a^2*b^7*c*f*j*m^2*z^2 + 1296*a^5*b*c^4*g*h*m^2*
z^2 - 324*a^4*b*c^5*g^2*j*k*z^2 + 324*a^4*b*c^5*g^2*h*l*z^2 + 972*a^4*b*c^5
*f*h^2*l*z^2 + 324*a^4*b*c^5*g*h^2*k*z^2 - 324*a^4*b*c^5*e*h^2*m*z^2 - 324*
a^4*b*c^5*d*j*k^2*z^2 - 324*a^3*b*c^6*d^2*j*k*z^2 + 972*a^4*b*c^5*f*g*k^2*z
^2 + 972*a^3*b*c^6*d^2*g*m*z^2 + 324*a^4*b*c^5*e*h*k^2*z^2 + 324*a^3*b*c^6*
d^2*h*l*z^2 + 81*a*b^5*c^4*d^2*g*m*z^2 + 972*a^4*b*c^5*e*f*l^2*z^2 + 324*a^
4*b*c^5*d*g*l^2*z^2 - 324*a^3*b*c^6*e^2*h*j*z^2 + 324*a^3*b*c^6*e^2*g*k*z^2
 - 324*a^3*b*c^6*e^2*f*l*z^2 - 1296*a^4*b*c^5*d*e*m^2*z^2 + 81*a*b^7*c^2*d*
e*m^2*z^2 - 324*a^3*b*c^6*d*g^2*j*z^2 - 81*a*b^4*c^5*d^2*g*j*z^2 + 81*a*b^4
*c^5*d^2*e*l*z^2 + 324*a^3*b*c^6*e*g^2*h*z^2 + 81*a*b^4*c^5*d*e^2*k*z^2 + 1
296*a^3*b*c^6*d*e*j^2*z^2 - 324*a^3*b*c^6*e*f*h^2*z^2 + 324*a^3*b*c^6*d*g*h
^2*z^2 + 81*a*b^5*c^4*d*e*j^2*z^2 - 324*a^2*b*c^7*d^2*f*g*z^2 + 324*a^2*b*c
^7*d^2*e*h*z^2 + 81*a*b^3*c^6*d^2*f*g*z^2 - 81*a*b^3*c^6*d^2*e*h*z^2 + 324*
a^2*b*c^7*d*e^2*g*z^2 - 81*a*b^3*c^6*d*e^2*g*z^2 + 1296*a^6*c^4*j*k*l*m*z^2
 - 1296*a^5*c^5*f*j*k*l*z^2 - 1296*a^5*c^5*e*j*k*m*z^2 - 1296*a^5*c^5*d*j*l
*m*z^2 - 1296*a^5*c^5*g*h*j*m*z^2 + 1296*a^5*c^5*e*h*l*m*z^2 + 1296*a^4*c^6
*e*f*j*k*z^2 + 1296*a^4*c^6*d*g*j*k*z^2 + 1296*a^4*c^6*d*f*j*l*z^2 - 1296*a
^4*c^6*d*e*k*l*z^2 + 1296*a^4*c^6*d*e*j*m*z^2 + 1296*a^4*c^6*f*g*h*j*z^2 -
1296*a^4*c^6*e*f*h*l*z^2 - 1296*a^3*c^7*d*e*f*j*z^2 + 648*a^5*b^3*c^2*k*l*m
^2*z^2 + 648*a^4*b^3*c^3*j^2*k*l*z^2 + 486*a^5*b^2*c^3*h*l^2*m*z^2 - 81*a^4
*b^4*c^2*h*l^2*m*z^2 + 81*a^4*b^3*c^3*h^2*l*m*z^2 - 81*a^3*b^5*c^2*j^2*k*l*
z^2 - 162*a^4*b^2*c^4*g^2*k*m*z^2 - 81*a^4*b^3*c^3*h*k^2*l*z^2 + 81*a^4*b^3
*c^3*g*k^2*m*z^2 - 567*a^4*b^3*c^3*h*j*l^2*z^2 + 486*a^4*b^2*c^4*h^2*j*l*z^
2 - 81*a^4*b^3*c^3*g*k*l^2*z^2 + 81*a^4*b^3*c^3*e*l^2*m*z^2 + 81*a^3*b^5*c^
2*h*j*l^2*z^2 - 81*a^3*b^4*c^3*h^2*j*l*z^2 + 81*a^3*b^3*c^4*e^2*l*m*z^2 + 2
430*a^4*b^3*c^3*f*j*m^2*z^2 - 2268*a^4*b^2*c^4*f*j^2*m*z^2 - 810*a^3*b^5*c^

2*f*j*m^2*z^2 + 810*a^3*b^4*c^3*f*j^2*m*z^2 - 648*a^4*b^3*c^3*e*k*m^2*z^2 -
648*a^4*b^3*c^3*d*l*m^2*z^2 - 648*a^4*b^2*c^4*h*j^2*k*z^2 - 648*a^4*b^2*c^
4*g*j^2*l*z^2 - 162*a^3*b^3*c^4*f^2*j*m*z^2 + 81*a^3*b^5*c^2*e*k*m^2*z^2 +
81*a^3*b^5*c^2*d*l*m^2*z^2 + 81*a^3*b^4*c^3*h*j^2*k*z^2 + 81*a^3*b^4*c^3*g*
j^2*l*z^2 - 81*a^2*b^6*c^2*f*j^2*m*z^2 - 648*a^4*b^3*c^3*g*h*m^2*z^2 + 486*
a^4*b^2*c^4*g*j*k^2*z^2 - 486*a^4*b^2*c^4*e*k^2*l*z^2 + 486*a^3*b^2*c^5*d^2
*k*m*z^2 - 162*a^4*b^2*c^4*d*k^2*m*z^2 + 81*a^3*b^5*c^2*g*h*m^2*z^2 - 81*a^
3*b^4*c^3*g*j*k^2*z^2 + 81*a^3*b^4*c^3*e*k^2*l*z^2 + 81*a^3*b^3*c^4*g^2*j*k
*z^2 - 81*a^2*b^4*c^4*d^2*k*m*z^2 + 486*a^4*b^2*c^4*e*j*l^2*z^2 - 486*a^4*b
^2*c^4*d*k*l^2*z^2 - 162*a^3*b^2*c^5*e^2*j*l*z^2 - 81*a^3*b^4*c^3*e*j*l^2*z
^2 + 81*a^3*b^4*c^3*d*k*l^2*z^2 - 81*a^3*b^3*c^4*g^2*h*l*z^2 - 1458*a^4*b^2
*c^4*f*h*l^2*z^2 + 648*a^3*b^4*c^3*f*h*l^2*z^2 - 567*a^3*b^3*c^4*f*h^2*l*z^
2 + 486*a^3*b^2*c^5*e^2*h*m*z^2 - 81*a^3*b^3*c^4*g*h^2*k*z^2 + 81*a^3*b^3*c
^4*e*h^2*m*z^2 - 81*a^2*b^6*c^2*f*h*l^2*z^2 + 81*a^2*b^5*c^3*f*h^2*l*z^2 -
81*a^2*b^4*c^4*e^2*h*m*z^2 - 1296*a^4*b^2*c^4*e*g*m^2*z^2 - 1296*a^4*b^2*c^
4*d*h*m^2*z^2 + 648*a^3*b^4*c^3*e*g*m^2*z^2 + 648*a^3*b^4*c^3*d*h*m^2*z^2 +
81*a^3*b^3*c^4*d*j*k^2*z^2 - 81*a^2*b^6*c^2*e*g*m^2*z^2 - 81*a^2*b^6*c^2*d
*h*m^2*z^2 + 81*a^2*b^3*c^5*d^2*j*k*z^2 - 567*a^3*b^3*c^4*f*g*k^2*z^2 - 567
*a^2*b^3*c^5*d^2*g*m*z^2 + 486*a^3*b^2*c^5*f*g^2*k*z^2 - 486*a^3*b^2*c^5*e*
g^2*l*z^2 + 486*a^3*b^2*c^5*d*g^2*m*z^2 - 81*a^3*b^3*c^4*e*h*k^2*z^2 + 81*a
^2*b^5*c^3*f*g*k^2*z^2 - 81*a^2*b^4*c^4*f*g^2*k*z^2 + 81*a^2*b^4*c^4*e*g^2*
l*z^2 - 81*a^2*b^4*c^4*d*g^2*m*z^2 - 81*a^2*b^3*c^5*d^2*h*l*z^2 - 567*a^3*b
^3*c^4*e*f*l^2*z^2 - 486*a^3*b^2*c^5*d*h^2*k*z^2 - 162*a^3*b^2*c^5*e*h^2*j*
z^2 - 81*a^3*b^3*c^4*d*g*l^2*z^2 + 81*a^2*b^5*c^3*e*f*l^2*z^2 + 81*a^2*b^4*
c^4*d*h^2*k*z^2 + 81*a^2*b^3*c^5*e^2*h*j*z^2 - 81*a^2*b^3*c^5*e^2*g*k*z^2 +
81*a^2*b^3*c^5*e^2*f*l*z^2 + 1944*a^3*b^3*c^4*d*e*m^2*z^2 - 729*a^2*b^5*c^
3*d*e*m^2*z^2 + 648*a^3*b^2*c^5*e*g*j^2*z^2 + 648*a^3*b^2*c^5*d*h*j^2*z^2 -
81*a^2*b^4*c^4*e*g*j^2*z^2 - 81*a^2*b^4*c^4*d*h*j^2*z^2 + 486*a^3*b^2*c^5*
d*f*k^2*z^2 + 486*a^2*b^2*c^6*d^2*g*j*z^2 - 486*a^2*b^2*c^6*d^2*e*l*z^2 - 1
62*a^2*b^2*c^6*d^2*f*k*z^2 - 81*a^2*b^4*c^4*d*f*k^2*z^2 + 81*a^2*b^3*c^5*d*
g^2*j*z^2 - 486*a^2*b^2*c^6*d*e^2*k*z^2 - 81*a^2*b^3*c^5*e*g^2*h*z^2 - 648*
a^2*b^3*c^5*d*e*j^2*z^2 - 162*a^2*b^2*c^6*e^2*f*h*z^2 + 81*a^2*b^3*c^5*e*f*
h^2*z^2 - 81*a^2*b^3*c^5*d*g*h^2*z^2 - 162*a^2*b^2*c^6*d*f*g^2*z^2 - 189*a^
5*b^3*c^2*l^3*m*z^2 + 162*a^5*b^2*c^3*k^3*m*z^2 - 27*a^4*b^4*c^2*k^3*m*z^2
- 702*a^4*b^3*c^3*j^3*m*z^2 - 81*a^3*b^6*c*j^2*m^2*z^2 + 81*a^3*b^5*c^2*j^3
*m*z^2 - 54*a^5*b^3*c^2*j*m^3*z^2 - 486*a^5*b^2*c^3*j*l^3*z^2 + 216*a^4*b^4
*c^2*j*l^3*z^2 - 189*a^4*b^3*c^3*j*k^3*z^2 - 54*a^4*b^2*c^4*h^3*m*z^2 + 27*
a^3*b^5*c^2*j*k^3*z^2 + 27*a^3*b^3*c^4*g^3*m*z^2 - 810*a^4*b^4*c^2*f*m^3*z^
2 + 540*a^5*b^2*c^3*f*m^3*z^2 - 324*a^3*b^2*c^5*f^3*m*z^2 + 54*a^2*b^4*c^4*
f^3*m*z^2 + 675*a^4*b^3*c^3*f*l^3*z^2 - 243*a^3*b^5*c^2*f*l^3*z^2 - 189*a^2
*b^3*c^5*e^3*m*z^2 + 27*a^3*b^3*c^4*h^3*j*z^2 - 486*a^4*b^2*c^4*f*k^3*z^2 -
486*a^2*b^2*c^6*d^3*m*z^2 + 216*a^3*b^4*c^3*f*k^3*z^2 - 54*a^3*b^2*c^5*g^3
*j*z^2 - 27*a^2*b^6*c^2*f*k^3*z^2 - 270*a^3*b^3*c^4*f*j^3*z^2 - 54*a^2*b^3*
c^5*f^3*j*z^2 + 27*a^2*b^5*c^3*f*j^3*z^2 + 162*a^2*b^2*c^6*e^3*j*z^2 + 162*
a^3*b^2*c^5*f*h^3*z^2 - 27*a^2*b^4*c^4*f*h^3*z^2 + 27*a^2*b^3*c^5*f*g^3*z^2

```
 + 81*a*b^2*c^7*d^2*e^2*z^2 - 648*a^6*c^4*h*l^2*m*z^2 + 648*a^5*c^5*g^2*k*m
*z^2 - 648*a^5*c^5*h^2*j*l*z^2 + 1296*a^5*c^5*h*j^2*k*z^2 + 1296*a^5*c^5*g*
j^2*l*z^2 + 1296*a^5*c^5*f*j^2*m*z^2 - 648*a^5*c^5*g*j*k^2*z^2 + 648*a^5*c^
5*e*k^2*l*z^2 + 648*a^5*c^5*d*k^2*m*z^2 - 648*a^4*c^6*d^2*k*m*z^2 - 648*a^5
*c^5*e*j*l^2*z^2 + 648*a^5*c^5*d*k*l^2*z^2 + 648*a^4*c^6*e^2*j*l*z^2 + 324*
a^6*b*c^3*l^3*m*z^2 + 27*a^4*b^5*c*l^3*m*z^2 + 648*a^5*c^5*f*h*l^2*z^2 - 64
8*a^4*c^6*e^2*h*m*z^2 + 1512*a^5*b*c^4*j^3*m*z^2 + 1080*a^6*b*c^3*j*m^3*z^2
 - 162*a^4*b^5*c*j*m^3*z^2 - 648*a^4*c^6*f*g^2*k*z^2 + 648*a^4*c^6*e*g^2*l*
z^2 - 648*a^4*c^6*d*g^2*m*z^2 - 27*a^3*b^6*c*j*l^3*z^2 + 648*a^4*c^6*e*h^2*
j*z^2 + 648*a^4*c^6*d*h^2*k*z^2 + 324*a^5*b*c^4*j*k^3*z^2 - 1296*a^4*c^6*e*
g*j^2*z^2 - 1296*a^4*c^6*d*h*j^2*z^2 - 108*a^4*b*c^5*g^3*m*z^2 - 648*a^4*c^
6*d*f*k^2*z^2 - 648*a^3*c^7*d^2*g*j*z^2 + 648*a^3*c^7*d^2*f*k*z^2 + 648*a^3
*c^7*d^2*e*l*z^2 + 270*a^3*b^6*c*f*m^3*z^2 + 648*a^3*c^7*d*e^2*k*z^2 - 540*
a^5*b*c^4*f*l^3*z^2 + 324*a^3*b*c^6*e^3*m*z^2 - 108*a^4*b*c^5*h^3*j*z^2 + 2
7*a^2*b^7*c*f*l^3*z^2 + 27*a*b^5*c^4*e^3*m*z^2 + 648*a^3*c^7*e^2*f*h*z^2 +
216*a*b^4*c^5*d^3*m*z^2 + 648*a^4*b*c^5*f*j^3*z^2 + 216*a^3*b*c^6*f^3*j*z^2
 + 648*a^3*c^7*d*f*g^2*z^2 - 27*a*b^4*c^5*e^3*j*z^2 + 324*a^2*b*c^7*d^3*j*z
^2 - 189*a*b^3*c^6*d^3*j*z^2 - 108*a^3*b*c^6*f*g^3*z^2 - 108*a^2*b*c^7*e^3*
f*z^2 + 27*a*b^3*c^6*e^3*f*z^2 + 162*a*b^2*c^7*d^3*f*z^2 - 1134*a^5*b^2*c^3
*j^2*m^2*z^2 + 648*a^4*b^4*c^2*j^2*m^2*z^2 + 81*a^5*b^2*c^3*k^2*l^2*z^2 + 1
62*a^4*b^2*c^4*f^2*m^2*z^2 + 81*a^4*b^2*c^4*h^2*k^2*z^2 + 81*a^4*b^2*c^4*g^
2*l^2*z^2 + 162*a^3*b^2*c^5*f^2*j^2*z^2 + 81*a^3*b^2*c^5*e^2*k^2*z^2 + 81*a
^3*b^2*c^5*d^2*l^2*z^2 + 81*a^3*b^2*c^5*g^2*h^2*z^2 + 81*a^2*b^2*c^6*e^2*g^
2*z^2 + 81*a^2*b^2*c^6*d^2*h^2*z^2 - 216*a^6*c^4*k^3*m*z^2 + 216*a^6*c^4*j*
l^3*z^2 + 27*a^3*b^7*j*m^3*z^2 + 216*a^5*c^5*h^3*m*z^2 + 432*a^6*c^4*f*m^3*
z^2 + 432*a^4*c^6*f^3*m*z^2 - 27*b^6*c^4*d^3*m*z^2 - 27*a^2*b^8*f*m^3*z^2 +
 216*a^5*c^5*f*k^3*z^2 + 216*a^4*c^6*g^3*j*z^2 + 216*a^3*c^7*d^3*m*z^2 + 21
6*a^5*b^4*c*m^4*z^2 - 216*a^3*c^7*e^3*j*z^2 + 27*b^5*c^5*d^3*j*z^2 - 216*a^
4*c^6*f*h^3*z^2 - 27*b^4*c^6*d^3*f*z^2 - 216*a^2*c^8*d^3*f*z^2 - 648*a^6*c^
4*j^2*m^2*z^2 - 324*a^6*c^4*k^2*l^2*z^2 - 648*a^5*c^5*f^2*m^2*z^2 - 324*a^5
*c^5*h^2*k^2*z^2 - 324*a^5*c^5*g^2*l^2*z^2 - 648*a^4*c^6*f^2*j^2*z^2 - 324*
a^4*c^6*e^2*k^2*z^2 - 324*a^4*c^6*d^2*l^2*z^2 - 405*a^6*b^2*c^2*m^4*z^2 - 3
24*a^4*c^6*g^2*h^2*z^2 - 324*a^3*c^7*e^2*g^2*z^2 - 324*a^3*c^7*d^2*h^2*z^2
+ 243*a^4*b^2*c^4*j^4*z^2 - 27*a^3*b^4*c^3*j^4*z^2 - 324*a^2*c^8*d^2*e^2*z^
2 + 27*a^2*b^2*c^6*f^4*z^2 - 108*a^7*c^3*m^4*z^2 - 27*a^4*b^6*m^4*z^2 - 540
*a^5*c^5*j^4*z^2 - 108*a^3*c^7*f^4*z^2 - 216*a^5*b*c^3*f*j*k*l*m*z - 54*a^3
*b^5*c*f*j*k*l*m*z + 27*a^3*b^5*c*g*h*k*l*m*z - 27*a^2*b^6*c*e*g*k*l*m*z -
27*a^2*b^6*c*d*h*k*l*m*z + 432*a^4*b*c^4*d*g*j*k*m*z - 432*a^4*b*c^4*d*e*k*
l*m*z + 216*a^4*b*c^4*e*g*j*k*l*z + 216*a^4*b*c^4*e*f*j*k*m*z + 216*a^4*b*c
^4*d*h*j*k*l*z + 216*a^4*b*c^4*d*f*j*l*m*z + 216*a^4*b*c^4*f*g*h*j*m*z - 27
*a*b^6*c^2*d*e*j*k*l*z - 27*a*b^6*c^2*d*e*h*k*m*z - 27*a*b^6*c^2*d*e*g*l*m*
z + 216*a^3*b*c^5*d*e*h*j*k*z + 216*a^3*b*c^5*d*e*g*j*l*z - 216*a^3*b*c^5*d
*e*f*j*m*z + 27*a*b^5*c^3*d*e*h*j*k*z + 27*a*b^5*c^3*d*e*g*j*l*z + 27*a*b^5
*c^3*d*e*g*h*m*z - 27*a*b^4*c^4*d*e*g*h*j*z + 27*a*b^7*c*d*e*k*l*m*z + 270*
a^4*b^3*c^2*f*j*k*l*m*z - 108*a^4*b^3*c^2*g*h*k*l*m*z - 216*a^4*b^2*c^3*f*h
```

```
*j*k*m*z - 216*a^4*b^2*c^3*f*g*j*l*m*z - 216*a^4*b^2*c^3*e*g*k*l*m*z - 216*
a^4*b^2*c^3*d*h*k*l*m*z + 162*a^3*b^4*c^2*e*g*k*l*m*z + 162*a^3*b^4*c^2*d*h
*k*l*m*z + 108*a^4*b^2*c^3*g*h*j*k*l*z + 108*a^4*b^2*c^3*e*h*j*l*m*z + 54*a
^3*b^4*c^2*f*h*j*k*m*z + 54*a^3*b^4*c^2*f*g*j*l*m*z - 27*a^3*b^4*c^2*g*h*j*
k*l*z + 540*a^3*b^3*c^3*d*e*k*l*m*z - 216*a^2*b^5*c^2*d*e*k*l*m*z - 162*a^3
*b^3*c^3*e*g*j*k*l*z - 162*a^3*b^3*c^3*d*h*j*k*l*z - 108*a^3*b^3*c^3*d*g*j*
k*m*z - 54*a^3*b^3*c^3*e*f*j*k*m*z - 54*a^3*b^3*c^3*d*f*j*l*m*z + 27*a^2*b^
5*c^2*e*g*j*k*l*z + 27*a^2*b^5*c^2*d*h*j*k*l*z - 108*a^3*b^3*c^3*e*g*h*k*m*
z - 108*a^3*b^3*c^3*d*g*h*l*m*z - 54*a^3*b^3*c^3*f*g*h*j*m*z + 27*a^2*b^5*c
^2*e*g*h*k*m*z + 27*a^2*b^5*c^2*d*g*h*l*m*z - 540*a^3*b^2*c^4*d*e*j*k*l*z +
 216*a^2*b^4*c^3*d*e*j*k*l*z - 216*a^3*b^2*c^4*d*e*h*k*m*z - 216*a^3*b^2*c^
4*d*e*g*l*m*z + 162*a^2*b^4*c^3*d*e*h*k*m*z + 162*a^2*b^4*c^3*d*e*g*l*m*z +
 108*a^3*b^2*c^4*e*g*h*j*k*z - 108*a^3*b^2*c^4*e*f*h*j*l*z + 108*a^3*b^2*c^
4*d*g*h*j*l*z + 108*a^3*b^2*c^4*d*f*g*k*m*z - 27*a^2*b^4*c^3*e*g*h*j*k*z -
27*a^2*b^4*c^3*d*g*h*j*l*z - 162*a^2*b^3*c^4*d*e*h*j*k*z - 162*a^2*b^3*c^4*
d*e*g*j*l*z + 54*a^2*b^3*c^4*d*e*f*j*m*z - 108*a^2*b^3*c^4*d*e*g*h*m*z + 10
8*a^2*b^2*c^5*d*e*g*h*j*z + 324*a^6*b*c^2*j*k*l*m^2*z - 81*a^5*b^3*c*j*k*l*
m^2*z + 27*a^4*b^4*c*j^2*k*l*m*z - 27*a^4*b^4*c*h*k^2*l*m*z - 27*a^4*b^4*c*
g*k*l^2*m*z + 216*a^5*b*c^3*h*j^2*k*m*z + 216*a^5*b*c^3*g*j^2*l*m*z + 54*a^
4*b^4*c*f*k*l*m^2*z + 27*a^4*b^4*c*h*j*k*m^2*z + 27*a^4*b^4*c*g*j*l*m^2*z +
 27*a^2*b^6*c*f^2*k*l*m*z + 216*a^5*b*c^3*e*k^2*l*m*z - 108*a^5*b*c^3*h*j*k
^2*l*z + 27*a^3*b^5*c*e*k^2*l*m*z + 216*a^5*b*c^3*d*k*l^2*m*z + 216*a^4*b*c
^4*e^2*j*l*m*z - 108*a^5*b*c^3*g*j*k*l^2*z + 27*a^3*b^5*c*d*k*l^2*m*z - 324
*a^5*b*c^3*e*j*k*m^2*z - 324*a^5*b*c^3*d*j*l*m^2*z - 216*a^5*b*c^3*f*h*l^2*
m*z - 108*a^4*b*c^4*f^2*j*k*l*z - 27*a^3*b^5*c*e*j*k*m^2*z - 27*a^3*b^5*c*d
*j*l*m^2*z - 324*a^5*b*c^3*g*h*j*m^2*z + 216*a^5*b*c^3*f*h*k*m^2*z + 216*a^
5*b*c^3*f*g*l*m^2*z + 216*a^5*b*c^3*e*h*l*m^2*z - 216*a^4*b*c^4*f^2*h*k*m*z
 - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5*c*g*h*j*m^2*z + 216*a^4*b*c^4*e*g
^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 216*a^4*b*c^4*f*h^2*j*l*z + 216*a^4*
b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k*m*z - 108*a^4*b*c^4*g*h^2*j*k*z -
 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^4*d*h*j^2*m*z + 216*a^4*b*c^4*f*h*
j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*a^2*b^6*c*e*g*j*m^2*z + 27*a^2*b^6
*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*z - 216*a^4*b*c^4*f*g*j*k^2*z + 21
6*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d^2*e*l*m*z - 108*a^4*b*c^4*e*h*j*k
^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^3*b*c^5*d^2*h*j*l*z + 108*a^3*b*c^
5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z + 27*a*b^5*c^3*d^2*g*k*l*z + 27*a*
b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j*l^2*z + 216*a^3*b*c^5*d*e^2*k*m*z
 - 108*a^4*b*c^4*d*g*j*l^2*z - 108*a^3*b*c^5*e^2*g*j*k*z + 27*a*b^5*c^3*d*e
^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^2*z + 216*a^3*b*c^5*e^2*f*h*m*z - 108*a^4*
b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5*e^2*g*h*l*z + 108*a^3*b*c^5*e*f^2*j*k*z +
 108*a^3*b*c^5*d*f^2*j*l*z + 27*a*b^6*c^2*d*e*j^2*m*z - 216*a^3*b*c^5*e*f^2
*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*z - 27*a*b^4*c^4*d^2*e*j*l*z + 216*a^3*b*c
^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*g^2*h*j*z + 54*a*b^4*c^4*d^2*f*g*m*z - 27*
a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*c^4*d^2*e*h*m*z - 27*a*b^4*c^4*d*e^2*j*k*z
 - 108*a^3*b*c^5*d*g*h^2*j*z + 54*a*b^4*c^4*d*e^2*h*l*z + 27*a*b^6*c^2*d*e*
```

h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*z - 27*a*b^4*c^4*d*e^2*g*m*z - 27*a*b^4*c^
4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2*f*g*j*z - 108*a^3*b*c^5*d*e*g*k^2*z - 108
*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*b*c^6*d^2*e*g*k*z - 54*a*b^3*c^5*d^2*f*g*j
*z - 27*a*b^5*c^3*d*e*g*k^2*z + 27*a*b^4*c^4*d*e*g^2*k*z + 27*a*b^3*c^5*d^2
*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k*z - 108*a^2*b*c^6*d*e^2*g*j*z + 27*a*b^3*
c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d*e*f^2*j*z + 27*a*b^3*c^5*d*e*f^2*j*z - 43
2*a^5*c^4*e*h*j*l*m*z + 432*a^4*c^5*d*e*j*k*l*z + 432*a^4*c^5*e*f*h*j*l*z -
432*a^4*c^5*d*f*g*k*m*z - 27*a*b^7*c*d*e*j*m^2*z - 54*a^5*b^2*c^2*j^2*k*l*
m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z + 108*a^5*b^2*c^2*g*k*l^2*m*z - 54*a^5*b^
2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c^3*f^2*k*l*m*z - 270*a^5*b^2*c^2*f*k*l*m^2
*z - 189*a^3*b^4*c^2*f^2*k*l*m*z - 108*a^5*b^2*c^2*h*j*k*m^2*z - 108*a^5*b^
2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^2*h*j^2*k*m*z - 54*a^4*b^3*c^2*g*j^2*l*m*z
- 162*a^4*b^3*c^2*e*k^2*l*m*z + 54*a^4*b^2*c^3*g^2*j*k*m*z + 27*a^4*b^3*c^
2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d*k*l^2*m*z + 108*a^4*b^2*c^3*g^2*h*l*m*z -
54*a^3*b^3*c^3*e^2*j*l*m*z + 27*a^4*b^3*c^2*g*j*k*l^2*z - 27*a^3*b^4*c^2*g
^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^2*k*l*z + 189*a^4*b^3*c^2*e*j*k*m^2*z + 18
9*a^4*b^3*c^2*d*j*l*m^2*z - 162*a^4*b^2*c^3*e*j^2*k*m*z - 162*a^4*b^2*c^3*d
*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*j*k*l*z + 108*a^4*b^2*c^3*g*h^2*k*m*z + 54
*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4*b^2*c^3*f*h^2*l*m*z + 54*a^3*b^4*c^2*f*j^
2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*m*z + 27*a^3*b^4*c^2*e*j^2*k*m*z + 27*a^3*
b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*c^2*f^2*j*k*l*z - 270*a^3*b^2*c^4*d^2*j*k*
m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z - 162*a^4*b^2*c^3*g*h*j^2*m*z + 162*a^4*b
^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*c^3*f^2*h*k*m*z + 162*a^3*b^3*c^3*f^2*g*l*
m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z - 54*a^4*b^3*c^2*f*g*l*m^2*z - 54*a^4*b^3*
c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*d*j*k^2*m*z + 54*a^2*b^4*c^3*d^2*j*k*m*z +
27*a^3*b^4*c^2*g*h*j^2*m*z - 27*a^3*b^4*c^2*e*j*k^2*l*z - 27*a^2*b^5*c^2*f
^2*h*k*m*z - 27*a^2*b^5*c^2*f^2*g*l*m*z + 162*a^4*b^2*c^3*d*j*k*l^2*z - 162
*a^3*b^3*c^3*e*g^2*l*m*z + 108*a^4*b^2*c^3*e*h*k^2*m*z + 108*a^3*b^2*c^4*d^
2*h*l*m*z - 54*a^4*b^2*c^3*f*g*k^2*m*z - 27*a^3*b^4*c^2*e*h*k^2*m*z - 27*a^
3*b^4*c^2*d*j*k*l^2*z + 27*a^3*b^3*c^3*g^2*h*j*l*z + 27*a^2*b^5*c^2*e*g^2*l
*m*z - 27*a^2*b^4*c^3*d^2*h*l*m*z + 270*a^4*b^2*c^3*f*h*j*l^2*z - 270*a^3*b
^2*c^4*e^2*h*j*m*z - 162*a^4*b^2*c^3*e*h*k*l^2*z - 162*a^3*b^3*c^3*d*h^2*k*
m*z + 162*a^3*b^2*c^4*e^2*h*k*l*z + 108*a^4*b^2*c^3*d*g*l^2*m*z + 108*a^3*b
^2*c^4*e^2*g*k*m*z - 54*a^4*b^2*c^3*e*f*l^2*m*z - 54*a^3*b^4*c^2*f*h*j*l^2*
z + 54*a^3*b^3*c^3*f*h^2*j*l*z - 54*a^3*b^3*c^3*e*h^2*j*m*z + 54*a^3*b^2*c^
4*e^2*f*l*m*z + 54*a^2*b^4*c^3*e^2*h*j*m*z + 27*a^3*b^4*c^2*e*h*k*l^2*z - 2
7*a^3*b^4*c^2*d*g*l^2*m*z + 27*a^3*b^3*c^3*g*h^2*j*k*z + 27*a^2*b^5*c^2*d*h
^2*k*m*z - 27*a^2*b^4*c^3*e^2*h*k*l*z - 27*a^2*b^4*c^3*e^2*g*k*m*z + 432*a^
4*b^2*c^3*e*g*j*m^2*z + 432*a^4*b^2*c^3*d*h*j*m^2*z - 270*a^4*b^2*c^3*d*g*k
*m^2*z - 216*a^3*b^4*c^2*e*g*j*m^2*z - 216*a^3*b^4*c^2*d*h*j*m^2*z + 216*a^
3*b^3*c^3*e*g*j^2*m*z + 216*a^3*b^3*c^3*d*h*j^2*m*z - 162*a^3*b^2*c^4*e*f^2
*k*m*z - 162*a^3*b^2*c^4*d*f^2*l*m*z - 108*a^3*b^2*c^4*f^2*h*j*k*z - 108*a^
3*b^2*c^4*f^2*g*j*l*z + 54*a^4*b^2*c^3*e*f*k*m^2*z + 54*a^4*b^2*c^3*d*f*l*m
^2*z + 54*a^3*b^4*c^2*d*g*k*m^2*z - 54*a^3*b^3*c^3*f*h*j^2*k*z - 54*a^3*b^3
*c^3*f*g*j^2*l*z - 27*a^2*b^5*c^2*e*g*j^2*m*z - 27*a^2*b^5*c^2*d*h*j^2*m*z

+ 27*a^2*b^4*c^3*f^2*h*j*k*z + 27*a^2*b^4*c^3*f^2*g*j*l*z + 27*a^2*b^4*c^3*
e*f^2*k*m*z + 27*a^2*b^4*c^3*d*f^2*l*m*z + 324*a^2*b^3*c^4*d^2*g*j*m*z - 27
0*a^3*b^2*c^4*d*g^2*j*m*z - 162*a^3*b^2*c^4*f^2*g*h*m*z + 162*a^3*b^2*c^4*e
*g^2*j*l*z - 162*a^2*b^3*c^4*d^2*e*l*m*z - 135*a^2*b^3*c^4*d^2*g*k*l*z + 10
8*a^3*b^2*c^4*d*g^2*k*l*z + 54*a^4*b^2*c^3*f*g*h*m^2*z + 54*a^3*b^3*c^3*f*g
*j*k^2*z - 54*a^3*b^2*c^4*f*g^2*j*k*z + 54*a^2*b^4*c^3*d*g^2*j*m*z - 54*a^2
*b^3*c^4*d^2*f*k*m*z + 27*a^3*b^3*c^3*e*h*j*k^2*z + 27*a^3*b^3*c^3*d*g*k^2*
l*z + 27*a^2*b^4*c^3*f^2*g*h*m*z - 27*a^2*b^4*c^3*e*g^2*j*l*z - 27*a^2*b^4*
c^3*d*g^2*k*l*z + 27*a^2*b^3*c^4*d^2*h*j*l*z + 162*a^3*b^2*c^4*d*h^2*j*k*z
- 162*a^2*b^3*c^4*d*e^2*k*m*z + 108*a^3*b^2*c^4*e*g^2*h*m*z + 54*a^3*b^3*c^
3*e*f*j*l^2*z + 27*a^3*b^3*c^3*d*g*j*l^2*z - 27*a^2*b^4*c^3*e*g^2*h*m*z - 2
7*a^2*b^4*c^3*d*h^2*j*k*z + 27*a^2*b^3*c^4*e^2*g*j*k*z - 621*a^3*b^3*c^3*d*
e*j*m^2*z + 594*a^3*b^2*c^4*d*e*j^2*m*z + 243*a^2*b^5*c^2*d*e*j*m^2*z - 243
*a^2*b^4*c^3*d*e*j^2*m*z + 135*a^3*b^3*c^3*e*g*h*l^2*z - 108*a^3*b^2*c^4*e*
g*h^2*l*z + 108*a^3*b^2*c^4*d*g*h^2*m*z + 54*a^3*b^2*c^4*e*f*j^2*k*z + 54*a
^3*b^2*c^4*e*f*h^2*m*z + 54*a^3*b^2*c^4*d*g*j^2*k*z + 54*a^3*b^2*c^4*d*f*j^
2*l*z - 54*a^2*b^3*c^4*e^2*f*h*m*z - 27*a^2*b^5*c^2*e*g*h*l^2*z + 27*a^2*b^
4*c^3*e*g*h^2*l*z - 27*a^2*b^4*c^3*d*g*h^2*m*z - 27*a^2*b^3*c^4*e^2*g*h*l*z
- 27*a^2*b^3*c^4*e*f^2*j*k*z - 27*a^2*b^3*c^4*d*f^2*j*l*z + 162*a^2*b^2*c^
5*d^2*e*j*l*z + 54*a^3*b^2*c^4*f*g*h*j^2*z - 54*a^3*b^2*c^4*d*f*j*k^2*z + 5
4*a^2*b^3*c^4*e*f^2*h*l*z + 54*a^2*b^2*c^5*d^2*f*j*k*z - 27*a^2*b^3*c^4*f^2
*g*h*j*z - 270*a^2*b^2*c^5*d^2*f*g*m*z - 162*a^3*b^2*c^4*d*g*h*k^2*z + 162*
a^2*b^2*c^5*d^2*g*h*k*z + 162*a^2*b^2*c^5*d*e^2*j*k*z + 108*a^2*b^2*c^5*d^2
*e*h*m*z - 54*a^2*b^3*c^4*d*f*g^2*m*z + 27*a^2*b^4*c^3*d*g*h*k^2*z + 27*a^2
*b^3*c^4*e*g^2*h*j*z + 270*a^3*b^2*c^4*d*e*h*l^2*z - 270*a^2*b^2*c^5*d*e^2*
h*l*z - 162*a^2*b^4*c^3*d*e*h*l^2*z + 108*a^2*b^3*c^4*d*e*h^2*l*z + 108*a^2
*b^2*c^5*d*e^2*g*m*z + 54*a^2*b^2*c^5*e^2*f*h*j*z + 27*a^2*b^3*c^4*d*g*h^2*
j*z + 162*a^2*b^2*c^5*d*e*f^2*m*z - 54*a^3*b^2*c^4*d*e*f*m^2*z - 54*a^2*b^2
*c^5*d*f^2*g*k*z + 135*a^2*b^3*c^4*d*e*g*k^2*z - 108*a^2*b^2*c^5*d*e*g^2*k*
z + 54*a^2*b^2*c^5*d*f*g^2*j*z - 54*a^2*b^2*c^5*d*e*f*j^2*z - 9*a*b^7*c*d*e
*l^3*z - 36*a*b*c^7*d^3*e*g*z - 108*a^6*b*c^2*k^2*l^2*m*z + 27*a^5*b^3*c*k^
2*l^2*m*z - 18*a^5*b^2*c^2*j*k^3*m*z - 27*a^4*b^3*c^2*j^3*k*l*z - 108*a^5*b
*c^3*h^2*k^2*m*z - 108*a^5*b*c^3*g^2*l^2*m*z + 108*a^5*b*c^3*h^2*k*l^2*z +
108*a^5*b*c^3*g^2*k*m^2*z + 90*a^5*b^2*c^2*f*l^3*m*z - 18*a^5*b^2*c^2*h*k*l
^3*z + 18*a^4*b^2*c^3*h^3*k*l*z + 18*a^4*b^2*c^3*h^3*j*m*z - 108*a^5*b*c^3*
h*j^2*l^2*z + 18*a^4*b^3*c^2*f*k^3*m*z - 18*a^3*b^3*c^3*g^3*j*m*z - 9*a^4*b
^3*c^2*g*k^3*l*z + 9*a^3*b^3*c^3*g^3*k*l*z + 252*a^4*b^2*c^3*f*j^3*m*z + 21
6*a^5*b*c^3*f*j^2*m^2*z + 180*a^3*b^2*c^4*f^3*j*m*z - 108*a^4*b*c^4*e^2*k^2
*m*z - 108*a^4*b*c^4*d^2*l^2*m*z + 90*a^5*b^2*c^2*e*k*m^3*z + 90*a^5*b^2*c^
2*d*l*m^3*z - 90*a^3*b^2*c^4*f^3*k*l*z + 54*a^3*b^5*c*f*j^2*m^2*z - 54*a^3*
b^4*c^2*f*j^3*m*z + 36*a^5*b^2*c^2*f*j*m^3*z + 36*a^4*b^2*c^3*h*j^3*k*z + 3
6*a^4*b^2*c^3*g*j^3*l*z - 36*a^2*b^4*c^3*f^3*j*m*z - 27*a^2*b^6*c*f^2*j*m^2
*z + 18*a^2*b^4*c^3*f^3*k*l*z - 216*a^4*b*c^4*d^2*k*m^2*z + 108*a^5*b*c^3*d
*k^2*m^2*z - 108*a^4*b^3*c^2*f*j*l^3*z - 108*a^4*b*c^4*g^2*h^2*m*z + 108*a^
2*b^3*c^4*e^3*j*m*z + 90*a^5*b^2*c^2*g*h*m^3*z + 54*a^4*b^3*c^2*e*k*l^3*z -

54*a^2*b^3*c^4*e^3*k*l*z + 234*a^2*b^2*c^5*d^3*j*m*z - 144*a^2*b^2*c^5*d^3
*k*l*z + 90*a^4*b^2*c^3*f*j*k^3*z - 72*a^4*b^2*c^3*d*k^3*l*z + 27*a^4*b^3*c
^2*g*h*l^3*z - 27*a^3*b^3*c^3*g*h^3*l*z - 18*a^3*b^4*c^2*f*j*k^3*z + 9*a^3*
b^4*c^2*d*k^3*l*z + 216*a^4*b*c^4*f^2*h*l^2*z - 216*a^4*b*c^4*e^2*h*m^2*z +
108*a^4*b*c^4*g^2*h*k^2*z - 18*a^4*b^2*c^3*g*h*k^3*z + 18*a^3*b^2*c^4*g^3*
h*k*z + 18*a^3*b^2*c^4*f*g^3*m*z + 9*a^3*b^4*c^2*g*h*k^3*z - 9*a^3*b^3*c^3*
e*j^3*k*z - 9*a^3*b^3*c^3*d*j^3*l*z - 144*a^4*b^3*c^2*e*g*m^3*z - 144*a^4*b
^3*c^2*d*h*m^3*z - 108*a^3*b*c^5*e^2*g^2*m*z + 108*a^3*b*c^5*d^2*j^2*k*z -
108*a^3*b*c^5*d^2*h^2*m*z - 18*a^2*b^3*c^4*f^3*h*k*z - 18*a^2*b^3*c^4*f^3*g
*l*z - 9*a^3*b^3*c^3*g*h*j^3*z - 216*a^4*b*c^4*d*g^2*m^2*z + 144*a^4*b^2*c^
3*e*g*l^3*z - 126*a^3*b^2*c^4*d*h^3*l*z - 108*a^4*b*c^4*d*h^2*l^2*z - 108*a
^3*b*c^5*f^2*g^2*k*z - 108*a^3*b*c^5*e^2*h^2*k*z - 90*a^2*b^2*c^5*e^3*f*m*z
+ 72*a^2*b^2*c^5*e^3*g*l*z - 63*a^3*b^4*c^2*e*g*l^3*z - 36*a^3*b^4*c^2*d*h
*l^3*z + 27*a^2*b^4*c^3*d*h^3*l*z + 27*a*b^6*c^2*d^2*g*m^2*z - 18*a^4*b^2*c
^3*d*h*l^3*z - 18*a^3*b^2*c^4*f*h^3*j*z - 18*a^3*b^2*c^4*e*h^3*k*z + 18*a^2
*b^2*c^5*e^3*h*k*z + 108*a^3*b*c^5*e^2*h*j^2*z + 54*a^3*b^3*c^3*d*h*k^3*z +
27*a^3*b^3*c^3*e*g*k^3*z - 27*a^2*b^3*c^4*e*g^3*k*z + 27*a^2*b^3*c^4*d*g^3
*l*z - 27*a*b^4*c^4*d^2*g^2*l*z - 9*a^2*b^5*c^2*e*g*k^3*z - 9*a^2*b^5*c^2*d
*h*k^3*z + 207*a^3*b^4*c^2*d*e*m^3*z - 108*a^2*b*c^6*d^2*e^2*m*z - 90*a^4*b
^2*c^3*d*e*m^3*z - 72*a^3*b^2*c^4*e*g*j^3*z - 72*a^3*b^2*c^4*d*h*j^3*z + 27
*a*b^3*c^5*d^2*e^2*m*z + 18*a^2*b^2*c^5*e*f^3*k*z + 18*a^2*b^2*c^5*d*f^3*l*
z + 9*a^2*b^4*c^3*e*g*j^3*z + 9*a^2*b^4*c^3*d*h*j^3*z - 216*a^3*b*c^5*d*e^2
*l^2*z - 198*a^3*b^3*c^3*d*e*l^3*z + 108*a^3*b*c^5*d*g^2*j^2*z - 108*a^3*b*
c^5*d*f^2*k^2*z + 72*a^2*b^5*c^2*d*e*l^3*z - 27*a*b^5*c^3*d*e^2*l^2*z + 27*
a*b^4*c^4*d^2*g*j^2*z + 18*a^2*b^2*c^5*f^3*g*h*z + 144*a^3*b^2*c^4*d*e*k^3*
z - 63*a^2*b^4*c^3*d*e*k^3*z + 27*a*b^4*c^4*d^2*e*k^2*z - 9*a^2*b^3*c^4*e*g
*h^3*z - 108*a^2*b*c^6*d^2*g^2*h*z + 81*a^2*b^3*c^4*d*e*j^3*z + 27*a*b^3*c^
5*d^2*g^2*h*z - 27*a*b^2*c^6*d^2*e^2*j*z - 18*a^2*b^2*c^5*d*g^3*h*z + 108*a
^2*b*c^6*d*e^2*h^2*z - 27*a*b^3*c^5*d*e^2*h^2*z + 27*a*b^2*c^6*d^2*f^2*g*z
- 18*a^2*b^2*c^5*d*e*h^3*z - 216*a^6*c^3*j^2*k*l*m*z + 216*a^6*c^3*h*j*l^2*
m*z + 216*a^6*c^3*f*k*l*m^2*z - 216*a^5*c^4*f^2*k*l*m*z - 216*a^5*c^4*g^2*j
*k*m*z + 216*a^5*c^4*f*j^2*k*l*z + 216*a^5*c^4*f*h^2*l*m*z + 216*a^5*c^4*e*
j^2*k*m*z + 216*a^5*c^4*d*j^2*l*m*z + 216*a^5*c^4*g*h*j^2*m*z - 216*a^5*c^4
*e*j*k^2*l*z - 216*a^5*c^4*d*j*k^2*m*z + 216*a^4*c^5*d^2*j*k*m*z - 18*a^6*b
^2*c*k*l*m^3*z + 216*a^5*c^4*f*g*k^2*m*z - 216*a^5*c^4*d*j*k*l^2*z - 72*a^6
*b*c^2*j*l^3*m*z + 18*a^5*b^3*c*j*l^3*m*z - 216*a^5*c^4*f*h*j*l^2*z + 216*a
^5*c^4*e*h*k*l^2*z + 216*a^5*c^4*e*f*l^2*m*z - 216*a^4*c^5*e^2*h*k*l*z + 21
6*a^4*c^5*e^2*h*j*m*z - 216*a^4*c^5*e^2*f*l*m*z - 216*a^5*c^4*e*f*k*m^2*z +
216*a^5*c^4*d*g*k*m^2*z - 216*a^5*c^4*d*f*l*m^2*z + 216*a^4*c^5*e*f^2*k*m*
z + 216*a^4*c^5*d*f^2*l*m*z + 108*a^5*b*c^3*j^3*k*l*z - 216*a^5*c^4*f*g*h*m
^2*z + 216*a^4*c^5*f^2*g*h*m*z + 216*a^4*c^5*f*g^2*j*k*z - 216*a^4*c^5*e*g^
2*j*l*z + 216*a^4*c^5*d*g^2*j*m*z - 72*a^6*b*c^2*h*k*m^3*z - 72*a^6*b*c^2*g
*l*m^3*z + 54*a^5*b^3*c*h*k*m^3*z + 54*a^5*b^3*c*g*l*m^3*z - 216*a^4*c^5*d*
h^2*j*k*z - 18*a^4*b^4*c*f*l^3*m*z + 9*a^4*b^4*c*h*k*l^3*z - 216*a^4*c^5*e*
f*j^2*k*z - 216*a^4*c^5*e*f*h^2*m*z - 216*a^4*c^5*d*g*j^2*k*z - 216*a^4*c^5

```
*d*f*j^2*l*z - 216*a^4*c^5*d*e*j^2*m*z - 72*a^5*b*c^3*f*k^3*m*z + 72*a^4*b*
c^4*g^3*j*m*z + 36*a^5*b*c^3*g*k^3*l*z - 36*a^4*b*c^4*g^3*k*l*z - 216*a^4*c
^5*f*g*h*j^2*z + 216*a^4*c^5*d*f*j*k^2*z - 216*a^3*c^6*d^2*f*j*k*z - 216*a^
3*c^6*d^2*e*j*l*z + 72*a^4*b^4*c*f*j*m^3*z - 63*a^4*b^4*c*e*k*m^3*z - 63*a^
4*b^4*c*d*l*m^3*z + 216*a^4*c^5*d*g*h*k^2*z - 216*a^3*c^6*d^2*g*h*k*z + 216
*a^3*c^6*d^2*f*g*m*z - 216*a^3*c^6*d*e^2*j*k*z + 144*a^5*b*c^3*f*j*l^3*z -
144*a^3*b*c^5*e^3*j*m*z - 72*a^5*b*c^3*e*k*l^3*z + 72*a^3*b*c^5*e^3*k*l*z -
 63*a^4*b^4*c*g*h*m^3*z + 18*a^3*b^5*c*f*j*l^3*z - 18*a*b^5*c^3*e^3*j*m*z -
 9*a^3*b^5*c*e*k*l^3*z + 9*a*b^5*c^3*e^3*k*l*z - 216*a^4*c^5*d*e*h*l^2*z -
216*a^3*c^6*e^2*f*h*j*z + 216*a^3*c^6*d*e^2*h*l*z - 126*a*b^4*c^4*d^3*j*m*z
 + 108*a^4*b*c^4*g*h^3*l*z + 63*a*b^4*c^4*d^3*k*l*z + 36*a^5*b*c^3*g*h*l^3*
z - 9*a^3*b^5*c*g*h*l^3*z + 216*a^4*c^5*d*e*f*m^2*z + 216*a^3*c^6*d*f^2*g*k
*z - 216*a^3*c^6*d*e*f^2*m*z + 36*a^4*b*c^4*e*j^3*k*z + 36*a^4*b*c^4*d*j^3*
l*z - 216*a^3*c^6*d*f*g^2*j*z + 72*a^3*b^5*c*e*g*m^3*z + 72*a^3*b^5*c*d*h*m
^3*z + 72*a^3*b*c^5*f^3*h*k*z + 72*a^3*b*c^5*f^3*g*l*z + 36*a^4*b*c^4*g*h*j
^3*z + 18*a*b^4*c^4*e^3*f*m*z + 9*a^2*b^6*c*e*g*l^3*z + 9*a^2*b^6*c*d*h*l^3
*z - 9*a*b^4*c^4*e^3*h*k*z - 9*a*b^4*c^4*e^3*g*l*z + 216*a^3*c^6*d*e*f*j^2*
z - 144*a^2*b*c^6*d^3*f*m*z + 108*a^3*b*c^5*e*g^3*k*z - 108*a^3*b*c^5*d*g^3
*l*z + 108*a*b^3*c^5*d^3*f*m*z - 72*a^4*b*c^4*d*h*k^3*z + 72*a^2*b*c^6*d^3*
h*k*z - 54*a*b^3*c^5*d^3*h*k*z + 36*a^4*b*c^4*e*g*k^3*z - 36*a^2*b*c^6*d^3*
g*l*z - 27*a*b^3*c^5*d^3*g*l*z - 81*a^2*b^6*c*d*e*m^3*z + 216*a^4*b*c^4*d*e
*l^3*z + 72*a^2*b*c^6*e^3*f*j*z + 72*a^2*b*c^6*d*e^3*l*z - 18*a*b^3*c^5*e^3
*f*j*z - 18*a*b^3*c^5*d*e^3*l*z - 90*a*b^2*c^6*d^3*f*j*z + 72*a*b^2*c^6*d^3
*e*k*z + 36*a^3*b*c^5*e*g*h^3*z - 36*a^2*b*c^6*e^3*g*h*z + 9*a*b^6*c^2*d*e*
k^3*z + 9*a*b^3*c^5*e^3*g*h*z - 180*a^3*b*c^5*d*e*j^3*z + 18*a*b^2*c^6*d^3*
g*h*z - 9*a*b^5*c^3*d*e*j^3*z + 18*a*b^2*c^6*d*e^3*h*z + 9*a*b^4*c^4*d*e*h^
3*z + 36*a^2*b*c^6*d*e*g^3*z - 9*a*b^3*c^5*d*e*g^3*z - 18*a*b^2*c^6*d*e*f^3
*z + 27*a^5*b^2*c^2*h^2*l*m^2*z - 27*a^5*b^2*c^2*j*k^2*l^2*z + 27*a^4*b^3*c
^2*h^2*k^2*m*z + 27*a^4*b^3*c^2*g^2*l^2*m*z + 27*a^5*b^2*c^2*g*k^2*m^2*z -
27*a^4*b^3*c^2*h^2*k*l^2*z - 27*a^4*b^3*c^2*g^2*k*m^2*z - 135*a^4*b^2*c^3*e
^2*l*m^2*z + 27*a^5*b^2*c^2*e*l^2*m^2*z + 27*a^4*b^3*c^2*h*j^2*l^2*z - 27*a
^4*b^2*c^3*h^2*j^2*l*z + 27*a^3*b^4*c^2*e^2*l*m^2*z - 270*a^4*b^3*c^2*f*j^2
*m^2*z - 270*a^4*b^2*c^3*f^2*j*m^2*z + 162*a^3*b^4*c^2*f^2*j*m^2*z - 108*a^
3*b^3*c^3*f^2*j^2*m*z - 27*a^4*b^2*c^3*h^2*j*k^2*z - 27*a^4*b^2*c^3*g^2*j*l
^2*z + 27*a^3*b^3*c^3*e^2*k^2*m*z + 27*a^3*b^3*c^3*d^2*l^2*m*z + 27*a^2*b^5
*c^2*f^2*j^2*m*z + 162*a^3*b^3*c^3*d^2*k*m^2*z - 27*a^4*b^3*c^2*d*k^2*m^2*z
 - 27*a^4*b^2*c^3*g*j^2*k^2*z + 27*a^3*b^3*c^3*g^2*h^2*m*z - 27*a^2*b^5*c^2
*d^2*k*m^2*z + 162*a^3*b^2*c^4*d^2*k^2*l*z - 108*a^4*b^2*c^3*g*h^2*l^2*z -
27*a^4*b^2*c^3*e*j^2*l^2*z + 27*a^3*b^4*c^2*g*h^2*l^2*z + 27*a^3*b^2*c^4*e^
2*j^2*l*z - 27*a^2*b^4*c^3*d^2*k^2*l*z - 162*a^3*b^3*c^3*f^2*h*l^2*z + 162*
a^3*b^3*c^3*e^2*h*m^2*z - 135*a^4*b^2*c^3*e*h^2*m^2*z + 135*a^3*b^2*c^4*f^2
*h^2*l*z + 27*a^3*b^4*c^2*e*h^2*m^2*z - 27*a^3*b^3*c^3*g^2*h*k^2*z - 27*a^3
*b^2*c^4*e^2*j*k^2*z - 27*a^3*b^2*c^4*d^2*j*l^2*z + 27*a^2*b^5*c^2*f^2*h*l^
2*z - 27*a^2*b^5*c^2*e^2*h*m^2*z - 27*a^2*b^4*c^3*f^2*h^2*l*z - 27*a^3*b^2*
c^4*g^2*h^2*j*z + 27*a^2*b^3*c^4*e^2*g^2*m*z - 27*a^2*b^3*c^4*d^2*j^2*k*z +
```

27*a^2*b^3*c^4*d^2*h^2*m*z + 351*a^3*b^2*c^4*d^2*g*m^2*z - 189*a^2*b^4*c^3
*d^2*g*m^2*z + 162*a^3*b^3*c^3*d*g^2*m^2*z - 162*a^3*b^2*c^4*e^2*g*l^2*z +
135*a^3*b^3*c^3*d*h^2*l^2*z + 135*a^3*b^2*c^4*f^2*g*k^2*z - 27*a^2*b^5*c^2*
d*h^2*l^2*z - 27*a^2*b^5*c^2*d*g^2*m^2*z - 27*a^2*b^4*c^3*f^2*g*k^2*z + 27*
a^2*b^4*c^3*e^2*g*l^2*z + 27*a^2*b^3*c^4*f^2*g^2*k*z + 27*a^2*b^3*c^4*e^2*h
^2*k*z + 135*a^3*b^2*c^4*e*f^2*l^2*z - 108*a^3*b^2*c^4*e*g^2*k^2*z + 108*a^
2*b^2*c^5*d^2*g^2*l*z + 27*a^3*b^2*c^4*e*h^2*j^2*z + 27*a^2*b^4*c^3*e*g^2*k
^2*z - 27*a^2*b^4*c^3*e*f^2*l^2*z - 27*a^2*b^3*c^4*e^2*h*j^2*z - 27*a^2*b^2
*c^5*e^2*f^2*l*z - 27*a^2*b^2*c^5*e^2*g^2*j*z - 27*a^2*b^2*c^5*d^2*h^2*j*z
+ 162*a^2*b^3*c^4*d*e^2*l^2*z - 135*a^2*b^2*c^5*d^2*g*j^2*z - 27*a^2*b^3*c^
4*d*g^2*j^2*z + 27*a^2*b^3*c^4*d*f^2*k^2*z - 162*a^2*b^2*c^5*d^2*e*k^2*z -
27*a^2*b^2*c^5*e*f^2*h^2*z - 72*a^7*c^2*k*l*m^3*z + 9*a^5*b^4*k*l*m^3*z + 7
2*a^6*c^3*j*k^3*m*z - 72*a^6*c^3*h*k*l^3*z - 72*a^6*c^3*f*l^3*m*z - 72*a^5*
c^4*h^3*k*l*z - 72*a^5*c^4*h^3*j*m*z - 9*a^4*b^5*h*k*m^3*z - 9*a^4*b^5*g*l*
m^3*z - 144*a^6*c^3*f*j*m^3*z - 144*a^5*c^4*h*j^3*k*z - 144*a^5*c^4*g*j^3*l
*z - 144*a^5*c^4*f*j^3*m*z - 144*a^4*c^5*f^3*j*m*z + 72*a^6*c^3*e*k*m^3*z +
72*a^6*c^3*d*l*m^3*z + 72*a^4*c^5*f^3*k*l*z + 72*a^6*c^3*g*h*m^3*z + 18*b^
6*c^3*d^3*j*m*z - 18*a^3*b^6*f*j*m^3*z - 9*b^6*c^3*d^3*k*l*z + 9*a^3*b^6*e*
k*m^3*z + 9*a^3*b^6*d*l*m^3*z + 144*a^5*c^4*d*k^3*l*z + 144*a^3*c^6*d^3*k*l
*z - 72*a^5*c^4*f*j*k^3*z - 72*a^3*c^6*d^3*j*m*z + 9*a^3*b^6*g*h*m^3*z - 72
*a^5*c^4*g*h*k^3*z - 72*a^4*c^5*g^3*h*k*z - 72*a^4*c^5*f*g^3*m*z - 108*a^5*
b*c^3*j^4*m*z + 63*a^6*b^2*c*j*m^4*z + 36*a^6*b*c^2*k*l^4*z - 9*a^5*b^3*c*k
*l^4*z - 144*a^5*c^4*e*g*l^3*z - 144*a^3*c^6*e^3*g*l*z + 72*a^5*c^4*d*h*l^3
*z + 72*a^4*c^5*f*h^3*j*z + 72*a^4*c^5*e*h^3*k*z + 72*a^4*c^5*d*h^3*l*z + 7
2*a^3*c^6*e^3*h*k*z + 72*a^3*c^6*e^3*f*m*z - 18*b^5*c^4*d^3*f*m*z + 9*b^5*c
^4*d^3*h*k*z + 9*b^5*c^4*d^3*g*l*z - 9*a^2*b^7*e*g*m^3*z - 9*a^2*b^7*d*h*m^
3*z + 144*a^4*c^5*e*g*j^3*z + 144*a^4*c^5*d*h*j^3*z - 72*a^5*c^4*d*e*m^3*z
- 72*a^3*c^6*e*f^3*k*z - 72*a^3*c^6*d*f^3*l*z + 144*a^6*b*c^2*f*m^4*z - 108
*a^5*b^3*c*f*m^4*z - 72*a^3*c^6*f^3*g*h*z + 36*a^5*b*c^3*h*k^4*z - 36*a^3*b
*c^5*f^4*m*z + 18*b^4*c^5*d^3*f*j*z - 9*b^4*c^5*d^3*e*k*z + 9*a^4*b^4*c*g*l
^4*z - 144*a^4*c^5*d*e*k^3*z - 144*a^2*c^7*d^3*e*k*z + 72*a^2*c^7*d^3*f*j*z
- 9*b^4*c^5*d^3*g*h*z + 72*a^3*c^6*d*g^3*h*z + 72*a^2*c^7*d^3*g*h*z - 72*a
^5*b*c^3*d*l^4*z - 72*a^4*b*c^4*f*j^4*z + 45*a*b^2*c^6*d^4*l*z - 36*a^2*b*c
^6*e^4*k*z - 9*a^3*b^5*c*d*l^4*z + 9*a*b^3*c^5*e^4*k*z - 72*a^3*c^6*d*e*h^3
*z - 72*a^2*c^7*d*e^3*h*z + 9*b^3*c^6*d^3*e*g*z + 72*a^2*c^7*d*e*f^3*z + 36
*a^3*b*c^5*d*h^4*z - 9*a*b^2*c^6*e^4*g*z + 36*a*b*c^7*d^3*f^2*z + 90*a^5*b^
2*c^2*j^3*m^2*z + 45*a^5*b^2*c^2*j^2*l^3*z + 9*a^4*b^3*c^2*j^2*k^3*z - 9*a^
4*b^3*c^2*h^3*m^2*z - 45*a^4*b^2*c^3*g^3*m^2*z + 9*a^3*b^4*c^2*g^3*m^2*z +
198*a^4*b^3*c^2*f^2*m^3*z - 108*a^3*b^3*c^3*f^3*m^2*z + 18*a^2*b^5*c^2*f^3*
m^2*z - 117*a^4*b^2*c^3*f^2*l^3*z + 117*a^3*b^2*c^4*e^3*m^2*z + 63*a^3*b^4*
c^2*f^2*l^3*z - 63*a^2*b^4*c^3*e^3*m^2*z - 171*a^2*b^3*c^4*d^3*m^2*z - 54*a
^3*b^3*c^3*f^2*k^3*z + 9*a^3*b^2*c^4*g^3*j^2*z + 9*a^2*b^5*c^2*f^2*k^3*z +
18*a^3*b^2*c^4*f^2*j^3*z + 18*a^2*b^3*c^4*f^3*j^2*z - 9*a^2*b^4*c^3*f^2*j^3
*z - 45*a^2*b^2*c^5*e^3*j^2*z + 9*a^2*b^3*c^4*f^2*h^3*z - 9*a^2*b^2*c^5*f^2
*g^3*z + 9*a*b^8*d*e*m^3*z - 36*a*b*c^7*d^4*h*z - 108*a^6*c^3*h^2*l*m^2*z +

```
 108*a^6*c^3*j*k^2*l^2*z - 108*a^6*c^3*g*k^2*m^2*z - 108*a^6*c^3*e*l^2*m^2*
z + 108*a^5*c^4*h^2*j^2*l*z + 108*a^5*c^4*e^2*l*m^2*z + 216*a^5*c^4*f^2*j*m
^2*z + 108*a^5*c^4*h^2*j*k^2*z + 108*a^5*c^4*g^2*j*l^2*z + 108*a^5*c^4*g*j^
2*k^2*z - 216*a^4*c^5*d^2*k^2*l*z + 108*a^5*c^4*e*j^2*l^2*z - 108*a^4*c^5*e
^2*j^2*l*z - 9*a^6*b^2*c*l^3*m^2*z + 108*a^5*c^4*e*h^2*m^2*z - 108*a^4*c^5*
f^2*h^2*l*z + 108*a^4*c^5*e^2*j*k^2*z + 108*a^4*c^5*d^2*j*l^2*z - 144*a^6*b
*c^2*j^2*m^3*z + 108*a^4*c^5*g^2*h^2*j*z - 27*a^4*b^4*c*j^3*m^2*z + 27*a^4*
b^3*c^2*j^4*m*z + 9*a^5*b^2*c^2*k^4*l*z + 216*a^4*c^5*e^2*g*l^2*z - 108*a^4
*c^5*f^2*g*k^2*z - 108*a^4*c^5*d^2*g*m^2*z - 9*a^4*b^4*c*j^2*l^3*z - 108*a^
4*c^5*e*h^2*j^2*z - 108*a^4*c^5*e*f^2*l^2*z + 108*a^3*c^6*e^2*f^2*l*z - 36*
a^5*b*c^3*j^2*k^3*z + 36*a^5*b*c^3*h^3*m^2*z + 108*a^3*c^6*e^2*g^2*j*z + 10
8*a^3*c^6*d^2*h^2*j*z - 216*a^5*b*c^3*f^2*m^3*z + 144*a^4*b*c^4*f^3*m^2*z +
 108*a^3*c^6*d^2*g*j^2*z - 72*a^3*b^5*c*f^2*m^3*z - 45*a^5*b^2*c^2*g*l^4*z
- 9*a^4*b^3*c^2*h*k^4*z - 9*a^3*b^2*c^4*g^4*l*z + 9*a^2*b^3*c^4*f^4*m*z + 2
16*a^3*c^6*d^2*e*k^2*z - 9*a^2*b^6*c*f^2*l^3*z + 9*a*b^6*c^2*e^3*m^2*z + 10
8*a^3*c^6*e*f^2*h^2*z + 108*a^3*b*c^5*d^3*m^2*z + 108*a^2*c^7*d^2*e^2*j*z +
 72*a^4*b*c^4*f^2*k^3*z + 72*a*b^5*c^3*d^3*m^2*z - 72*a^3*b*c^5*f^3*j^2*z +
 54*a^4*b^3*c^2*d*l^4*z - 45*a^4*b^2*c^3*e*k^4*z + 18*a^3*b^3*c^3*f*j^4*z +
 9*a^3*b^4*c^2*e*k^4*z - 9*a^2*b^2*c^5*f^4*j*z - 108*a^2*c^7*d^2*f^2*g*z +
9*a^3*b^2*c^4*g*h^4*z + 9*a*b^4*c^4*e^3*j^2*z - 72*a^2*b*c^6*d^3*j^2*z + 54
*a*b^3*c^5*d^3*j^2*z - 36*a^3*b*c^5*f^2*h^3*z - 9*a^2*b^3*c^4*d*h^4*z + 9*a
^2*b^2*c^5*e*g^4*z + 9*a*b^2*c^6*e^3*f^2*z + 36*a^7*c^2*l^3*m^2*z + 72*a^6*
c^3*j^3*m^2*z - 36*a^6*c^3*j^2*l^3*z + 9*a^4*b^5*j^2*m^3*z + 36*a^5*c^4*g^3
*m^2*z + 36*a^5*c^4*f^2*l^3*z - 36*a^4*c^5*e^3*m^2*z - 9*b^7*c^2*d^3*m^2*z
+ 9*a^2*b^7*f^2*m^3*z - 36*a^4*c^5*g^3*j^2*z + 72*a^4*c^5*f^2*j^3*z + 36*a^
3*c^6*e^3*j^2*z - 9*b^5*c^4*d^3*j^2*z + 36*a^3*c^6*f^2*g^3*z - 9*a^4*b^2*c^
3*j^5*z - 36*a^2*c^7*e^3*f^2*z - 9*b^3*c^6*d^3*f^2*z + 36*a^7*c^2*j*m^4*z -
 36*a^6*c^3*k^4*l*z - 18*a^5*b^4*j*m^4*z + 36*a^6*c^3*g*l^4*z + 36*a^4*c^5*
g^4*l*z + 18*a^4*b^5*f*m^4*z - 9*b^4*c^5*d^4*l*z + 36*a^5*c^4*e*k^4*z + 36*
a^3*c^6*f^4*j*z - 36*a^2*c^7*d^4*l*z - 36*a^4*c^5*g*h^4*z + 9*b^3*c^6*d^4*h
*z - 36*a^3*c^6*e*g^4*z + 36*a^2*c^7*e^4*g*z - 9*b^2*c^7*d^4*e*z - 36*a^7*b
*c*m^5*z + 36*a*c^8*d^4*e*z + 9*a^6*b^3*m^5*z + 36*a^5*c^4*j^5*z + 9*a^4*b^
3*c*g*h*j*k*l*m - 9*a^3*b^4*c*e*g*j*k*l*m - 9*a^3*b^4*c*d*h*j*k*l*m - 9*a^3
*b^4*c*f*g*h*k*l*m + 36*a^4*b*c^3*d*e*j*k*l*m + 9*a^2*b^5*c*d*e*j*k*l*m + 3
6*a^4*b*c^3*e*f*h*j*l*m + 36*a^4*b*c^3*e*f*g*k*l*m + 36*a^4*b*c^3*d*f*h*k*l
*m + 9*a^2*b^5*c*e*f*g*k*l*m + 9*a^2*b^5*c*d*f*h*k*l*m + 36*a^3*b*c^4*d*e*f
*j*k*l + 9*a*b^5*c^2*d*e*f*j*k*l + 36*a^3*b*c^4*d*e*g*h*k*l + 36*a^3*b*c^4*
d*e*f*h*k*m + 36*a^3*b*c^4*d*e*f*g*l*m + 9*a*b^5*c^2*d*e*f*h*k*m + 9*a*b^5*
c^2*d*e*f*g*l*m - 9*a*b^4*c^3*d*e*f*h*j*k - 9*a*b^4*c^3*d*e*f*g*j*l - 9*a*b
^4*c^3*d*e*f*g*h*m + 9*a*b^3*c^4*d*e*f*g*h*j - 9*a*b^6*c*d*e*f*k*l*m + 18*a
^4*b^2*c^2*e*g*j*k*l*m + 18*a^4*b^2*c^2*d*h*j*k*l*m + 18*a^4*b^2*c^2*f*g*h*
k*l*m - 36*a^3*b^3*c^2*d*e*j*k*l*m - 36*a^3*b^3*c^2*e*f*g*k*l*m - 36*a^3*b^
3*c^2*d*f*h*k*l*m + 9*a^3*b^3*c^2*f*g*h*j*k*l + 9*a^3*b^3*c^2*e*g*h*j*k*m +
 9*a^3*b^3*c^2*d*g*h*j*l*m - 108*a^3*b^2*c^3*d*e*f*k*l*m + 54*a^2*b^4*c^2*d
*e*f*k*l*m - 36*a^3*b^2*c^3*d*f*g*j*k*m + 18*a^3*b^2*c^3*e*f*g*j*k*l + 18*a
```

^3*b^2*c^3*d*f*h*j*k*l + 18*a^3*b^2*c^3*d*e*h*j*k*m + 18*a^3*b^2*c^3*d*e*g*
j*l*m - 9*a^2*b^4*c^2*e*f*g*j*k*l - 9*a^2*b^4*c^2*d*f*h*j*k*l - 9*a^2*b^4*c
^2*d*e*h*j*k*m - 9*a^2*b^4*c^2*d*e*g*j*l*m + 18*a^3*b^2*c^3*e*f*g*h*k*m + 1
8*a^3*b^2*c^3*d*f*g*h*l*m - 9*a^2*b^4*c^2*e*f*g*h*k*m - 9*a^2*b^4*c^2*d*f*g
*h*l*m - 36*a^2*b^3*c^3*d*e*f*j*k*l - 36*a^2*b^3*c^3*d*e*f*h*k*m - 36*a^2*b
^3*c^3*d*e*f*g*l*m + 9*a^2*b^3*c^3*e*f*g*h*j*k + 9*a^2*b^3*c^3*d*f*g*h*j*l
+ 9*a^2*b^3*c^3*d*e*g*h*j*m + 18*a^2*b^2*c^4*d*e*f*h*j*k + 18*a^2*b^2*c^4*d
*e*f*g*j*l + 18*a^2*b^2*c^4*d*e*f*g*h*m - 9*a^5*b^2*c*h*j*k^2*l*m - 9*a^5*b
^2*c*g*j*k*l^2*m + 27*a^5*b^2*c*f*j*k*l*m^2 - 9*a^4*b^3*c*f*j^2*k*l*m + 9*a
^3*b^4*c*f^2*j*k*l*m - 18*a^5*b*c^2*e*j*k^2*l*m - 9*a^5*b^2*c*g*h*k*l*m^2 +
 9*a^4*b^3*c*e*j*k^2*l*m - 18*a^5*b*c^2*f*h*k^2*l*m - 18*a^5*b*c^2*d*j*k*l^
2*m + 9*a^4*b^3*c*f*h*k^2*l*m + 9*a^4*b^3*c*d*j*k*l^2*m + 36*a^5*b*c^2*e*h*
k*l^2*m - 36*a^4*b*c^3*e^2*h*k*l*m + 18*a^5*b*c^2*f*h*j*l^2*m - 18*a^5*b*c^
2*f*g*k*l^2*m - 18*a^4*b^3*c*e*h*k*l^2*m + 9*a^4*b^3*c*f*g*k*l^2*m + 9*a^3*
b^4*c*e*h^2*k*l*m - 9*a^2*b^5*c*e^2*h*k*l*m - 54*a^5*b*c^2*e*h*j*l*m^2 - 18
*a^5*b*c^2*e*g*k*l*m^2 - 18*a^5*b*c^2*d*h*k*l*m^2 + 18*a^4*b^3*c*e*h*j*l*m^
2 - 9*a^4*b^3*c*f*h*j*k*m^2 - 9*a^4*b^3*c*f*g*j*l*m^2 + 9*a^4*b^3*c*e*g*k*l
*m^2 + 9*a^4*b^3*c*d*h*k*l*m^2 + 18*a^4*b*c^3*f*g^2*j*k*m - 18*a^4*b*c^3*e*
g^2*j*l*m + 18*a^3*b^4*c*d*g*k^2*l*m - 9*a^3*b^4*c*e*f*k^2*l*m - 9*a^2*b^5*
c*d*g^2*k*l*m - 18*a^4*b*c^3*f*g^2*h*l*m - 18*a^4*b*c^3*d*h^2*j*k*m - 9*a^3
*b^4*c*d*f*k*l^2*m - 54*a^4*b*c^3*d*g*j^2*k*m - 18*a^4*b*c^3*f*g*h^2*k*m -
18*a^4*b*c^3*e*g*j^2*k*l - 18*a^4*b*c^3*d*h*j^2*k*l - 18*a^3*b^4*c*d*g*j*k*
m^2 + 9*a^3*b^4*c*e*f*j*k*m^2 + 9*a^3*b^4*c*d*f*j*l*m^2 - 9*a^3*b^4*c*d*e*k
*l*m^2 - 54*a^3*b*c^4*d^2*f*j*k*m + 36*a^4*b*c^3*d*g*j*k^2*l - 36*a^3*b*c^4
*d^2*g*j*k*l - 18*a^4*b*c^3*e*f*j*k^2*l + 18*a^4*b*c^3*d*f*j*k^2*m - 18*a^3
*b*c^4*d^2*e*j*l*m + 9*a^3*b^4*c*f*g*h*j*m^2 - 9*a*b^5*c^2*d^2*g*j*k*l + 36
*a^4*b*c^3*d*g*h*k^2*m - 36*a^3*b*c^4*d^2*g*h*k*m + 18*a^4*b*c^3*e*g*h*k^2*
l - 18*a^4*b*c^3*e*f*h*k^2*m - 18*a^4*b*c^3*d*f*j*k*l^2 - 18*a^3*b*c^4*d^2*
f*h*l*m - 18*a^3*b*c^4*d*e^2*j*k*m - 9*a*b^5*c^2*d^2*g*h*k*m - 54*a^4*b*c^3
*d*g*h*k*l^2 - 54*a^3*b*c^4*e^2*f*h*j*m - 18*a^4*b*c^3*d*f*g*l^2*m - 18*a^3
*b*c^4*e^2*f*g*k*m - 54*a^4*b*c^3*d*f*g*k*m^2 - 36*a^4*b*c^3*e*f*g*j*m^2 -
36*a^4*b*c^3*d*f*h*j*m^2 + 36*a^3*b*c^4*e*f^2*g*j*m + 36*a^3*b*c^4*d*f^2*h*
j*m - 18*a^4*b*c^3*d*e*h*k*m^2 - 18*a^4*b*c^3*d*e*g*l*m^2 + 18*a^3*b*c^4*e*
f^2*h*j*l - 18*a^3*b*c^4*e*f^2*g*k*l - 18*a^3*b*c^4*d*f^2*h*k*l + 18*a^3*b*
c^4*d*f^2*g*k*m - 9*a^2*b^5*c*e*f*g*j*m^2 - 9*a^2*b^5*c*d*f*h*j*m^2 - 54*a^
3*b*c^4*d*f*g^2*j*m - 18*a^3*b*c^4*e*f*g^2*j*l - 18*a*b^4*c^3*d^2*f*g*j*m +
 9*a*b^4*c^3*d^2*g*h*j*k + 9*a*b^4*c^3*d^2*f*g*k*l + 9*a*b^4*c^3*d^2*e*g*k*
m - 9*a*b^4*c^3*d^2*e*f*l*m - 18*a^3*b*c^4*e*f*g^2*h*m - 18*a^3*b*c^4*d*f*h
^2*j*k - 9*a*b^4*c^3*d*e^2*f*k*m + 18*a^3*b*c^4*d*f*g*j^2*k - 18*a^3*b*c^4*
d*f*g*h^2*m - 18*a^3*b*c^4*d*e*h*j^2*k - 18*a^3*b*c^4*d*e*g*j^2*l + 18*a*b^
4*c^3*d*e*f^2*j*m - 9*a*b^5*c^2*d*e*f*j^2*m - 9*a*b^4*c^3*d*e*f^2*k*l - 18*
a^2*b*c^5*d^2*e*f*j*l - 9*a*b^3*c^4*d^2*e*g*j*k + 9*a*b^3*c^4*d^2*e*f*j*l -
 54*a^2*b*c^5*d^2*e*g*h*l - 18*a^2*b*c^5*d^2*e*f*h*m - 18*a^2*b*c^5*d*e^2*f
*j*k + 18*a*b^3*c^4*d^2*e*g*h*l - 9*a*b^3*c^4*d^2*f*g*h*k + 9*a*b^3*c^4*d^2
*e*f*h*m + 9*a*b^3*c^4*d*e^2*f*j*k - 36*a^3*b*c^4*d*e*f*h*l^2 + 36*a^2*b*c^

5*d*e^2*f*h*l + 18*a^2*b*c^5*d*e^2*g*h*k - 18*a^2*b*c^5*d*e^2*f*g*m - 18*a*
b^3*c^4*d*e^2*f*h*l - 9*a*b^5*c^2*d*e*f*h*l^2 + 9*a*b^4*c^3*d*e*f*h^2*l + 9
*a*b^3*c^4*d*e^2*f*g*m - 18*a^2*b*c^5*d*e*f^2*h*k - 18*a^2*b*c^5*d*e*f^2*g*
l + 9*a*b^3*c^4*d*e*f^2*h*k + 9*a*b^3*c^4*d*e*f^2*g*l + 27*a*b^2*c^5*d^2*e*
f*g*k + 9*a*b^4*c^3*d*e*f*g*k^2 - 9*a*b^3*c^4*d*e*f*g^2*k - 9*a*b^2*c^5*d^2
*e*f*h*j - 9*a*b^2*c^5*d*e^2*f*g*j - 9*a*b^2*c^5*d*e*f^2*g*h + 72*a^4*c^4*d
*f*g*j*k*m + 72*a^4*c^4*d*e*f*k*l*m + 9*a*b^6*c*d^2*g*k*l*m + 9*a*b^6*c*d*e
*f*j*m^2 - 27*a^4*b^2*c^2*f^2*j*k*l*m - 9*a^4*b^2*c^2*g^2*h*j*l*m + 36*a^3*
b^3*c^2*e^2*h*k*l*m - 18*a^4*b^2*c^2*e*h^2*k*l*m - 9*a^4*b^2*c^2*g*h^2*j*k*
m + 18*a^4*b^2*c^2*f*h*j^2*k*m + 18*a^4*b^2*c^2*f*g*j^2*l*m - 18*a^4*b^2*c^
2*e*h*j^2*l*m - 9*a^4*b^2*c^2*g*h*j^2*k*l - 9*a^3*b^3*c^2*f^2*h*j*k*m - 9*a
^3*b^3*c^2*f^2*g*j*l*m - 63*a^4*b^2*c^2*d*g*k^2*l*m + 63*a^3*b^2*c^3*d^2*g*
k*l*m - 45*a^2*b^4*c^2*d^2*g*k*l*m + 36*a^4*b^2*c^2*e*f*k^2*l*m + 27*a^3*b^
3*c^2*d*g^2*k*l*m - 9*a^4*b^2*c^2*f*h*j*k^2*l - 9*a^4*b^2*c^2*e*h*j*k^2*m +
9*a^3*b^3*c^2*e*g^2*j*l*m - 9*a^3*b^2*c^3*d^2*h*j*l*m + 36*a^4*b^2*c^2*d*f
*k*l^2*m + 27*a^4*b^2*c^2*e*h*j*k*l^2 - 27*a^3*b^2*c^3*e^2*h*j*k*l - 18*a^3
*b^2*c^3*e^2*f*j*l*m - 9*a^4*b^2*c^2*f*g*j*k*l^2 - 9*a^4*b^2*c^2*d*g*j*l^2*
m + 9*a^3*b^3*c^2*f*g^2*h*l*m - 9*a^3*b^3*c^2*e*h^2*j*k*l + 9*a^3*b^3*c^2*d
*h^2*j*k*m - 9*a^3*b^2*c^3*e^2*g*j*k*m + 9*a^2*b^4*c^2*e^2*h*j*k*l + 72*a^4
*b^2*c^2*d*g*j*k*m^2 + 36*a^4*b^2*c^2*d*e*k*l*m^2 + 27*a^4*b^2*c^2*e*g*h*l^
2*m - 27*a^4*b^2*c^2*e*f*j*k*m^2 - 27*a^4*b^2*c^2*d*f*j*l*m^2 - 27*a^3*b^2*
c^3*e^2*g*h*l*m + 27*a^3*b^2*c^3*e*f^2*j*k*m + 27*a^3*b^2*c^3*d*f^2*j*l*m +
18*a^3*b^3*c^2*d*g*j^2*k*m + 9*a^3*b^3*c^2*f*g*h^2*k*m + 9*a^3*b^3*c^2*e*g
*j^2*k*l - 9*a^3*b^3*c^2*e*g*h^2*l*m - 9*a^3*b^3*c^2*e*f*j^2*k*m + 9*a^3*b^
3*c^2*d*h*j^2*k*l - 9*a^3*b^3*c^2*d*f*j^2*l*m + 9*a^2*b^4*c^2*e^2*g*h*l*m +
36*a^2*b^3*c^3*d^2*g*j*k*l - 27*a^4*b^2*c^2*f*g*h*j*m^2 + 27*a^3*b^2*c^3*f
^2*g*h*j*m - 18*a^4*b^2*c^2*e*f*h*l*m^2 - 18*a^3*b^3*c^2*d*g*j*k^2*l - 18*a
^3*b^2*c^3*d*g^2*j*k*l + 18*a^2*b^3*c^3*d^2*f*j*k*m - 9*a^4*b^2*c^2*e*g*h*k
*m^2 - 9*a^4*b^2*c^2*d*g*h*l*m^2 - 9*a^3*b^3*c^2*f*g*h*j^2*m + 9*a^3*b^3*c^
2*e*f*j*k^2*l - 9*a^3*b^2*c^3*f^2*g*h*k*l + 9*a^2*b^4*c^2*d*g^2*j*k*l + 9*a
^2*b^3*c^3*d^2*e*j*l*m + 36*a^3*b^2*c^3*e*f*g^2*l*m + 36*a^2*b^3*c^3*d^2*g*
h*k*m - 18*a^3*b^3*c^2*d*g*h*k^2*m - 18*a^3*b^2*c^3*d*g^2*h*k*m + 9*a^3*b^3
*c^2*e*f*h*k^2*m + 9*a^3*b^3*c^2*d*f*j*k*l^2 - 9*a^3*b^2*c^3*f*g^2*h*j*l -
9*a^3*b^2*c^3*e*g^2*h*j*m - 9*a^2*b^4*c^2*e*f*g^2*l*m + 9*a^2*b^4*c^2*d*g^2
*h*k*m + 9*a^2*b^3*c^3*d^2*f*h*l*m + 9*a^2*b^3*c^3*d*e^2*j*k*m + 36*a^3*b^2
*c^3*d*f*h^2*k*m + 36*a^3*b^2*c^3*d*e*j^2*k*l + 18*a^3*b^3*c^2*d*g*h*k*l^2
+ 18*a^3*b^2*c^3*e*g*h^2*j*l + 18*a^3*b^2*c^3*e*f*h^2*k*l - 18*a^3*b^2*c^3*
e*f*h^2*j*m - 18*a^3*b^2*c^3*d*g*h^2*k*l + 18*a^3*b^2*c^3*d*e*h^2*l*m + 18*
a^2*b^3*c^3*e^2*f*h*j*m - 9*a^3*b^3*c^2*e*g*h*j*l^2 - 9*a^3*b^3*c^2*e*f*h*k
*l^2 + 9*a^3*b^3*c^2*d*f*g*l^2*m - 9*a^3*b^3*c^2*d*e*h*l^2*m - 9*a^3*b^2*c^
3*f*g*h^2*j*k - 9*a^3*b^2*c^3*d*g*h^2*j*m - 9*a^2*b^4*c^2*d*f*h^2*k*m - 9*a
^2*b^4*c^2*d*e*j^2*k*l - 9*a^2*b^3*c^3*e^2*g*h*j*l - 9*a^2*b^3*c^3*e^2*f*h*
k*l + 9*a^2*b^3*c^3*e^2*f*g*k*m - 9*a^2*b^3*c^3*d*e^2*h*l*m + 36*a^3*b^3*c^
2*e*f*g*j*m^2 + 36*a^3*b^3*c^2*d*f*h*j*m^2 + 18*a^3*b^3*c^2*d*f*g*k*m^2 - 1
8*a^3*b^2*c^3*e*f*g*j^2*m - 18*a^3*b^2*c^3*d*f*h*j^2*m - 18*a^2*b^3*c^3*e*f

^2*g*j*m - 18*a^2*b^3*c^3*d*f^2*h*j*m + 9*a^3*b^3*c^2*d*e*h*k*m^2 + 9*a^3*b
^3*c^2*d*e*g*l*m^2 - 9*a^3*b^2*c^3*e*g*h*j^2*k - 9*a^3*b^2*c^3*d*g*h*j^2*l
+ 9*a^2*b^4*c^2*e*f*g*j^2*m + 9*a^2*b^4*c^2*d*f*h*j^2*m + 9*a^2*b^3*c^3*e*f
^2*g*k*l + 9*a^2*b^3*c^3*d*f^2*h*k*l + 72*a^2*b^2*c^4*d^2*f*g*j*m + 36*a^2*
b^2*c^4*d^2*e*f*l*m + 27*a^3*b^2*c^3*d*g*h*j*k^2 + 27*a^3*b^2*c^3*d*f*g*k^2
*l + 27*a^3*b^2*c^3*d*e*g*k^2*m - 27*a^2*b^2*c^4*d^2*g*h*j*k - 27*a^2*b^2*c
^4*d^2*f*g*k*l - 27*a^2*b^2*c^4*d^2*e*g*k*m + 18*a^2*b^3*c^3*d*f*g^2*j*m -
18*a^2*b^2*c^4*d^2*e*h*k*l - 9*a^3*b^2*c^3*e*f*h*j*k^2 + 9*a^2*b^3*c^3*e*f*
g^2*j*l - 9*a^2*b^3*c^3*d*g^2*h*j*k - 9*a^2*b^3*c^3*d*f*g^2*k*l - 9*a^2*b^3
*c^3*d*e*g^2*k*m - 9*a^2*b^2*c^4*d^2*f*h*j*l - 9*a^2*b^2*c^4*d^2*e*h*j*m +
36*a^2*b^2*c^4*d*e^2*f*k*m - 27*a^3*b^2*c^3*d*e*h*j*l^2 + 27*a^2*b^2*c^4*d*
e^2*h*j*l - 18*a^3*b^2*c^3*d*e*g*k*l^2 - 9*a^3*b^2*c^3*d*f*g*j*l^2 + 9*a^2*
b^4*c^2*d*e*h*j*l^2 + 9*a^2*b^3*c^3*e*f*g^2*h*m + 9*a^2*b^3*c^3*d*f*h^2*j*k
 - 9*a^2*b^3*c^3*d*e*h^2*j*l - 9*a^2*b^2*c^4*e^2*f*g*j*k - 9*a^2*b^2*c^4*d*
e^2*g*j*m + 63*a^3*b^2*c^3*d*e*f*j*m^2 - 63*a^2*b^2*c^4*d*e*f^2*j*m - 45*a^
2*b^4*c^2*d*e*f*j*m^2 + 36*a^2*b^2*c^4*d*e*f^2*k*l - 27*a^3*b^2*c^3*e*f*g*h
*l^2 + 27*a^2*b^3*c^3*d*e*f*j^2*m + 27*a^2*b^2*c^4*e^2*f*g*h*l + 9*a^2*b^4*
c^2*e*f*g*h*l^2 - 9*a^2*b^3*c^3*e*f*g*h^2*l + 9*a^2*b^3*c^3*d*f*g*h^2*m + 9
*a^2*b^3*c^3*d*e*h*j^2*k + 9*a^2*b^3*c^3*d*e*g*j^2*l + 18*a^2*b^2*c^4*d*e*g
^2*j*k - 9*a^3*b^2*c^3*d*e*g*h*m^2 - 9*a^2*b^3*c^3*d*e*g*j*k^2 - 9*a^2*b^2*
c^4*e*f^2*g*h*k - 9*a^2*b^2*c^4*d*f^2*g*h*l + 18*a^2*b^2*c^4*d*f*g^2*h*k -
18*a^2*b^2*c^4*d*e*g^2*h*l - 9*a^2*b^3*c^3*d*f*g*h*k^2 - 9*a^2*b^2*c^4*e*f*
g^2*h*j + 36*a^2*b^3*c^3*d*e*f*h*l^2 - 18*a^2*b^2*c^4*d*e*f*h^2*l - 9*a^2*b
^2*c^4*d*f*g*h^2*j - 9*a^2*b^2*c^4*d*e*g*h*j^2 - 27*a^2*b^2*c^4*d*e*f*g*k^2
 + 18*a^2*b^2*c^4*d^2*f*h*k^2 - 9*a^2*b^3*c^3*e*f*g^2*k^2 - 9*a^2*b^2*c^4*e
^2*f*h*j^2 - 9*a^2*b^2*c^4*d*f^2*h^2*k + 45*a^2*b^3*c^3*d*e*f^2*m^2 + 36*a^
2*b^2*c^4*d^2*e*g*l^2 + 9*a^2*b^3*c^3*d*e*g^2*l^2 + 9*a^2*b^2*c^4*e*f^2*g*j
^2 + 9*a^2*b^2*c^4*d*f^2*h*j^2 - 9*a^2*b^2*c^4*d*e^2*h*k^2 - 36*a^2*b^2*c^4
*d*e^2*f*l^2 - 9*a^2*b^2*c^4*d*f*g^2*j^2 - 12*a^6*b*c*h*k*l^3*m + 3*a*b^6*c
*e^3*k*l*m + 3*a*b^6*c*d*e*f*l^3 - 12*a*b*c^6*d*e^3*f*h + 9*a^5*b^2*c*h^2*k
*l^2*m + 18*a^5*b*c^2*g^2*k^2*l*m - 9*a^5*b^2*c*h^2*j*l*m^2 + 9*a^5*b*c^2*h
^2*j^2*l*m - 9*a^4*b^3*c*g^2*k^2*l*m - 3*a^4*b^2*c^2*g^3*k*l*m + 18*a^5*b*c
^2*f^2*k*l*m^2 + 15*a^3*b^3*c^2*f^3*k*l*m + 9*a^5*b^2*c*h*j^2*k*m^2 + 9*a^5
*b^2*c*g*j^2*l*m^2 - 9*a^5*b^2*c*f*k^2*l^2*m + 9*a^5*b*c^2*h^2*j*k^2*m + 9*
a^5*b*c^2*g^2*j*l^2*m - 9*a^4*b^3*c*f^2*k*l*m^2 + 36*a^3*b^2*c^3*e^3*k*l*m
- 27*a^5*b*c^2*g^2*j*k*m^2 - 18*a^5*b*c^2*h^2*j*k*l^2 - 18*a^2*b^4*c^2*e^3*
k*l*m - 9*a^5*b^2*c*g*j*k^2*m^2 - 9*a^5*b^2*c*e*k^2*l*m^2 + 9*a^5*b*c^2*h*j
^2*k^2*l + 9*a^5*b*c^2*g*j^2*k^2*m + 9*a^4*b^3*c*g^2*j*k*m^2 + 9*a^3*b^4*c*
e^2*k*l^2*m + 3*a^4*b^2*c^2*h^3*j*k*l - 54*a^4*b*c^3*d^2*k^2*l*m - 51*a^2*b
^3*c^3*d^3*k*l*m - 27*a^4*b*c^3*e^2*j^2*l*m - 18*a^5*b*c^2*g*h^2*l^2*m - 9*
a^5*b^2*c*e*j*l^2*m^2 - 9*a^5*b^2*c*d*k*l^2*m^2 + 9*a^5*b*c^2*g^2*h*l*m^2 +
 9*a^5*b*c^2*g*j^2*k*l^2 + 9*a^5*b*c^2*e*j^2*l^2*m - 9*a^3*b^4*c*e^2*j*l*m^
2 - 9*a^2*b^5*c*d^2*k^2*l*m + 3*a^4*b^2*c^2*g*h^3*l*m - 3*a^3*b^3*c^2*g^3*j
*k*l + 18*a^5*b*c^2*e*j^2*k*m^2 + 18*a^5*b*c^2*d*j^2*l*m^2 + 18*a^4*b*c^3*f
^2*j^2*k*l + 9*a^5*b*c^2*g*h^2*k*m^2 + 9*a^5*b*c^2*f*h^2*l*m^2 + 9*a^5*b*c^

2*f*j*k^2*l^2 - 9*a^4*b^3*c*e*j^2*k*m^2 - 9*a^4*b^3*c*d*j^2*l*m^2 + 9*a^4*b
^2*c^2*f*j^3*k*l + 9*a^4*b^2*c^2*e*j^3*k*m + 9*a^4*b^2*c^2*d*j^3*l*m + 9*a^
4*b*c^3*f^2*h^2*l*m + 9*a^4*b*c^3*e^2*j*k^2*m + 9*a^4*b*c^3*d^2*j*l^2*m - 3
*a^3*b^3*c^2*g^3*h*k*m - 3*a^3*b^2*c^3*f^3*j*k*l + 3*a^2*b^4*c^2*f^3*j*k*l
+ 45*a^4*b*c^3*d^2*j*k*m^2 - 27*a^5*b*c^2*d*j*k^2*m^2 + 18*a^5*b*c^2*g*h*j^
2*m^2 + 18*a^4*b*c^3*e^2*j*k*l^2 + 15*a^2*b^3*c^3*e^3*j*k*l - 12*a^3*b^2*c^
3*f^3*h*k*m - 12*a^3*b^2*c^3*f^3*g*l*m + 9*a^5*b*c^2*g*h*k^2*l^2 - 9*a^4*b^
3*c*g*h*j^2*m^2 + 9*a^4*b^3*c*d*j*k^2*m^2 + 9*a^4*b^2*c^2*g*h*j^3*m + 9*a^4
*b*c^3*g^2*h^2*k*l + 9*a^4*b*c^3*g^2*h^2*j*m + 9*a^2*b^5*c*d^2*j*k*m^2 + 3*
a^2*b^4*c^2*f^3*h*k*m + 3*a^2*b^4*c^2*f^3*g*l*m + 36*a^2*b^2*c^4*d^3*j*k*l
+ 18*a^4*b*c^3*e^2*g*l^2*m + 15*a^2*b^3*c^3*e^3*g*l*m + 12*a^4*b^2*c^2*d*j*
k^3*l + 9*a^5*b*c^2*f*g*k^2*m^2 + 9*a^5*b*c^2*e*h*k^2*m^2 + 9*a^4*b*c^3*g^2
*h*j^2*l + 9*a^4*b*c^3*f^2*h*k^2*l + 9*a^4*b*c^3*f^2*g*k^2*m + 9*a^4*b*c^3*
d^2*h*l*m^2 - 9*a^3*b^3*c^2*e*h^3*k*m + 6*a^2*b^3*c^3*e^3*h*k*m + 45*a^4*b*
c^3*e^2*h*j*m^2 + 36*a^2*b^2*c^4*d^3*h*k*m - 33*a^3*b^2*c^3*d*g^3*l*m - 27*
a^4*b*c^3*f^2*h*j*l^2 - 27*a^4*b*c^3*e^2*f*l*m^2 - 27*a^4*b*c^3*e*h^2*j^2*m
 - 18*a^4*b*c^3*g^2*h*j*k^2 - 18*a^4*b*c^3*f*g^2*k^2*l - 18*a^4*b*c^3*e*g^2
*k^2*m - 18*a^3*b*c^4*d^2*g^2*l*m + 12*a^4*b^2*c^2*d*h*k^3*m + 9*a^5*b*c^2*
e*f*l^2*m^2 + 9*a^5*b*c^2*d*g*l^2*m^2 + 9*a^4*b*c^3*f^2*g*k*l^2 + 9*a^4*b*c
^3*e^2*g*k*m^2 + 9*a^4*b*c^3*g*h^2*j^2*k + 9*a^4*b*c^3*f*h^2*j^2*l + 9*a^4*
b*c^3*e*f^2*l^2*m - 9*a^3*b^4*c*e*h^2*j*m^2 + 9*a^3*b*c^4*e^2*f^2*l*m + 9*a
^2*b^5*c*e^2*h*j*m^2 + 9*a^2*b^4*c^2*d*g^3*l*m - 9*a^2*b^2*c^4*d^3*g*l*m -
9*a*b^5*c^2*d^2*g^2*l*m - 6*a^4*b^2*c^2*e*h*k^3*l - 6*a^3*b^2*c^3*f*g^3*j*m
 + 3*a^4*b^2*c^2*g*h*j*k^3 + 3*a^4*b^2*c^2*f*g*k^3*l + 3*a^4*b^2*c^2*e*g*k^
3*m + 3*a^3*b^2*c^3*g^3*h*j*k + 3*a^3*b^2*c^3*f*g^3*k*l + 3*a^3*b^2*c^3*e*g
^3*k*m - 27*a^3*b*c^4*d^2*h^2*k*l + 18*a^4*b*c^3*e*f^2*k*m^2 + 18*a^4*b*c^3
*d*f^2*l*m^2 + 9*a^4*b*c^3*f*h^2*j*k^2 + 9*a^4*b*c^3*f*g^2*j*l^2 + 9*a^4*b*
c^3*e*g^2*k*l^2 + 9*a^4*b*c^3*d*h^2*k^2*l + 9*a^3*b^4*c*e*g*j^2*m^2 + 9*a^3
*b^4*c*d*h*j^2*m^2 - 9*a^3*b^3*c^2*e*g*j^3*m - 9*a^3*b^3*c^2*d*h*j^3*m + 9*
a^3*b*c^4*e^2*g^2*k*l + 9*a^3*b*c^4*e^2*g^2*j*m + 9*a^3*b*c^4*d^2*h^2*j*m -
 3*a^2*b^3*c^3*f^3*h*j*k - 3*a^2*b^3*c^3*f^3*g*j*l - 3*a^2*b^3*c^3*e*f^3*k*
m - 3*a^2*b^3*c^3*d*f^3*l*m + 45*a^4*b*c^3*d*g^2*j*m^2 + 45*a^3*b*c^4*d^2*g
*j^2*m + 24*a^4*b^2*c^2*d*g*k*l^3 + 24*a^2*b^2*c^4*e^3*f*j*m + 18*a^4*b*c^3
*f^2*g*h*m^2 + 18*a^4*b*c^3*d*h^2*j*l^2 + 18*a^3*b*c^4*e^2*h^2*j*k - 12*a^4
*b^2*c^2*e*g*j*l^3 - 12*a^4*b^2*c^2*e*f*k*l^3 - 12*a^4*b^2*c^2*d*e*l^3*m -
12*a^2*b^2*c^4*e^3*g*j*l - 12*a^2*b^2*c^4*e^3*f*k*l - 12*a^2*b^2*c^4*d*e^3*
l*m + 9*a^4*b*c^3*f*g*j^2*k^2 + 9*a^4*b*c^3*e*h*j^2*k^2 + 9*a^3*b^2*c^3*e*h
^3*j*k + 9*a^3*b^2*c^3*d*h^3*j*l + 9*a^3*b*c^4*f^2*g^2*j*k + 9*a^3*b*c^4*d^
2*h*j^2*l + 9*a^2*b^5*c*d*g^2*j*m^2 + 9*a*b^5*c^2*d^2*g*j^2*m - 3*a^4*b^2*c
^2*d*h*j*l^3 - 3*a^2*b^3*c^3*f^3*g*h*m - 3*a^2*b^2*c^4*e^3*h*j*k + 18*a^4*b
*c^3*f*g*h^2*l^2 + 18*a^3*b*c^4*e^2*g*h^2*m + 18*a^3*b*c^4*d^2*h*j*k^2 + 18
*a^3*b*c^4*d^2*f*k^2*l + 18*a^3*b*c^4*d^2*e*k^2*m + 9*a^4*b*c^3*e*g^2*h*m^2
 + 9*a^4*b*c^3*e*f*j^2*l^2 + 9*a^4*b*c^3*d*g*j^2*l^2 + 9*a^3*b^2*c^3*f*g*h^
3*l + 9*a^3*b^2*c^3*e*g*h^3*m + 9*a^3*b*c^4*f^2*g^2*h*l + 9*a^3*b*c^4*e^2*g
*j^2*k + 9*a^3*b*c^4*e^2*f*j^2*l - 9*a^2*b^3*c^3*d*g^3*j*l + 9*a*b^4*c^3*d^

2*g^2*j*l - 3*a^4*b^2*c^2*f*g*h*l^3 - 3*a^3*b^3*c^2*e*g*j*k^3 - 3*a^3*b^3*c
^2*d*h*j*k^3 - 3*a^3*b^3*c^2*d*f*k^3*l - 3*a^3*b^3*c^2*d*e*k^3*m - 3*a^2*b^
2*c^4*e^3*g*h*m - 33*a^3*b^2*c^3*d*e*j^3*m - 27*a^4*b*c^3*e*f*h^2*m^2 - 27*
a^3*b*c^4*d^2*e*k*l^2 - 18*a^4*b*c^3*d*e*j^2*m^2 - 18*a^3*b*c^4*e*f^2*j^2*k
 - 18*a^3*b*c^4*d*f^2*j^2*l - 9*a^4*b^2*c^2*d*e*j*m^3 + 9*a^4*b*c^3*d*g*h^2
*m^2 + 9*a^4*b*c^3*d*e*k^2*l^2 + 9*a^3*b*c^4*f^2*g*h^2*k + 9*a^3*b*c^4*e^2*
f*j*k^2 + 9*a^3*b*c^4*d^2*f*j*l^2 + 9*a^3*b*c^4*e*f^2*h^2*m + 9*a^3*b*c^4*d
*e^2*k^2*l - 9*a^2*b^5*c*d*e*j^2*m^2 + 9*a^2*b^4*c^2*d*e*j^3*m - 9*a^2*b^3*
c^3*d*g^3*h*m + 9*a^2*b*c^5*d^2*e^2*k*l + 9*a^2*b*c^5*d^2*e^2*j*m + 9*a*b^4
*c^3*d^2*g^2*h*m - 6*a^3*b^2*c^3*d*g*j^3*k - 3*a^3*b^3*c^2*f*g*h*k^3 + 3*a^
3*b^2*c^3*e*f*j^3*k + 3*a^3*b^2*c^3*d*f*j^3*l + 3*a^2*b^2*c^4*e*f^3*j*k + 3
*a^2*b^2*c^4*d*f^3*j*l + 45*a^3*b*c^4*d^2*g*h*l^2 + 36*a^4*b^2*c^2*e*f*g*m^
3 + 36*a^4*b^2*c^2*d*f*h*m^3 - 27*a^3*b*c^4*e^2*g*h*k^2 - 27*a^3*b*c^4*d*g^
2*h^2*l - 18*a^3*b*c^4*f^2*g*h*j^2 + 18*a^3*b*c^4*d*e^2*j*l^2 + 15*a^3*b^3*
c^2*d*e*j*l^3 + 12*a^2*b^2*c^4*e*f^3*g*m + 12*a^2*b^2*c^4*d*f^3*h*m + 9*a^3
*b*c^4*f*g^2*h^2*j + 9*a^3*b*c^4*e*g^2*h^2*k + 9*a^3*b*c^4*d*f^2*j*k^2 + 9*
a^2*b*c^5*d^2*f^2*j*k + 9*a*b^5*c^2*d^2*g*h*l^2 - 9*a*b^4*c^3*d^2*g*h^2*l -
 6*a^2*b^2*c^4*e*f^3*h*l + 3*a^3*b^2*c^3*f*g*h*j^3 + 3*a^2*b^2*c^4*f^3*g*h*
j + 45*a^3*b*c^4*d^2*f*g*m^2 - 27*a^2*b*c^5*d^2*f^2*g*m + 18*a^3*b*c^4*e^2*
f*g*l^2 + 15*a^3*b^3*c^2*e*f*g*l^3 - 12*a^3*b^2*c^3*d*e*j*k^3 + 9*a^3*b*c^4
*d^2*e*h*m^2 + 9*a^3*b*c^4*e*g^2*h*j^2 + 9*a^3*b*c^4*e*f^2*h*k^2 - 9*a^2*b^
3*c^3*d*f*h^3*l + 9*a^2*b*c^5*d^2*f^2*h*l + 9*a*b^5*c^2*d^2*f*g*m^2 + 9*a*b
^3*c^4*d^2*f^2*g*m + 6*a^3*b^3*c^2*d*f*h*l^3 + 3*a^2*b^4*c^2*d*e*j*k^3 + 18
*a^3*b*c^4*e*f*g^2*k^2 + 18*a^2*b*c^5*d^2*g^2*h*j + 18*a^2*b*c^5*d^2*f*g^2*
l + 18*a^2*b*c^5*d^2*e*g^2*m - 12*a^3*b^2*c^3*d*f*h*k^3 + 9*a^3*b*c^4*e*f*h
^2*j^2 + 9*a^3*b*c^4*d*f^2*g*l^2 + 9*a^3*b*c^4*d*e^2*g*m^2 + 9*a^3*b*c^4*d*
g*h^2*j^2 + 9*a^2*b^2*c^4*e*f*g^3*k + 9*a^2*b^2*c^4*d*g^3*h*j + 9*a^2*b^2*c
^4*d*f*g^3*l + 9*a^2*b^2*c^4*d*e*g^3*m + 9*a^2*b*c^5*e^2*f^2*h*j + 9*a^2*b*
c^5*e^2*f^2*g*k - 9*a*b^3*c^4*d^2*g^2*h*j - 9*a*b^3*c^4*d^2*f*g^2*l - 9*a*b
^3*c^4*d^2*e*g^2*m - 3*a^3*b^2*c^3*e*f*g*k^3 + 3*a^2*b^4*c^2*e*f*g*k^3 + 3*
a^2*b^4*c^2*d*f*h*k^3 - 54*a^3*b*c^4*d*e*f^2*m^2 - 51*a^3*b^3*c^2*d*e*f*m^3
 - 27*a^3*b*c^4*d*e*g^2*l^2 + 9*a^3*b*c^4*d*e*h^2*k^2 + 9*a^2*b*c^5*e^2*f*g
^2*j + 9*a^2*b*c^5*d^2*f*h^2*j + 9*a^2*b*c^5*d^2*e*h^2*k + 9*a^2*b*c^5*d*e^
2*g^2*l - 9*a*b^5*c^2*d*e*f^2*m^2 - 9*a*b^4*c^3*d^2*e*g*l^2 - 9*a*b^2*c^5*d
^2*e^2*g*l - 9*a*b^2*c^5*d^2*e^2*f*m - 3*a^2*b^3*c^3*e*f*g*j^3 - 3*a^2*b^3*
c^3*d*f*h*j^3 + 36*a^3*b^2*c^3*d*e*f*l^3 - 27*a^2*b*c^5*d^2*f*g*j^2 - 18*a^
2*b^4*c^2*d*e*f*l^3 - 18*a^2*b*c^5*d*e^2*h^2*j + 9*a^2*b*c^5*d^2*e*h*j^2 +
9*a^2*b*c^5*d*f^2*g^2*j + 9*a*b^4*c^3*d*e^2*f*l^2 + 9*a*b^3*c^4*d^2*f*g*j^2
 - 9*a*b^2*c^5*d^2*f^2*g*j - 9*a*b^2*c^5*d^2*e*f^2*l + 3*a^2*b^2*c^4*d*e*h^
3*j - 18*a^2*b*c^5*e^2*f*g*h^2 + 18*a^2*b*c^5*d^2*e*f*k^2 + 15*a^2*b^3*c^3*
d*e*f*k^3 + 9*a^2*b*c^5*e*f^2*g^2*h + 9*a^2*b*c^5*d*e^2*g*j^2 - 9*a*b^3*c^4
*d^2*e*f*k^2 + 9*a*b^2*c^5*d^2*e*g^2*j - 9*a*b^2*c^5*d*e^2*f^2*k + 3*a^2*b^
2*c^4*e*f*g*h^3 + 18*a^2*b*c^5*d*e*f^2*j^2 + 9*a^2*b*c^5*d*f^2*g*h^2 - 9*a*
b^3*c^4*d*e*f^2*j^2 + 9*a*b^2*c^5*d^2*f*g^2*h - 3*a^2*b^2*c^4*d*e*f*j^3 + 9
*a^2*b*c^5*d*e*g^2*h^2 - 9*a*b^2*c^5*d^2*e*g*h^2 + 9*a*b^2*c^5*d*e^2*f*h^2

- 36*a^6*c^2*f*j*k*l*m^2 + 36*a^5*c^3*f^2*j*k*l*m - 36*a^5*c^3*f*h^2*j*l*m
+ 36*a^5*c^3*e*h*j^2*l*m - 18*a^6*b*c*j^2*k*l*m^2 + 9*a^6*b*c*j*k^2*l^2*m +
3*a^5*b^2*c*j^3*k*l*m - 36*a^5*c^3*f*g*j*k^2*m - 36*a^5*c^3*e*f*k^2*l*m +
36*a^5*c^3*d*g*k^2*l*m - 36*a^4*c^4*d^2*g*k*l*m - 36*a^5*c^3*e*h*j*k*l^2 -
36*a^5*c^3*e*f*j*l^2*m - 36*a^5*c^3*d*f*k*l^2*m + 36*a^4*c^4*e^2*h*j*k*l +
36*a^4*c^4*e^2*f*j*l*m + 9*a^6*b*c*h*k^2*l*m^2 - 3*a^4*b^3*c*h^3*k*l*m - 36
*a^5*c^3*e*g*h*l^2*m + 36*a^5*c^3*e*f*j*k*m^2 - 36*a^5*c^3*d*g*j*k*m^2 + 36
*a^5*c^3*d*f*j*l*m^2 - 36*a^5*c^3*d*e*k*l*m^2 + 36*a^4*c^4*e^2*g*h*l*m - 36
*a^4*c^4*e*f^2*j*k*m - 36*a^4*c^4*d*f^2*j*l*m + 9*a^6*b*c*h*j*l^2*m^2 + 9*a
^6*b*c*g*k*l^2*m^2 + 9*a^5*b^2*c*g*k^3*l*m + 3*a^3*b^4*c*g^3*k*l*m + 36*a^5
*c^3*f*g*h*j*m^2 + 36*a^5*c^3*e*f*h*l*m^2 - 36*a^4*c^4*f^2*g*h*j*m - 36*a^4
*c^4*e*f^2*h*l*m - 24*a^4*b*c^3*f^3*k*l*m - 12*a^5*b*c^2*h*j^3*k*m - 12*a^5
*b*c^2*g*j^3*l*m - 3*a^2*b^5*c*f^3*k*l*m - 36*a^4*c^4*e*g^2*h*k*l - 36*a^4*
c^4*e*f*g^2*l*m + 12*a^5*b^2*c*e*k*l^3*m - 6*a^5*b^2*c*f*j*l^3*m + 3*a^5*b^
2*c*h*j*k*l^3 + 48*a^3*b*c^4*d^3*k*l*m + 36*a^4*c^4*e*f*h^2*j*m + 36*a^4*c^
4*d*g*h^2*k*l - 36*a^4*c^4*d*f*h^2*k*m - 36*a^4*c^4*d*e*j^2*k*l + 24*a^5*b*
c^2*d*k^3*l*m + 21*a*b^5*c^2*d^3*k*l*m - 12*a^5*b*c^2*g*j*k^3*l - 9*a^4*b^3
*c*d*k^3*l*m + 6*a^5*b*c^2*f*j*k^3*m + 3*a^5*b^2*c*g*h*l^3*m - 36*a^4*c^4*e
*f*h*j^2*l - 12*a^5*b*c^2*g*h*k^3*m - 3*a^5*b^2*c*e*j*k*m^3 - 3*a^5*b^2*c*d
*j*l*m^3 - 36*a^4*c^4*d*g*h*j*k^2 - 36*a^4*c^4*d*f*g*k^2*l - 36*a^4*c^4*d*e
*h*k^2*l - 36*a^4*c^4*d*e*g*k^2*m + 36*a^3*c^5*d^2*g*h*j*k + 36*a^3*c^5*d^2
*f*g*k*l - 36*a^3*c^5*d^2*f*g*j*m + 36*a^3*c^5*d^2*e*h*k*l + 36*a^3*c^5*d^2
*e*g*k*m - 36*a^3*c^5*d^2*e*f*l*m + 24*a^5*b^2*c*e*h*l*m^3 - 24*a^3*b*c^4*e
^3*j*k*l - 12*a^5*b^2*c*f*h*k*m^3 - 12*a^5*b^2*c*f*g*l*m^3 - 3*a^5*b^2*c*g*
h*j*m^3 - 3*a^4*b^3*c*e*j*k*l^3 - 3*a*b^5*c^2*e^3*j*k*l + 36*a^4*c^4*d*e*h*
j*l^2 + 36*a^4*c^4*d*e*g*k*l^2 - 36*a^3*c^5*d*e^2*h*j*l - 36*a^3*c^5*d*e^2*
g*k*l - 36*a^3*c^5*d*e^2*f*k*m + 24*a^4*b*c^3*e*h^3*k*m - 24*a^3*b*c^4*e^3*
g*l*m - 18*a*b^4*c^3*d^3*j*k*l - 12*a^4*b*c^3*g*h^3*j*l - 12*a^4*b*c^3*f*h^
3*k*l - 12*a^4*b*c^3*d*h^3*l*m + 12*a^3*b*c^4*e^3*h*k*m + 6*a^4*b*c^3*f*h^3
*j*m - 3*a^4*b^3*c*g*h*j*l^3 - 3*a^4*b^3*c*f*h*k*l^3 - 3*a^4*b^3*c*e*g*l^3*
m - 3*a^4*b^3*c*d*h*l^3*m - 3*a*b^5*c^2*e^3*h*k*m - 3*a*b^5*c^2*e^3*g*l*m +
36*a^4*c^4*e*f*g*h*l^2 - 36*a^4*c^4*d*e*f*j*m^2 - 36*a^3*c^5*e^2*f*g*h*l -
36*a^3*c^5*d*f^2*g*j*k - 36*a^3*c^5*d*e*f^2*k*l + 36*a^3*c^5*d*e*f^2*j*m -
18*a*b^4*c^3*d^3*h*k*m - 9*a*b^4*c^3*d^3*g*l*m + 30*a^5*b*c^2*d*g*k*m^3 -
30*a^4*b^3*c*d*g*k*m^3 - 24*a^5*b*c^2*e*f*k*m^3 - 24*a^5*b*c^2*d*f*l*m^3 +
24*a^4*b*c^3*e*g*j^3*m + 24*a^4*b*c^3*d*h*j^3*m + 15*a^4*b^3*c*e*f*k*m^3 +
15*a^4*b^3*c*d*f*l*m^3 + 12*a^5*b*c^2*e*g*j*m^3 + 12*a^5*b*c^2*d*h*j*m^3 -
12*a^4*b*c^3*f*h*j^3*k - 12*a^4*b*c^3*f*g*j^3*l + 6*a^4*b^3*c*e*g*j*m^3 + 6
*a^4*b^3*c*d*h*j*m^3 + 6*a^4*b*c^3*e*h*j^3*l + 36*a^3*c^5*d*e*g^2*h*l - 24*
a^5*b*c^2*f*g*h*m^3 + 15*a^4*b^3*c*f*g*h*m^3 - 9*a*b^6*c*d^2*g*j*m^2 - 6*a^
3*b^4*c*d*g*k*l^3 - 6*a*b^4*c^3*e^3*f*j*m + 3*a^3*b^4*c*e*g*j*l^3 + 3*a^3*b
^4*c*e*f*k*l^3 + 3*a^3*b^4*c*d*h*j*l^3 + 3*a^3*b^4*c*d*e*l^3*m + 3*a*b^4*c^
3*e^3*h*j*k + 3*a*b^4*c^3*e^3*g*j*l + 3*a*b^4*c^3*e^3*f*k*l + 3*a*b^4*c^3*d
*e^3*l*m - 36*a^3*c^5*d*e*g*h^2*k + 30*a^2*b*c^5*d^3*f*j*m - 30*a*b^3*c^4*d
^3*f*j*m + 24*a^3*b*c^4*d*g^3*j*l - 24*a^2*b*c^5*d^3*h*j*k - 24*a^2*b*c^5*d

^3*f*k*l - 24*a^2*b*c^5*d^3*e*k*m + 15*a*b^3*c^4*d^3*h*j*k + 15*a*b^3*c^4*d
^3*f*k*l + 15*a*b^3*c^4*d^3*e*k*m - 12*a^3*b*c^4*e*g^3*j*k + 12*a^2*b*c^5*d
^3*g*j*l + 6*a*b^3*c^4*d^3*g*j*l + 3*a^3*b^4*c*f*g*h*l^3 + 3*a*b^4*c^3*e^3*
g*h*m + 24*a^3*b*c^4*d*g^3*h*m - 12*a^3*b*c^4*f*g^3*h*k + 12*a^2*b*c^5*d^3*
g*h*m - 9*a^3*b^4*c*d*e*j*m^3 + 6*a^3*b*c^4*e*g^3*h*l + 6*a*b^3*c^4*d^3*g*h
*m + 36*a^3*c^5*d*e*f*g*k^2 - 36*a^2*c^6*d^2*e*f*g*k - 24*a^4*b*c^3*d*e*j*l
^3 - 18*a^3*b^4*c*e*f*g*m^3 - 18*a^3*b^4*c*d*f*h*m^3 - 3*a^2*b^5*c*d*e*j*l^
3 - 3*a*b^3*c^4*d*e^3*j*l - 24*a^4*b*c^3*e*f*g*l^3 + 24*a^3*b*c^4*d*f*h^3*l
 + 12*a^4*b*c^3*d*f*h*l^3 - 12*a^3*b*c^4*e*g*h^3*j - 12*a^3*b*c^4*e*f*h^3*k
 - 12*a^3*b*c^4*d*e*h^3*m - 12*a*b^2*c^5*d^3*e*j*k + 6*a^3*b*c^4*d*g*h^3*k
- 3*a^2*b^5*c*e*f*g*l^3 - 3*a^2*b^5*c*d*f*h*l^3 - 3*a*b^3*c^4*e^3*g*h*j - 3
*a*b^3*c^4*e^3*f*h*k - 3*a*b^3*c^4*e^3*f*g*l - 3*a*b^3*c^4*d*e^3*h*m + 24*a
*b^2*c^5*d^3*e*h*l - 12*a*b^2*c^5*d^3*f*h*k - 3*a*b^2*c^5*d^3*g*h*j - 3*a*b
^2*c^5*d^3*f*g*l - 3*a*b^2*c^5*d^3*e*g*m + 48*a^4*b*c^3*d*e*f*m^3 + 24*a^2*
b*c^5*d*e*f^3*m + 21*a^2*b^5*c*d*e*f*m^3 - 12*a^2*b*c^5*e*f^3*g*j - 12*a^2*
b*c^5*d*f^3*h*j - 9*a*b^3*c^4*d*e*f^3*m + 6*a^2*b*c^5*d*f^3*g*k + 12*a*b^2*
c^5*d*e^3*f*l - 6*a*b^2*c^5*d*e^3*g*k + 3*a*b^2*c^5*d*e^3*h*j - 24*a^3*b*c^
4*d*e*f*k^3 - 12*a^2*b*c^5*d*e*g^3*j - 3*a*b^5*c^2*d*e*f*k^3 + 3*a*b^2*c^5*
e^3*f*g*h - 12*a^2*b*c^5*d*f*g^3*h + 9*a*b^2*c^5*d*e*f^3*j + 9*a*b*c^6*d^2*
e^2*f*j + 3*a*b^4*c^3*d*e*f*j^3 + 9*a*b*c^6*d^2*e^2*g*h + 9*a*b*c^6*d^2*e*f
^2*h - 3*a*b^3*c^4*d*e*f*h^3 - 18*a*b*c^6*d^2*e*f*g^2 + 9*a*b*c^6*d*e^2*f^2
*g + 3*a*b^2*c^5*d*e*f*g^3 - 36*a^4*b^2*c^2*e^2*k*l^2*m - 9*a^4*b^2*c^2*g^2
*j^2*k*m + 45*a^3*b^3*c^2*d^2*k^2*l*m + 36*a^4*b^2*c^2*e^2*j*l*m^2 + 9*a^4*
b^2*c^2*g^2*j*k^2*l + 9*a^3*b^3*c^2*e^2*j^2*l*m + 9*a^4*b^2*c^2*g^2*h*k^2*m
 - 9*a^4*b^2*c^2*f^2*h*l^2*m - 9*a^3*b^3*c^2*f^2*j^2*k*l - 45*a^3*b^3*c^2*d
^2*j*k*m^2 + 36*a^3*b^2*c^3*d^2*j^2*k*m + 18*a^4*b^2*c^2*f^2*h*k*m^2 + 18*a
^4*b^2*c^2*f^2*g*l*m^2 - 9*a^4*b^2*c^2*g^2*h*k*l^2 - 9*a^4*b^2*c^2*f*h^2*k^
2*m - 9*a^4*b^2*c^2*f*g^2*l^2*m - 9*a^4*b^2*c^2*e*j^2*k^2*l - 9*a^4*b^2*c^2
*d*j^2*k^2*m - 9*a^3*b^3*c^2*e^2*j*k*l^2 - 9*a^2*b^4*c^2*d^2*j^2*k*m - 36*a
^3*b^2*c^3*d^2*j*k^2*l - 27*a^3*b^2*c^3*e^2*h^2*k*m + 9*a^4*b^2*c^2*g*h^2*j
*l^2 + 9*a^4*b^2*c^2*f*h^2*k*l^2 - 9*a^4*b^2*c^2*f*g^2*k*m^2 - 9*a^4*b^2*c^
2*e*g^2*l*m^2 - 9*a^4*b^2*c^2*d*j^2*k*l^2 + 9*a^4*b^2*c^2*d*h^2*l^2*m - 9*a
^3*b^3*c^2*e^2*g*l^2*m + 9*a^2*b^4*c^2*e^2*h^2*k*m + 9*a^2*b^4*c^2*d^2*j*k^
2*l - 45*a^3*b^3*c^2*e^2*h*j*m^2 + 36*a^4*b^2*c^2*e*h^2*j*m^2 + 36*a^3*b^2*
c^3*e^2*h*j^2*m - 36*a^3*b^2*c^3*d^2*h*k^2*m + 36*a^2*b^3*c^3*d^2*g^2*l*m -
 9*a^4*b^2*c^2*f*h*j^2*l^2 - 9*a^4*b^2*c^2*d*h^2*k*m^2 + 9*a^3*b^3*c^2*f^2*
h*j*l^2 + 9*a^3*b^3*c^2*e^2*f*l*m^2 + 9*a^3*b^3*c^2*e*h^2*j^2*m - 9*a^3*b^2
*c^3*f^2*h^2*j*l - 9*a^2*b^4*c^2*e^2*h*j^2*m + 9*a^2*b^4*c^2*d^2*h*k^2*m +
36*a^3*b^2*c^3*d^2*h*k*l^2 - 27*a^4*b^2*c^2*e*g*j^2*m^2 - 27*a^4*b^2*c^2*d*
h*j^2*m^2 - 9*a^4*b^2*c^2*d*h*k^2*l^2 - 9*a^3*b^3*c^2*e*f^2*k*m^2 - 9*a^3*b
^3*c^2*d*f^2*l*m^2 + 9*a^3*b^2*c^3*f^2*h*j^2*k + 9*a^3*b^2*c^3*f^2*g*j^2*l
- 9*a^3*b^2*c^3*e^2*g*k^2*l - 9*a^3*b^2*c^3*e^2*f*k^2*m - 9*a^3*b^2*c^3*d^2
*f*l^2*m - 9*a^2*b^4*c^2*d^2*h*k*l^2 + 9*a^2*b^3*c^3*d^2*h^2*k*l - 81*a^3*b
^2*c^3*d^2*g*j*m^2 + 54*a^2*b^4*c^2*d^2*g*j*m^2 - 45*a^3*b^3*c^2*d*g^2*j*m^
2 - 45*a^2*b^3*c^3*d^2*g*j^2*m + 36*a^3*b^2*c^3*d^2*f*k*m^2 + 36*a^3*b^2*c^

3*d*g^2*j^2*m + 18*a^3*b^2*c^3*e^2*g*j*l^2 + 18*a^3*b^2*c^3*e^2*f*k*l^2 + 1
8*a^3*b^2*c^3*d*e^2*l^2*m - 9*a^4*b^2*c^2*d*f*k^2*m^2 - 9*a^3*b^3*c^2*f^2*g
*h*m^2 - 9*a^3*b^3*c^2*d*h^2*j*l^2 - 9*a^3*b^2*c^3*f^2*g*j*k^2 - 9*a^3*b^2*
c^3*d^2*e*l*m^2 - 9*a^3*b^2*c^3*f*g^2*h^2*m - 9*a^3*b^2*c^3*e*g^2*j^2*l - 9
*a^3*b^2*c^3*e*f^2*k^2*l - 9*a^2*b^4*c^2*d^2*f*k*m^2 - 9*a^2*b^4*c^2*d*g^2*
j^2*m - 9*a^2*b^3*c^3*e^2*h^2*j*k - 9*a^2*b^2*c^4*d^2*f^2*k*m - 27*a^2*b^2*
c^4*d^2*g^2*j*l - 9*a^3*b^3*c^2*f*g*h^2*l^2 + 9*a^3*b^2*c^3*e*g^2*j*k^2 - 9
*a^3*b^2*c^3*e*f^2*j*l^2 - 9*a^3*b^2*c^3*d*h^2*j^2*k - 9*a^3*b^2*c^3*d*f^2*
k*l^2 - 9*a^3*b^2*c^3*d*e^2*k*m^2 - 9*a^2*b^3*c^3*e^2*g*h^2*m - 9*a^2*b^3*c
^3*d^2*h*j*k^2 - 9*a^2*b^3*c^3*d^2*f*k^2*l - 9*a^2*b^3*c^3*d^2*e*k^2*m + 36
*a^3*b^3*c^2*d*e*j^2*m^2 + 36*a^3*b^2*c^3*e^2*f*h*m^2 - 27*a^2*b^2*c^4*d^2*
g^2*h*m + 9*a^3*b^3*c^2*e*f*h^2*m^2 + 9*a^3*b^2*c^3*f*g^2*h*k^2 - 9*a^2*b^4
*c^2*e^2*f*h*m^2 + 9*a^2*b^3*c^3*d^2*e*k*l^2 - 9*a^2*b^2*c^4*e^2*f^2*h*m -
45*a^2*b^3*c^3*d^2*g*h*l^2 - 36*a^3*b^2*c^3*e*f^2*g*m^2 + 36*a^3*b^2*c^3*d*
g^2*h*l^2 - 36*a^3*b^2*c^3*d*f^2*h*m^2 + 36*a^2*b^2*c^4*d^2*g*h^2*l - 9*a^3
*b^2*c^3*e*g*h^2*k^2 + 9*a^2*b^4*c^2*e*f^2*g*m^2 - 9*a^2*b^4*c^2*d*g^2*h*l^
2 + 9*a^2*b^4*c^2*d*f^2*h*m^2 + 9*a^2*b^3*c^3*e^2*g*h*k^2 + 9*a^2*b^3*c^3*d
*g^2*h^2*l - 9*a^2*b^3*c^3*d*e^2*j*l^2 - 9*a^2*b^2*c^4*e^2*g^2*h*k - 9*a^2*
b^2*c^4*e^2*f*g^2*m - 9*a^2*b^2*c^4*d^2*f*j^2*k - 9*a^2*b^2*c^4*d^2*f*h^2*m
 - 9*a^2*b^2*c^4*d^2*e*j^2*l - 45*a^2*b^3*c^3*d^2*f*g*m^2 + 36*a^3*b^2*c^3*
d*f*g^2*m^2 - 27*a^3*b^2*c^3*d*f*h^2*l^2 + 18*a^2*b^2*c^4*d^2*e*j*k^2 + 9*a
^2*b^4*c^2*d*f*h^2*l^2 - 9*a^2*b^4*c^2*d*f*g^2*m^2 - 9*a^2*b^3*c^3*e^2*f*g*
l^2 + 9*a^2*b^2*c^4*e^2*g*h^2*j + 9*a^2*b^2*c^4*e^2*f*h^2*k - 9*a^2*b^2*c^4
*e*f^2*g^2*l - 9*a^2*b^2*c^4*d*f^2*g^2*m - 9*a^2*b^2*c^4*d*e^2*j^2*k + 9*a^
2*b^2*c^4*d*e^2*h^2*m + 18*a^4*b^2*c^2*f^2*j^2*m^2 + 18*a^3*b^2*c^3*e^2*h^2
*l^2 - 9*a^2*b^4*c^2*e^2*h^2*l^2 + 18*a^2*b^2*c^4*d^2*g^2*k^2 + 12*a^6*c^2*
j^3*k*l*m + 3*a^6*b^2*j*k*l*m^3 - 12*a^6*c^2*g*k^3*l*m - 12*a^5*c^3*g^3*k*l
*m - 24*a^6*c^2*e*k*l^3*m - 24*a^4*c^4*e^3*k*l*m + 12*a^6*c^2*h*j*k*l^3 + 1
2*a^6*c^2*f*j*l^3*m + 12*a^5*c^3*h^3*j*k*l - 3*a^5*b^3*h*j*k*m^3 - 3*a^5*b^
3*g*j*l*m^3 - 3*a^5*b^3*f*k*l*m^3 + 12*a^6*c^2*g*h*l^3*m + 12*a^5*c^3*g*h^3
*l*m - 12*a^6*c^2*e*j*k*m^3 - 12*a^6*c^2*d*j*l*m^3 - 12*a^5*c^3*f*j^3*k*l -
 12*a^5*c^3*e*j^3*k*m - 12*a^5*c^3*d*j^3*l*m - 12*a^4*c^4*f^3*j*k*l + 24*a^
6*c^2*f*h*k*m^3 + 24*a^6*c^2*f*g*l*m^3 + 24*a^4*c^4*f^3*h*k*m + 24*a^4*c^4*
f^3*g*l*m - 12*a^6*c^2*g*h*j*m^3 - 12*a^6*c^2*e*h*l*m^3 - 12*a^5*c^3*g*h*j^
3*m + 3*b^6*c^2*d^3*j*k*l + 3*a^4*b^4*e*j*k*m^3 + 3*a^4*b^4*d*j*l*m^3 - 24*
a^5*c^3*d*j*k^3*l - 24*a^3*c^5*d^3*j*k*l - 6*a^4*b^4*e*h*l*m^3 + 3*b^6*c^2*
d^3*h*k*m + 3*b^6*c^2*d^3*g*l*m + 3*a^6*b*c*j^2*l^3*m + 3*a^4*b^4*g*h*j*m^3
 + 3*a^4*b^4*f*h*k*m^3 + 3*a^4*b^4*f*g*l*m^3 - 24*a^5*c^3*d*h*k^3*m - 24*a^
3*c^5*d^3*h*k*m + 12*a^5*c^3*g*h*j*k^3 + 12*a^5*c^3*f*g*k^3*l + 12*a^5*c^3*
e*h*k^3*l + 12*a^5*c^3*e*g*k^3*m + 12*a^4*c^4*g^3*h*j*k + 12*a^4*c^4*f*g^3*
k*l + 12*a^4*c^4*f*g^3*j*m + 12*a^4*c^4*e*g^3*k*m + 12*a^4*c^4*d*g^3*l*m +
12*a^3*c^5*d^3*g*l*m + 3*a^6*b*c*j*k^3*m^2 - 9*a^6*b*c*h^2*l*m^3 - 3*a^5*b*
c^2*j^4*k*l + 24*a^5*c^3*e*g*j*l^3 + 24*a^5*c^3*e*f*k*l^3 + 24*a^5*c^3*d*e*
l^3*m + 24*a^3*c^5*e^3*g*j*l + 24*a^3*c^5*e^3*f*k*l + 24*a^3*c^5*d*e^3*l*m
 - 12*a^5*c^3*d*h*j*l^3 - 12*a^5*c^3*d*g*k*l^3 - 12*a^4*c^4*e*h^3*j*k - 12*a

^4*c^4*d*h^3*j*l - 12*a^3*c^5*e^3*h*j*k - 12*a^3*c^5*e^3*f*j*m + 9*a^4*b*c^
3*g^4*l*m + 6*b^5*c^3*d^3*f*j*m + 6*a^3*b^5*d*g*k*m^3 - 3*b^5*c^3*d^3*h*j*k
 - 3*b^5*c^3*d^3*g*j*l - 3*b^5*c^3*d^3*f*k*l - 3*b^5*c^3*d^3*e*k*m - 3*a^3*
b^5*e*g*j*m^3 - 3*a^3*b^5*e*f*k*m^3 - 3*a^3*b^5*d*h*j*m^3 - 3*a^3*b^5*d*f*l
*m^3 - 12*a^5*c^3*f*g*h*l^3 - 12*a^4*c^4*f*g*h^3*l - 12*a^4*c^4*e*g*h^3*m -
 12*a^3*c^5*e^3*g*h*m - 9*a^6*b*c*g*k^2*m^3 - 3*b^5*c^3*d^3*g*h*m + 3*a^6*b
*c*f*l^3*m^2 - 3*a^3*b^5*f*g*h*m^3 + 12*a^5*c^3*d*e*j*m^3 + 12*a^4*c^4*e*f*
j^3*k + 12*a^4*c^4*d*g*j^3*k + 12*a^4*c^4*d*f*j^3*l + 12*a^4*c^4*d*e*j^3*m
+ 12*a^3*c^5*e*f^3*j*k + 12*a^3*c^5*d*f^3*j*l - 9*a^6*b*c*e*l^2*m^3 - 24*a^
5*c^3*e*f*g*m^3 - 24*a^5*c^3*d*f*h*m^3 - 24*a^3*c^5*e*f^3*g*m - 24*a^3*c^5*
d*f^3*h*m - 15*a^2*b*c^5*d^4*l*m + 15*a*b^3*c^4*d^4*l*m + 12*a^4*c^4*f*g*h*
j^3 + 12*a^3*c^5*f^3*g*h*j + 12*a^3*c^5*e*f^3*h*l + 9*a^3*b*c^4*f^4*k*l - 9
*a^3*b*c^4*f^4*j*m + 3*b^4*c^4*d^3*e*j*k + 3*a^5*b^2*c*g*j*l^4 + 3*a^5*b^2*
c*f*k*l^4 + 3*a^5*b^2*c*d*l^4*m - 3*a^5*b*c^2*h*j*k^4 - 3*a^5*b*c^2*f*k^4*l
 - 3*a^5*b*c^2*e*k^4*m - 3*a^4*b*c^3*h^4*j*k + 3*a^2*b^6*d*e*j*m^3 + 3*a*b^
4*c^3*e^4*k*m + 24*a^4*c^4*d*e*j*k^3 + 24*a^2*c^6*d^3*e*j*k - 6*b^4*c^4*d^3
*e*h*l + 3*b^4*c^4*d^3*g*h*j + 3*b^4*c^4*d^3*f*h*k + 3*b^4*c^4*d^3*f*g*l +
3*b^4*c^4*d^3*e*g*m - 3*a^4*b*c^3*g*h^4*m + 3*a^2*b^6*e*f*g*m^3 + 3*a^2*b^6
*d*f*h*m^3 - 3*a*b^6*c*e^3*j*m^2 + 24*a^4*c^4*d*f*h*k^3 + 24*a^2*c^6*d^3*f*
h*k - 12*a^4*c^4*e*f*g*k^3 - 12*a^3*c^5*e*f*g^3*k - 12*a^3*c^5*d*g^3*h*j -
12*a^3*c^5*d*f*g^3*l - 12*a^3*c^5*d*e*g^3*m - 12*a^2*c^6*d^3*g*h*j - 12*a^2
*c^6*d^3*f*g*l - 12*a^2*c^6*d^3*e*h*l - 12*a^2*c^6*d^3*e*g*m - 12*a*b^2*c^5
*d^4*j*l + 9*a^5*b*c^2*d*j*l^4 + 9*a^2*b*c^5*e^4*j*k - 3*a^4*b^3*c*d*j*l^4
- 3*a^4*b*c^3*e*j^4*k - 3*a^4*b*c^3*d*j^4*l - 3*a*b^3*c^4*e^4*j*k - 24*a^4*
c^4*d*e*f*l^3 - 24*a^2*c^6*d*e^3*f*l - 12*a^5*b^2*c*e*g*m^4 - 12*a^5*b^2*c*
d*h*m^4 + 12*a^3*c^5*d*e*h^3*j + 12*a^2*c^6*d*e^3*h*j + 12*a^2*c^6*d*e^3*g*
k - 12*a*b^2*c^5*d^4*h*m + 9*a^5*b*c^2*f*g*l^4 - 9*a^5*b*c^2*e*h*l^4 - 9*a^
2*b*c^5*e^4*h*l + 9*a^2*b*c^5*e^4*g*m + 6*a^4*b^3*c*e*h*l^4 + 6*a*b^3*c^4*e
^4*h*l - 3*b^3*c^5*d^3*e*g*j - 3*b^3*c^5*d^3*e*f*k - 3*a^4*b^3*c*f*g*l^4 -
3*a^4*b*c^3*g*h*j^4 - 3*a^3*b*c^4*g^4*h*j - 3*a^3*b*c^4*f*g^4*l - 3*a^3*b*c
^4*e*g^4*m - 3*a*b^3*c^4*e^4*g*m + 12*a^3*c^5*e*f*g*h^3 + 12*a^2*c^6*e^3*f*
g*h - 3*b^3*c^5*d^3*f*g*h - 12*a^3*c^5*d*e*f*j^3 - 12*a^2*c^6*d*e*f^3*j - 3
*a*b^6*c*d^2*g*l^3 - 15*a^5*b*c^2*d*e*m^4 + 15*a^4*b^3*c*d*e*m^4 + 9*a^4*b*
c^3*e*f*k^4 - 9*a^4*b*c^3*d*g*k^4 + 3*a^3*b^4*c*d*f*l^4 - 3*a^3*b*c^4*d*h^4
*j - 3*a^2*b*c^5*e*f^4*k - 3*a^2*b*c^5*d*f^4*l + 3*a*b^2*c^5*e^4*g*j + 3*a*
b^2*c^5*e^4*f*k + 3*a*b^2*c^5*d*e^4*m - 9*a*b*c^6*d^3*e^2*l + 3*b^2*c^6*d^3
*e*f*g - 3*a^3*b*c^4*f*g*h^4 - 3*a^2*b*c^5*f^4*g*h + 12*a^2*c^6*d*e*f*g^3 -
 9*a*b*c^6*d^3*f^2*j + 3*a*b*c^6*d^2*e^3*k + 9*a^3*b*c^4*d*e*j^4 - 3*a^2*b*
c^5*e*f*g^4 - 9*a*b*c^6*d^3*e*h^2 + 3*a*b*c^6*d^2*f^3*g + 3*a*b*c^6*d*e^3*g
^2 - 3*a^4*b^2*c^2*h^3*j^2*m + 12*a^4*b^2*c^2*g^3*j*m^2 - 3*a^4*b^2*c^2*f^2
*k^3*m + 3*a^3*b^3*c^2*g^3*j^2*m - 9*a^3*b^4*c*f^2*j^2*m^2 + 9*a^3*b^3*c^2*
f^2*j^3*m - 6*a^3*b^3*c^2*f^3*j*m^2 - 6*a^3*b^2*c^3*f^3*j^2*m - 3*a^2*b^4*c
^2*f^3*j^2*m - 27*a^4*b^2*c^2*d^2*k*m^3 - 27*a^3*b^2*c^3*e^3*j*m^2 + 18*a^2
*b^4*c^2*e^3*j*m^2 - 15*a^2*b^3*c^3*e^3*j^2*m + 12*a^4*b^2*c^2*f^2*j*l^3 +
3*a^3*b^3*c^2*e^2*k^3*l + 42*a^2*b^3*c^3*d^3*j*m^2 - 27*a^2*b^2*c^4*d^3*j^2

*m - 15*a^3*b^3*c^2*d^2*k*l^3 - 3*a^4*b^2*c^2*f*j^2*k^3 - 3*a^4*b^2*c^2*f*h
^3*m^2 + 3*a^3*b^3*c^2*g^3*h*l^2 + 3*a^3*b^3*c^2*f^2*j*k^3 - 3*a^3*b^2*c^3*
g^3*h^2*l - 3*a^3*b^2*c^3*e^2*j^3*l - 27*a^4*b^2*c^2*e^2*h*m^3 + 12*a^3*b^2
*c^3*f^3*h*l^2 + 3*a^3*b^3*c^2*f*g^3*m^2 - 3*a^2*b^4*c^2*f^3*h*l^2 + 3*a^2*
b^3*c^3*f^3*h^2*l + 9*a^3*b^3*c^2*e*h^3*l^2 + 9*a^2*b^3*c^3*e^2*h^3*l - 6*a
^4*b^2*c^2*e*h^2*l^3 - 6*a^3*b^3*c^2*e^2*h*l^3 - 6*a^2*b^3*c^3*e^3*h*l^2 -
6*a^2*b^2*c^4*e^3*h^2*l + 3*a^2*b^3*c^3*d^2*j^3*k + 42*a^3*b^3*c^2*d^2*g*m^
3 - 27*a^4*b^2*c^2*d*g^2*m^3 - 27*a^2*b^2*c^4*d^3*h*l^2 - 15*a^2*b^3*c^3*e^
3*f*m^2 + 12*a^3*b^2*c^3*e^2*h*k^3 + 3*a^3*b^3*c^2*e*h^2*k^3 - 3*a^3*b^2*c^
3*e*g^3*l^2 - 3*a^2*b^4*c^2*e^2*h*k^3 + 3*a^2*b^3*c^3*f^3*g*k^2 - 3*a^2*b^2
*c^4*f^3*g^2*k - 27*a^3*b^2*c^3*d^2*g*l^3 - 27*a^2*b^2*c^4*d^3*f*m^2 + 18*a
^2*b^4*c^2*d^2*g*l^3 - 15*a^3*b^3*c^2*d*g^2*l^3 + 12*a^2*b^2*c^4*e^3*g*k^2
- 3*a^3*b^2*c^3*e*h^2*j^3 + 3*a^2*b^3*c^3*e^2*h*j^3 + 3*a^2*b^3*c^3*e*f^3*l
^2 - 3*a^2*b^2*c^4*d^2*h^3*k + 9*a^2*b^3*c^3*d*g^3*k^2 - 9*a*b^4*c^3*d^2*g^
2*k^2 - 6*a^3*b^2*c^3*d*g^2*k^3 - 6*a^2*b^3*c^3*d^2*g*k^3 - 3*a^2*b^4*c^2*d
*g^2*k^3 + 12*a^2*b^2*c^4*d^2*g*j^3 + 3*a^2*b^3*c^3*d*g^2*j^3 - 3*a^2*b^2*c
^4*d*f^3*k^2 - 3*a^2*b^2*c^4*d*g^2*h^3 + 12*a^7*c*j*k*l*m^3 - 3*b^7*c*d^3*k
*l*m - 3*a^6*b*c*k^4*l*m - 3*a^6*b*c*j*k*l^4 - 3*a^6*b*c*g*l^4*m - 9*a^6*b*
c*f*j*m^4 + 9*a^6*b*c*e*k*m^4 + 9*a^6*b*c*d*l*m^4 + 9*a^6*b*c*g*h*m^4 - 3*a
*b^7*d*e*f*m^3 + 9*a*b*c^6*d^4*h*j - 9*a*b*c^6*d^4*g*k + 9*a*b*c^6*d^4*f*l
+ 9*a*b*c^6*d^4*e*m + 12*a*c^7*d^3*e*f*g - 3*a*b*c^6*d*e^4*j - 3*a*b*c^6*e^
4*f*g - 3*a*b*c^6*d*e*f^4 + 18*a^6*c^2*h^2*j*l*m^2 - 18*a^6*c^2*h*j^2*l^2*m
 + 18*a^6*c^2*f*k^2*l^2*m + 36*a^5*c^3*e^2*k*l^2*m + 18*a^6*c^2*g*j*k^2*m^2
 + 18*a^6*c^2*e*k^2*l*m^2 + 18*a^5*c^3*g^2*j^2*k*m + 18*a^6*c^2*e*j*l^2*m^2
 + 18*a^6*c^2*d*k*l^2*m^2 - 18*a^5*c^3*e^2*j*l*m^2 - 18*a^6*c^2*f*h*l^2*m^2
 + 18*a^5*c^3*f^2*h*l^2*m - 36*a^5*c^3*f^2*h*k*m^2 - 36*a^5*c^3*f^2*g*l*m^2
 + 18*a^5*c^3*g^2*h*k*l^2 - 18*a^5*c^3*g*h^2*k^2*l + 18*a^5*c^3*f*h^2*k^2*m
 + 18*a^5*c^3*f*g^2*l^2*m + 18*a^5*c^3*e*j^2*k^2*l + 18*a^5*c^3*d*j^2*k^2*m
 - 18*a^4*c^4*d^2*j^2*k*m + 36*a^4*c^4*d^2*j*k^2*l + 18*a^5*c^3*f*g^2*k*m^2
 + 18*a^5*c^3*e*g^2*l*m^2 + 18*a^5*c^3*d*j^2*k*l^2 - 18*a^4*c^4*f^2*g^2*k*m
 + 36*a^4*c^4*d^2*h*k^2*m + 18*a^5*c^3*f*h*j^2*l^2 - 18*a^5*c^3*e*h^2*j*m^2
 + 18*a^5*c^3*d*h^2*k*m^2 + 18*a^4*c^4*f^2*h^2*j*l - 18*a^4*c^4*e^2*h*j^2*m
 - 18*a^5*c^3*e*g*k^2*l^2 + 18*a^5*c^3*d*h*k^2*l^2 + 18*a^4*c^4*e^2*g*k^2*l
 + 18*a^4*c^4*e^2*f*k^2*m - 18*a^4*c^4*d^2*h*k*l^2 + 18*a^4*c^4*d^2*f*l^2*m
 - 36*a^4*c^4*e^2*g*j*l^2 - 36*a^4*c^4*e^2*f*k*l^2 - 36*a^4*c^4*d*e^2*l^2*m
 + 18*a^5*c^3*d*f*k^2*m^2 + 18*a^4*c^4*f^2*g*j*k^2 + 18*a^4*c^4*d^2*g*j*m^2
 - 18*a^4*c^4*d^2*f*k*m^2 + 18*a^4*c^4*d^2*e*l*m^2 - 18*a^4*c^4*f*g^2*j^2*k
 + 18*a^4*c^4*f*g^2*h^2*m + 18*a^4*c^4*e*g^2*j^2*l + 18*a^4*c^4*e*f^2*k^2*l
 - 18*a^4*c^4*d*g^2*j^2*m - 18*a^4*c^4*d*f^2*k^2*m + 18*a^3*c^5*d^2*f^2*k*m
 + 3*a^4*b^2*c^2*h^4*k*m - 3*a^3*b^3*c^2*g^4*l*m + 18*a^4*c^4*e*f^2*j*l^2 +
 18*a^4*c^4*d*h^2*j^2*k + 18*a^4*c^4*d*f^2*k*l^2 + 18*a^4*c^4*d*e^2*k*m^2 -
 18*a^3*c^5*e^2*f^2*j*l + 12*a^5*b^2*c*g^2*k*m^3 - 9*a^5*b*c^2*h^3*j*m^2 -
9*a^5*b*c^2*f^2*l^3*m + 3*a^5*b*c^2*h^2*k^3*l + 3*a^4*b^3*c*h^3*j*m^2 + 3*a
^4*b^3*c*f^2*l^3*m - 18*a^4*c^4*e^2*f*h*m^2 + 18*a^3*c^5*e^2*f^2*h*m + 15*a
^5*b*c^2*e^2*l*m^3 - 15*a^4*b^3*c*e^2*l*m^3 - 9*a^5*b*c^2*g^2*k*l^3 - 9*a^4

*b*c^3*g^3*j^2*m - 3*a^5*b^2*c*g*k^2*l^3 + 3*a^5*b*c^2*h*j^3*l^2 + 3*a^4*b^
3*c*g^2*k*l^3 - 3*a^3*b^4*c*g^3*j*m^2 + 36*a^4*c^4*e*f^2*g*m^2 + 36*a^4*c^4
*d*f^2*h*m^2 + 18*a^4*c^4*e*g*h^2*k^2 - 18*a^4*c^4*d*g^2*h*l^2 - 18*a^4*c^4
*d*f*j^2*k^2 + 18*a^3*c^5*e^2*g^2*h*k + 18*a^3*c^5*e^2*f*g^2*m - 18*a^3*c^5
*d^2*g*h^2*l + 18*a^3*c^5*d^2*f*j^2*k + 18*a^3*c^5*d^2*f*h^2*m + 18*a^3*c^5
*d^2*e*j^2*l - 12*a^2*b^2*c^4*e^4*k*m + 9*a^4*b^3*c*f*j^3*m^2 - 9*a^4*b^2*c
^2*f*j^4*m - 6*a^5*b^2*c*f*j^2*m^3 + 6*a^5*b*c^2*f^2*j*m^3 - 6*a^5*b*c^2*f*
j^3*m^2 - 6*a^4*b^3*c*f^2*j*m^3 + 6*a^4*b*c^3*f^3*j*m^2 - 6*a^4*b*c^3*f^2*j
^3*m + 6*a^2*b^3*c^3*f^4*j*m + 3*a^3*b^2*c^3*g^4*j*l + 3*a^2*b^5*c*f^3*j*m^
2 - 3*a^2*b^3*c^3*f^4*k*l - 36*a^3*c^5*d^2*e*j*k^2 - 18*a^4*c^4*d*f*g^2*m^2
 + 18*a^3*c^5*e*f^2*g^2*l + 18*a^3*c^5*d*f^2*g^2*m + 18*a^3*c^5*d*e^2*j^2*k
 + 18*a^3*b^4*c*d^2*k*m^3 + 15*a^3*b*c^4*e^3*j^2*m + 12*a^5*b^2*c*d*k^2*m^3
 - 9*a^5*b*c^2*f*j^2*l^3 - 9*a^4*b*c^3*e^2*k^3*l + 3*a^5*b*c^2*e*k^3*l^2 +
3*a^4*b^3*c*f*j^2*l^3 + 3*a^4*b*c^3*g^2*j^3*k - 3*a^3*b^4*c*f^2*j*l^3 + 3*a
^3*b^2*c^3*g^4*h*m + 3*a*b^5*c^2*e^3*j^2*m - 36*a^3*c^5*d^2*f*h*k^2 - 21*a^
3*b*c^4*d^3*j*m^2 - 21*a*b^5*c^2*d^3*j*m^2 + 18*a^3*c^5*e^2*f*h*j^2 - 18*a^
3*c^5*e*f^2*h^2*j + 18*a^3*c^5*d*f^2*h^2*k + 18*a*b^4*c^3*d^3*j^2*m + 15*a^
4*b*c^3*d^2*k*l^3 - 9*a^5*b*c^2*d*k^2*l^3 - 9*a^4*b*c^3*g^3*h*l^2 - 9*a^4*b
*c^3*f^2*j*k^3 + 3*a^4*b^3*c*d*k^2*l^3 + 3*a^2*b^5*c*d^2*k*l^3 - 18*a^3*c^5
*d^2*e*g*l^2 + 18*a^3*c^5*d*e^2*h*k^2 + 18*a^3*b^4*c*e^2*h*m^3 - 18*a^2*c^6
*d^2*e^2*h*k + 18*a^2*c^6*d^2*e^2*g*l + 18*a^2*c^6*d^2*e^2*f*m + 15*a^5*b*c
^2*e*h^2*m^3 - 15*a^4*b^3*c*e*h^2*m^3 - 9*a^4*b*c^3*f*g^3*m^2 - 9*a^3*b*c^4
*f^3*h^2*l + 3*a^4*b^2*c^2*e*j*k^4 + 3*a^4*b*c^3*g*h^3*k^2 + 3*a^3*b*c^4*f^
2*g^3*m + 36*a^3*c^5*d*e^2*f*l^2 + 18*a^3*c^5*d*f*g^2*j^2 + 18*a^2*c^6*d^2*
f^2*g*j + 18*a^2*c^6*d^2*e*f^2*l - 9*a^3*b^2*c^3*e*h^4*l - 9*a^3*b*c^4*d^2*
j^3*k + 6*a^4*b*c^3*e^2*h*l^3 - 6*a^4*b*c^3*e*h^3*l^2 + 6*a^3*b*c^4*e^3*h*l
^2 - 6*a^3*b*c^4*e^2*h^3*l + 3*a^4*b^2*c^2*f*h*k^4 + 3*a^4*b*c^3*d*j^3*k^2
- 3*a^3*b^4*c*e*h^2*l^3 + 3*a^2*b^5*c*e^2*h*l^3 + 3*a^2*b^2*c^4*f^4*h*k + 3
*a^2*b^2*c^4*f^4*g*l + 3*a*b^5*c^2*e^3*h*l^2 - 3*a*b^4*c^3*e^3*h^2*l - 21*a
^4*b*c^3*d^2*g*m^3 - 21*a^2*b^5*c*d^2*g*m^3 + 18*a^3*b^4*c*d*g^2*m^3 + 18*a
^2*c^6*d*e^2*f^2*k + 18*a*b^4*c^3*d^3*h*l^2 + 15*a^3*b*c^4*e^3*f*m^2 + 15*a
^2*b*c^5*d^3*h^2*l - 15*a*b^3*c^4*d^3*h^2*l - 9*a^4*b*c^3*e*h^2*k^3 - 9*a^3
*b*c^4*f^3*g*k^2 - 9*a^2*b*c^5*e^3*f^2*m + 3*a^3*b*c^4*f^2*h^3*j + 3*a*b^5*
c^2*e^3*f*m^2 + 3*a*b^3*c^4*e^3*f^2*m + 18*a*b^4*c^3*d^3*f*m^2 + 15*a^4*b*c
^3*d*g^2*l^3 + 12*a*b^2*c^5*d^3*f^2*m - 9*a^3*b*c^4*e^2*h*j^3 - 9*a^3*b*c^4
*e*f^3*l^2 - 9*a^2*b*c^5*e^3*g^2*k + 3*a^3*b*c^4*f*g^3*j^2 + 3*a^2*b^5*c*d*
g^2*l^3 + 3*a^2*b*c^5*e^2*f^3*l - 3*a*b^4*c^3*e^3*g*k^2 + 3*a*b^3*c^4*e^3*g
^2*k + 18*a^2*c^6*d^2*e*g*h^2 - 18*a^2*c^6*d*e^2*g^2*h - 12*a^4*b^2*c^2*d*f
*l^4 - 9*a^2*b^2*c^4*d*g^4*k + 9*a*b^3*c^4*d^2*g^3*k + 6*a^3*b^3*c^2*d*g*k^
4 + 6*a^3*b*c^4*d^2*g*k^3 - 6*a^3*b*c^4*d*g^3*k^2 + 6*a^2*b*c^5*d^3*g*k^2 -
 6*a^2*b*c^5*d^2*g^3*k - 6*a*b^3*c^4*d^3*g*k^2 - 6*a*b^2*c^5*d^3*g^2*k - 3*
a^3*b^3*c^2*e*f*k^4 + 3*a^3*b^2*c^3*e*g*j^4 + 3*a^3*b^2*c^3*d*h*j^4 + 3*a*b
^5*c^2*d^2*g*k^3 + 15*a^2*b*c^5*d^3*e*l^2 - 15*a*b^3*c^4*d^3*e*l^2 - 9*a^3*
b*c^4*d*g^2*j^3 - 9*a^2*b*c^5*e^3*f*j^2 - 3*a*b^4*c^3*d^2*g*j^3 + 3*a*b^3*c
^4*e^3*f*j^2 - 3*a*b^2*c^5*e^3*f^2*j + 12*a*b^2*c^5*d^3*f*j^2 - 9*a^2*b*c^5

*d*e^3*k^2 + 3*a^2*b*c^5*e^2*g^3*h + 3*a*b^3*c^4*d*e^3*k^2 - 9*a^2*b*c^5*d^
2*g*h^3 - 3*a^2*b^3*c^3*d*e*j^4 + 3*a^2*b*c^5*e*f^3*h^2 + 3*a*b^3*c^4*d^2*g
*h^3 + 3*a^2*b^2*c^4*d*f*h^4 - 9*a^7*c*k^2*l^2*m^2 - 6*a^6*c^2*j^2*k^3*m -
3*a^6*b^2*h*l^2*m^3 + 3*a^5*b^3*h^2*l*m^3 - 6*a^6*c^2*g^2*k*m^3 - 6*a^6*c^2
*h*k^3*l^2 + 6*a^5*c^3*h^3*j^2*m + 6*a^6*c^2*g*k^2*l^3 - 6*a^6*c^2*f*k^3*m^
2 - 6*a^5*c^3*h^2*j^3*l - 6*a^5*c^3*g^3*j*m^2 + 6*a^5*c^3*f^2*k^3*m + 3*a^5
*b^3*g*k^2*m^3 - 3*a^4*b^4*g^2*k*m^3 + 12*a^6*c^2*f*j^2*m^3 + 12*a^4*c^4*f^
3*j^2*m + 3*a^5*b^3*e*l^2*m^3 + 3*a^3*b^5*e^2*l*m^3 - 6*a^6*c^2*d*k^2*m^3 -
6*a^5*c^3*f^2*j*l^3 + 6*a^5*c^3*d^2*k*m^3 - 6*a^5*c^3*g*j^3*k^2 + 6*a^4*c^
4*e^3*j*m^2 - 3*b^6*c^2*d^3*j^2*m - 3*a^4*b^4*f*j^2*m^3 + 3*a^3*b^5*f^2*j*m
^3 + 6*a^5*c^3*f*j^2*k^3 + 6*a^5*c^3*f*h^3*m^2 - 6*a^5*c^3*e*j^3*l^2 + 6*a^
4*c^4*g^3*h^2*l - 6*a^4*c^4*f^2*h^3*m + 6*a^4*c^4*e^2*j^3*l + 6*a^3*c^5*d^3
*j^2*m - 3*a^4*b^4*d*k^2*m^3 - 3*a^2*b^6*d^2*k*m^3 + 6*a^5*c^3*e^2*h*m^3 -
6*a^4*c^4*g^2*h^3*k - 6*a^4*c^4*f^3*h*l^2 + 12*a^5*c^3*e*h^2*l^3 + 12*a^3*c
^5*e^3*h^2*l - 3*b^6*c^2*d^3*h*l^2 + 3*b^5*c^3*d^3*h^2*l - 3*a^5*b^2*c*j^4*
m^2 + 3*a^3*b^5*e*h^2*m^3 - 3*a^2*b^6*e^2*h*m^3 + 6*a^5*c^3*d*g^2*m^3 - 6*a
^4*c^4*e^2*h*k^3 - 6*a^4*c^4*f*h^3*j^2 + 6*a^4*c^4*e*g^3*l^2 + 6*a^3*c^5*f^
3*g^2*k - 6*a^3*c^5*e^2*g^3*l + 6*a^3*c^5*d^3*h*l^2 - 3*b^6*c^2*d^3*f*m^2 -
3*b^4*c^4*d^3*f^2*m + 6*a^4*c^4*d^2*g*l^3 + 6*a^4*c^4*e*h^2*j^3 - 6*a^4*c^
4*d*h^3*k^2 - 6*a^3*c^5*f^2*g^3*j - 6*a^3*c^5*e^3*g*k^2 + 6*a^3*c^5*d^3*f*m
^2 + 6*a^3*c^5*d^2*h^3*k - 6*a^2*c^6*d^3*f^2*m + 4*a^5*b^2*c*h^3*m^3 + 3*b^
5*c^3*d^3*g*k^2 - 3*b^4*c^4*d^3*g^2*k - 3*a^2*b^6*d*g^2*m^3 + a^5*b*c^2*j^3
*k^3 + 12*a^4*c^4*d*g^2*k^3 + 12*a^2*c^6*d^3*g^2*k + 6*a^5*b*c^2*h^3*l^3 +
5*a^5*b*c^2*g^3*m^3 - 5*a^4*b^3*c*g^3*m^3 + 3*b^5*c^3*d^3*e*l^2 + 3*b^3*c^5
*d^3*e^2*l - 3*a^5*b^2*c*h^2*l^4 + a^4*b^3*c*h^3*l^3 + 12*a^5*b^2*c*f^2*m^4
- 6*a^3*c^5*d^2*g*j^3 + 6*a^3*c^5*d*f^3*k^2 + 6*a^3*b^4*c*f^3*m^3 + 6*a^2*
c^6*e^3*f^2*j - 6*a^2*c^6*d^2*f^3*k - 3*b^4*c^4*d^3*f*j^2 + 3*b^3*c^5*d^3*f
^2*j - 3*a^2*b^2*c^4*f^5*m - 7*a^4*b*c^3*e^3*m^3 - 7*a^2*b^5*c*e^3*m^3 + 6*
a^4*b*c^3*g^3*k^3 - 6*a^3*c^5*e*g^3*h^2 - 6*a^2*c^6*d^3*f*j^2 + 5*a^4*b*c^3
*f^3*l^3 + a^4*b*c^3*h^3*j^3 + a^2*b^5*c*f^3*l^3 + 6*a^3*c^5*d*g^2*h^3 - 6*
a^2*c^6*e^2*f^3*h - 3*a^3*b^4*c*e^2*l^4 - 3*a*b^4*c^3*e^4*l^2 - 7*a^3*b*c^4
*d^3*l^3 - 7*a*b^5*c^2*d^3*l^3 + 6*a^3*b*c^4*f^3*j^3 + 5*a^3*b*c^4*e^3*k^3
+ 3*b^3*c^5*d^3*e*h^2 - 3*b^2*c^6*d^3*e^2*h + a*b^5*c^2*e^3*k^3 + 12*a*b^2*
c^5*d^4*k^2 - 6*a^2*c^6*d*f^3*g^2 + 6*a*b^4*c^3*d^3*k^3 - 3*a^4*b^2*c^2*d*k
^5 + a^3*b*c^4*g^3*h^3 + 5*a^2*b*c^5*d^3*j^3 - 5*a*b^3*c^4*d^3*j^3 - 9*a*c^
7*d^2*e^2*f^2 + 6*a^2*b*c^5*e^3*h^3 - 3*a*b^2*c^5*e^4*h^2 + a^2*b*c^5*f^3*g
^3 + a*b^3*c^4*e^3*h^3 + 4*a*b^2*c^5*d^3*h^3 - 3*a*b^2*c^5*d^2*g^4 - 6*a^7*
c*j*l^3*m^2 + 6*a^7*c*h*l^2*m^3 + 6*a^6*c^2*j*k^4*l + 6*a^6*c^2*h*k^4*m - 6
*a^5*c^3*h^4*k*m + 3*a^6*b^2*h*k*m^4 + 3*a^6*b^2*g*l*m^4 - 3*b^5*c^3*d^4*l*
m - 6*a^6*c^2*g*j*l^4 - 6*a^6*c^2*f*k*l^4 - 6*a^6*c^2*d*l^4*m + 6*a^5*c^3*h
*j^4*k + 6*a^5*c^3*g*j^4*l + 6*a^5*c^3*f*j^4*m - 6*a^4*c^4*g^4*j*l + 6*a^3*
c^5*e^4*k*m + 6*a^5*b^3*f*j*m^4 - 6*a^4*c^4*g^4*h*m + 3*b^7*c*d^3*j*m^2 - 3
*a^5*b^3*e*k*m^4 - 3*a^5*b^3*d*l*m^4 + 3*b^4*c^4*d^4*j*l - 3*a^5*b^3*g*h*m^
4 - 6*a^5*c^3*e*j*k^4 + 6*a^2*c^6*d^4*j*l + 3*b^4*c^4*d^4*h*m + 6*a^6*c^2*e
*g*m^4 + 6*a^6*c^2*d*h*m^4 + 6*a^6*b*c*j^3*m^3 - 6*a^5*c^3*f*h*k^4 + 6*a^4*

c^4*g*h^4*j + 6*a^4*c^4*f*h^4*k + 6*a^4*c^4*e*h^4*l + 6*a^4*c^4*d*h^4*m - 6
*a^3*c^5*f^4*h*k - 6*a^3*c^5*f^4*g*l + 6*a^2*c^6*d^4*h*m + 3*a^5*b*c^2*j^5*
m + a^6*b*c*k^3*l^3 + 3*a^4*b^4*e*g*m^4 + 3*a^4*b^4*d*h*m^4 + 6*b^3*c^5*d^4
*g*k - 3*b^3*c^5*d^4*h*j - 3*b^3*c^5*d^4*f*l - 3*b^3*c^5*d^4*e*m + 3*a*b^7*
d^2*g*m^3 + 6*a^5*c^3*d*f*l^4 - 6*a^4*c^4*e*g*j^4 - 6*a^4*c^4*d*h*j^4 + 6*a
^3*c^5*e*g^4*j + 6*a^3*c^5*d*g^4*k - 6*a^2*c^6*e^4*g*j - 6*a^2*c^6*e^4*f*k
- 6*a^2*c^6*d*e^4*m + 3*a^4*b*c^3*h^5*l + 6*a^3*c^5*f*g^4*h - 3*a^3*b^5*d*e
*m^4 + 3*b^2*c^6*d^4*e*j + 3*a^5*b*c^2*g*k^5 + 3*a^3*b*c^4*g^5*k + 8*a*b^6*
c*d^3*m^3 + 3*b^2*c^6*d^4*f*h - 3*a^5*b^2*c*e*l^5 - 3*a*b^2*c^5*e^5*l - 6*a
^3*c^5*d*f*h^4 + 6*a^2*c^6*e*f^4*g + 6*a^2*c^6*d*f^4*h + 3*a^4*b*c^3*f*j^5
+ 3*a^2*b*c^5*f^5*j + 6*a*c^7*d^3*e^2*h - 6*a*c^7*d^2*e^3*g + 3*a^3*b*c^4*e
*h^5 + 6*a*b*c^6*d^3*g^3 + 3*a^2*b*c^5*d*g^5 + a*b*c^6*e^3*f^3 - 9*a^6*c^2*
j^2*k^2*l^2 - 9*a^6*c^2*h^2*k^2*m^2 - 9*a^6*c^2*g^2*l^2*m^2 - 18*a^5*c^3*f^
2*j^2*m^2 - 9*a^5*c^3*h^2*j^2*k^2 - 9*a^5*c^3*g^2*j^2*l^2 - 9*a^5*c^3*f^2*k
^2*l^2 - 9*a^5*c^3*e^2*k^2*m^2 - 9*a^5*c^3*d^2*l^2*m^2 - 9*a^5*c^3*g^2*h^2*
m^2 - 9*a^4*c^4*e^2*j^2*k^2 - 9*a^4*c^4*d^2*j^2*l^2 - 18*a^4*c^4*e^2*h^2*l^
2 - 9*a^4*c^4*g^2*h^2*j^2 - 9*a^4*c^4*f^2*h^2*k^2 - 9*a^4*c^4*f^2*g^2*l^2 -
9*a^4*c^4*e^2*g^2*m^2 - 9*a^4*c^4*d^2*h^2*m^2 - 18*a^3*c^5*d^2*g^2*k^2 - 9
*a^3*c^5*e^2*g^2*j^2 - 9*a^3*c^5*e^2*f^2*k^2 - 9*a^3*c^5*d^2*h^2*j^2 - 9*a^
3*c^5*d^2*f^2*l^2 - 9*a^3*c^5*d^2*e^2*m^2 - 3*a^4*b^2*c^2*h^4*l^2 - 18*a^4*
b^2*c^2*f^3*m^3 + 12*a^3*b^2*c^3*f^4*m^2 - 9*a^3*c^5*f^2*g^2*h^2 + 4*a^4*b^
2*c^2*g^3*l^3 - 3*a^2*b^4*c^2*f^4*m^2 + 14*a^3*b^3*c^2*e^3*m^3 - 5*a^3*b^3*
c^2*f^3*l^3 - 3*a^4*b^2*c^2*g^2*k^4 - 3*a^3*b^2*c^3*g^4*k^2 + a^3*b^3*c^2*g
^3*k^3 - 20*a^2*b^4*c^2*d^3*m^3 - 18*a^3*b^2*c^3*e^3*l^3 + 16*a^3*b^2*c^3*d
^3*m^3 + 12*a^4*b^2*c^2*e^2*l^4 + 12*a^2*b^2*c^4*e^4*l^2 - 9*a^2*c^6*d^2*e^
2*j^2 + 6*a^2*b^4*c^2*e^3*l^3 + 4*a^3*b^2*c^3*f^3*k^3 + 14*a^2*b^3*c^3*d^3*
l^3 - 9*a^2*c^6*e^2*f^2*g^2 - 9*a^2*c^6*d^2*f^2*h^2 - 5*a^2*b^3*c^3*e^3*k^3
- 3*a^3*b^2*c^3*f^2*j^4 - 3*a^2*b^2*c^4*f^4*j^2 + a^2*b^3*c^3*f^3*j^3 - 18
*a^2*b^2*c^4*d^3*k^3 + 12*a^3*b^2*c^3*d^2*k^4 + 4*a^2*b^2*c^4*e^3*j^3 - 3*a
^2*b^4*c^2*d^2*k^4 - 3*a^2*b^2*c^4*e^2*h^4 + 6*a^7*c*k*l^4*m - 3*a^7*b*k*l*
m^4 - 6*a^7*c*h*k*m^4 - 6*a^7*c*g*l*m^4 + 3*a^6*b*c*h*l^5 - 6*a*c^7*d^4*e*j
- 6*a*c^7*d^4*f*h - 3*b*c^7*d^4*e*f + 6*a*c^7*d*e^4*f + 3*a*b*c^6*e^5*h -
a^5*b^2*c*j^3*l^3 - a^3*b^4*c*g^3*l^3 - a*b^4*c^3*e^3*j^3 - a*b^2*c^5*e^3*g
^3 + 3*a^7*b*j*m^5 + 6*a^7*c*f*m^5 + 6*a*c^7*d^5*k + 3*b*c^7*d^5*g - 3*a^6*
c^2*j^4*m^2 - 3*a^6*b^2*j^2*m^4 + 2*a^6*c^2*j^3*l^3 + a^5*b^3*j^3*m^3 - 2*a
^6*c^2*h^3*m^3 - 3*a^6*c^2*h^2*l^4 - 3*a^5*c^3*h^4*l^2 - a*b^6*c*e^3*l^3 +
20*a^5*c^3*f^3*m^3 - 15*a^6*c^2*f^2*m^4 - 15*a^4*c^4*f^4*m^2 + 2*a^5*c^3*h^
3*k^3 - 2*a^5*c^3*g^3*l^3 + a^3*b^5*g^3*m^3 - 3*a^5*c^3*g^2*k^4 - 3*a^4*c^4
*g^4*k^2 - 3*a^4*b^4*f^2*m^4 + 20*a^4*c^4*e^3*l^3 - 15*a^5*c^3*e^2*l^4 - 15
*a^3*c^5*e^4*l^2 + 2*a^4*c^4*g^3*j^3 - 2*a^4*c^4*f^3*k^3 - 2*a^4*c^4*d^3*m^
3 - 3*b^4*c^4*d^4*k^2 - 3*a^4*c^4*f^2*j^4 - 3*a^3*c^5*f^4*j^2 + 20*a^3*c^5*
d^3*k^3 - 15*a^4*c^4*d^2*k^4 - 15*a^2*c^6*d^4*k^2 - 2*a^3*c^5*e^3*j^3 + b^5
*c^3*d^3*j^3 + 2*a^3*c^5*f^3*h^3 - 3*a^3*c^5*e^2*h^4 - 3*a^2*c^6*e^4*h^2 -
3*b^2*c^6*d^4*g^2 + 2*a^2*c^6*e^3*g^3 - 2*a^2*c^6*d^3*h^3 + b^3*c^5*d^3*g^3
- 3*a^2*c^6*d^2*g^4 - a^4*b^2*c^2*h^3*k^3 - a^3*b^2*c^3*g^3*j^3 - a^2*b^4*

```
c^2*f^3*k^3 - a^2*b^2*c^4*f^3*h^3 + 2*a^7*c*k^3*m^3 + a^7*b*l^3*m^3 - 3*a^7
*c*j^2*m^4 + 6*a^3*c^5*f^5*m - 3*a^6*b^2*f*m^5 + 6*a^6*c^2*e*l^5 + 6*a^2*c^
6*e^5*l + b^7*c*d^3*l^3 + a*b^7*e^3*m^3 - 3*b^2*c^6*d^5*k + 6*a^5*c^3*d*k^5
 - 3*a*c^7*d^4*g^2 + 2*a*c^7*d^3*f^3 + b*c^7*d^3*e^3 - a^6*b^2*k^3*m^3 - a^
4*b^4*h^3*m^3 - a^2*b^6*f^3*m^3 - b^6*c^2*d^3*k^3 - b^4*c^4*d^3*h^3 - b^2*c
^6*d^3*f^3 - b^8*d^3*m^3 - a^6*c^2*k^6 - a^5*c^3*j^6 - a^4*c^4*h^6 - a^3*c^
5*g^6 - a^2*c^6*f^6 - a^7*c*l^6 - a*c^7*e^6 - a^8*m^6 - c^8*d^6, z, k1)*(ro
ot(34992*a^4*b^2*c^8*z^6 - 8748*a^3*b^4*c^7*z^6 + 729*a^2*b^6*c^6*z^6 - 466
56*a^5*c^9*z^6 + 34992*a^4*b^3*c^6*m*z^5 - 8748*a^3*b^5*c^5*m*z^5 + 729*a^2
*b^7*c^4*m*z^5 - 34992*a^4*b^2*c^7*j*z^5 + 8748*a^3*b^4*c^6*j*z^5 - 729*a^2
*b^6*c^5*j*z^5 - 46656*a^5*b*c^7*m*z^5 + 46656*a^5*c^8*j*z^5 + 34992*a^5*b*
c^6*j*m*z^4 - 11664*a^5*b*c^6*k*l*z^4 + 3888*a^4*b*c^7*f*j*z^4 + 3888*a^4*b
*c^7*e*k*z^4 + 3888*a^4*b*c^7*d*l*z^4 + 3888*a^4*b*c^7*g*h*z^4 + 3888*a^3*b
*c^8*d*e*z^4 + 243*a*b^5*c^6*d*e*z^4 - 25272*a^4*b^3*c^5*j*m*z^4 + 9720*a^4
*b^3*c^5*k*l*z^4 + 6075*a^3*b^5*c^4*j*m*z^4 - 2673*a^3*b^5*c^4*k*l*z^4 - 48
6*a^2*b^7*c^3*j*m*z^4 + 243*a^2*b^7*c^3*k*l*z^4 - 7776*a^4*b^2*c^6*h*k*z^4
- 7776*a^4*b^2*c^6*g*l*z^4 - 7776*a^4*b^2*c^6*f*m*z^4 + 2430*a^3*b^4*c^5*h*
k*z^4 + 2430*a^3*b^4*c^5*g*l*z^4 + 2430*a^3*b^4*c^5*f*m*z^4 - 243*a^2*b^6*c
^4*h*k*z^4 - 243*a^2*b^6*c^4*g*l*z^4 - 243*a^2*b^6*c^4*f*m*z^4 - 1944*a^3*b
^3*c^6*f*j*z^4 - 1944*a^3*b^3*c^6*e*k*z^4 - 1944*a^3*b^3*c^6*d*l*z^4 + 243*
a^2*b^5*c^5*f*j*z^4 + 243*a^2*b^5*c^5*e*k*z^4 + 243*a^2*b^5*c^5*d*l*z^4 - 1
944*a^3*b^3*c^6*g*h*z^4 + 243*a^2*b^5*c^5*g*h*z^4 + 3888*a^3*b^2*c^7*e*g*z^
4 + 3888*a^3*b^2*c^7*d*h*z^4 - 486*a^2*b^4*c^6*e*g*z^4 - 486*a^2*b^4*c^6*d*
h*z^4 - 1944*a^2*b^3*c^7*d*e*z^4 + 7776*a^5*c^7*h*k*z^4 + 7776*a^5*c^7*g*l*
z^4 + 7776*a^5*c^7*f*m*z^4 - 7776*a^4*c^8*e*g*z^4 - 7776*a^4*c^8*d*h*z^4 -
13608*a^5*b^2*c^5*m^2*z^4 + 11421*a^4*b^4*c^4*m^2*z^4 - 2916*a^3*b^6*c^3*m^
2*z^4 + 243*a^2*b^8*c^2*m^2*z^4 + 13608*a^4*b^2*c^6*j^2*z^4 - 3159*a^3*b^4*
c^5*j^2*z^4 + 243*a^2*b^6*c^4*j^2*z^4 + 1944*a^3*b^2*c^7*f^2*z^4 - 243*a^2*
b^4*c^6*f^2*z^4 - 3888*a^6*c^6*m^2*z^4 - 19440*a^5*c^7*j^2*z^4 - 3888*a^4*c
^8*f^2*z^4 + 3078*a^4*b^4*c^3*k*l*m*z^3 - 2592*a^5*b^2*c^4*k*l*m*z^3 - 891*
a^3*b^6*c^2*k*l*m*z^3 - 4536*a^4*b^3*c^4*j*k*l*z^3 + 1053*a^3*b^5*c^3*j*k*l
*z^3 - 81*a^2*b^7*c^2*j*k*l*z^3 - 2592*a^4*b^3*c^4*h*k*m*z^3 - 2592*a^4*b^3
*c^4*g*l*m*z^3 + 810*a^3*b^5*c^3*h*k*m*z^3 + 810*a^3*b^5*c^3*g*l*m*z^3 - 81
*a^2*b^7*c^2*h*k*m*z^3 - 81*a^2*b^7*c^2*g*l*m*z^3 + 7776*a^4*b^2*c^5*f*j*m*
z^3 + 3888*a^4*b^2*c^5*h*j*k*z^3 + 3888*a^4*b^2*c^5*g*j*l*z^3 - 3888*a^4*b^
2*c^5*f*k*l*z^3 - 2916*a^3*b^4*c^4*f*j*m*z^3 + 1458*a^3*b^4*c^4*f*k*l*z^3 -
 972*a^3*b^4*c^4*h*j*k*z^3 - 972*a^3*b^4*c^4*g*j*l*z^3 - 486*a^3*b^4*c^4*e*
k*m*z^3 - 486*a^3*b^4*c^4*d*l*m*z^3 + 324*a^2*b^6*c^3*f*j*m*z^3 - 162*a^2*b
^6*c^3*f*k*l*z^3 + 81*a^2*b^6*c^3*h*j*k*z^3 + 81*a^2*b^6*c^3*g*j*l*z^3 + 81
*a^2*b^6*c^3*e*k*m*z^3 + 81*a^2*b^6*c^3*d*l*m*z^3 - 486*a^3*b^4*c^4*g*h*m*z
^3 + 81*a^2*b^6*c^3*g*h*m*z^3 + 648*a^3*b^3*c^5*e*j*k*z^3 + 648*a^3*b^3*c^5
*d*j*l*z^3 - 81*a^2*b^5*c^4*e*j*k*z^3 - 81*a^2*b^5*c^4*d*j*l*z^3 + 2592*a^3
*b^3*c^5*e*g*m*z^3 + 2592*a^3*b^3*c^5*d*h*m*z^3 - 1296*a^3*b^3*c^5*f*h*k*z^
3 - 1296*a^3*b^3*c^5*f*g*l*z^3 - 1296*a^3*b^3*c^5*e*h*l*z^3 + 648*a^3*b^3*c
^5*g*h*j*z^3 - 324*a^2*b^5*c^4*e*g*m*z^3 - 324*a^2*b^5*c^4*d*h*m*z^3 + 162*
```

a^2*b^5*c^4*f*h*k*z^3 + 162*a^2*b^5*c^4*f*g*l*z^3 + 162*a^2*b^5*c^4*e*h*l*z
^3 - 81*a^2*b^5*c^4*g*h*j*z^3 + 5184*a^3*b^2*c^6*d*e*m*z^3 - 2592*a^3*b^2*c
^6*e*g*j*z^3 - 2592*a^3*b^2*c^6*d*h*j*z^3 - 2106*a^2*b^4*c^5*d*e*m*z^3 + 12
96*a^3*b^2*c^6*e*f*k*z^3 + 1296*a^3*b^2*c^6*d*g*k*z^3 + 1296*a^3*b^2*c^6*d*
f*l*z^3 + 324*a^2*b^4*c^5*e*g*j*z^3 + 324*a^2*b^4*c^5*d*h*j*z^3 - 162*a^2*b
^4*c^5*e*f*k*z^3 - 162*a^2*b^4*c^5*d*g*k*z^3 - 162*a^2*b^4*c^5*d*f*l*z^3 +
1296*a^3*b^2*c^6*f*g*h*z^3 - 162*a^2*b^4*c^5*f*g*h*z^3 + 1944*a^2*b^3*c^6*d
*e*j*z^3 - 1296*a^2*b^2*c^7*d*e*f*z^3 + 81*a^2*b^8*c*k*l*m*z^3 + 6480*a^5*b
*c^5*j*k*l*z^3 + 2592*a^5*b*c^5*h*k*m*z^3 + 2592*a^5*b*c^5*g*l*m*z^3 - 1296
*a^4*b*c^6*e*j*k*z^3 - 1296*a^4*b*c^6*d*j*l*z^3 - 5184*a^4*b*c^6*e*g*m*z^3
- 5184*a^4*b*c^6*d*h*m*z^3 + 2592*a^4*b*c^6*f*h*k*z^3 + 2592*a^4*b*c^6*f*g*
l*z^3 + 2592*a^4*b*c^6*e*h*l*z^3 - 1296*a^4*b*c^6*g*h*j*z^3 + 243*a*b^6*c^4
*d*e*m*z^3 - 3888*a^3*b*c^7*d*e*j*z^3 - 243*a*b^5*c^5*d*e*j*z^3 + 162*a*b^4
*c^6*d*e*f*z^3 - 2592*a^6*c^5*k*l*m*z^3 - 5184*a^5*c^6*h*j*k*z^3 - 5184*a^5
*c^6*g*j*l*z^3 - 5184*a^5*c^6*f*j*m*z^3 + 2592*a^5*c^6*f*k*l*z^3 + 2592*a^5
*c^6*e*k*m*z^3 + 2592*a^5*c^6*d*l*m*z^3 + 2592*a^5*c^6*g*h*m*z^3 + 5184*a^4
*c^7*e*g*j*z^3 + 5184*a^4*c^7*d*h*j*z^3 - 2592*a^4*c^7*e*f*k*z^3 - 2592*a^4
*c^7*d*g*k*z^3 - 2592*a^4*c^7*d*f*l*z^3 - 2592*a^4*c^7*d*e*m*z^3 - 2592*a^4
*c^7*f*g*h*z^3 + 2592*a^3*c^8*d*e*f*z^3 + 6480*a^5*b^2*c^4*j*m^2*z^3 + 6480
*a^4*b^3*c^4*j^2*m*z^3 - 5022*a^4*b^4*c^3*j*m^2*z^3 - 1296*a^3*b^5*c^3*j^2*
m*z^3 + 1134*a^3*b^6*c^2*j*m^2*z^3 + 81*a^2*b^7*c^2*j^2*m*z^3 + 2592*a^4*b^
3*c^4*h*l^2*z^3 - 1944*a^4*b^2*c^5*h^2*l*z^3 - 810*a^3*b^5*c^3*h*l^2*z^3 +
729*a^3*b^4*c^4*h^2*l*z^3 + 81*a^2*b^7*c^2*h*l^2*z^3 - 81*a^2*b^6*c^3*h^2*l
*z^3 - 5184*a^4*b^3*c^4*f*m^2*z^3 + 1620*a^3*b^5*c^3*f*m^2*z^3 + 1296*a^3*b
^3*c^5*f^2*m*z^3 - 162*a^2*b^7*c^2*f*m^2*z^3 - 162*a^2*b^5*c^4*f^2*m*z^3 -
1944*a^4*b^2*c^5*g*k^2*z^3 + 729*a^3*b^4*c^4*g*k^2*z^3 - 648*a^3*b^3*c^5*g^
2*k*z^3 - 81*a^2*b^6*c^3*g*k^2*z^3 + 81*a^2*b^5*c^4*g^2*k*z^3 - 1944*a^4*b^
2*c^5*e*l^2*z^3 + 729*a^3*b^4*c^4*e*l^2*z^3 + 648*a^3*b^2*c^6*e^2*l*z^3 - 8
1*a^2*b^6*c^3*e*l^2*z^3 - 81*a^2*b^4*c^5*e^2*l*z^3 + 1296*a^3*b^3*c^5*f*j^2
*z^3 - 1296*a^3*b^2*c^6*f^2*j*z^3 - 162*a^2*b^5*c^4*f*j^2*z^3 + 162*a^2*b^4
*c^5*f^2*j*z^3 - 648*a^3*b^3*c^5*d*k^2*z^3 + 81*a^2*b^5*c^4*d*k^2*z^3 + 648
*a^3*b^2*c^6*e*h^2*z^3 - 81*a^2*b^4*c^5*e*h^2*z^3 - 648*a^2*b^2*c^7*d^2*g*z
^3 - 10368*a^5*b*c^5*j^2*m*z^3 - 81*a^2*b^8*c*j*m^2*z^3 - 2592*a^5*b*c^5*h*
l^2*z^3 + 5184*a^5*b*c^5*f*m^2*z^3 - 2592*a^4*b*c^6*f^2*m*z^3 + 1296*a^4*b*
c^6*g^2*k*z^3 - 2592*a^4*b*c^6*f*j^2*z^3 + 1296*a^4*b*c^6*d*k^2*z^3 + 81*a*
b^4*c^6*d^2*g*z^3 + 2592*a^6*c^5*j*m^2*z^3 + 1296*a^5*c^6*h^2*l*z^3 + 1296*
a^5*c^6*g*k^2*z^3 + 1296*a^5*c^6*e*l^2*z^3 - 1296*a^4*c^7*e^2*l*z^3 + 2592*
a^4*c^7*f^2*j*z^3 - 2592*a^6*b*c^4*m^3*z^3 - 324*a^3*b^7*c*m^3*z^3 - 27*a^2
*b^8*c*l^3*z^3 - 1296*a^4*c^7*e*h^2*z^3 - 864*a^5*b*c^5*k^3*z^3 + 1296*a^3*
c^8*d^2*g*z^3 + 432*a^4*b*c^6*h^3*z^3 + 27*a*b^4*c^6*e^3*z^3 - 432*a^2*b*c^
8*d^3*z^3 + 216*a*b^3*c^7*d^3*z^3 + 1134*a^4*b^5*c^2*m^3*z^3 - 432*a^5*b^3*
c^3*m^3*z^3 + 1512*a^5*b^2*c^4*l^3*z^3 - 1107*a^4*b^4*c^3*l^3*z^3 + 297*a^3
*b^6*c^2*l^3*z^3 + 864*a^4*b^3*c^4*k^3*z^3 - 270*a^3*b^5*c^3*k^3*z^3 + 27*a
^2*b^7*c^2*k^3*z^3 - 2592*a^4*b^2*c^5*j^3*z^3 + 486*a^3*b^4*c^4*j^3*z^3 - 2
7*a^2*b^6*c^3*j^3*z^3 - 216*a^3*b^3*c^5*h^3*z^3 + 27*a^2*b^5*c^4*h^3*z^3 +

216*a^3*b^2*c^6*g^3*z^3 - 27*a^2*b^4*c^5*g^3*z^3 - 216*a^2*b^2*c^7*e^3*z^3
- 432*a^6*c^5*l^3*z^3 + 27*a^2*b^9*m^3*z^3 + 4320*a^5*c^6*j^3*z^3 - 432*a^4
*c^7*g^3*z^3 + 432*a^3*c^8*e^3*z^3 - 27*b^5*c^6*d^3*z^3 + 81*a^3*b^6*c*j*k*
l*m*z^2 - 1296*a^5*b*c^4*h*j*k*m*z^2 - 1296*a^5*b*c^4*g*j*l*m*z^2 + 1296*a^
5*b*c^4*f*k*l*m*z^2 - 81*a^2*b^7*c*f*k*l*m*z^2 + 2592*a^4*b*c^5*e*g*j*m*z^2
 + 2592*a^4*b*c^5*d*h*j*m*z^2 - 1296*a^4*b*c^5*f*h*j*k*z^2 - 1296*a^4*b*c^5
*f*g*j*l*z^2 - 1296*a^4*b*c^5*e*f*k*m*z^2 - 1296*a^4*b*c^5*d*f*l*m*z^2 - 64
8*a^4*b*c^5*e*h*j*l*z^2 - 648*a^4*b*c^5*e*g*k*l*z^2 - 648*a^4*b*c^5*d*h*k*l
*z^2 - 648*a^4*b*c^5*d*g*k*m*z^2 - 1296*a^4*b*c^5*f*g*h*m*z^2 - 162*a*b^6*c
^3*d*e*j*m*z^2 + 81*a*b^6*c^3*d*e*k*l*z^2 + 1296*a^3*b*c^6*d*e*f*m*z^2 - 64
8*a^3*b*c^6*d*f*g*k*z^2 - 648*a^3*b*c^6*d*e*h*k*z^2 - 648*a^3*b*c^6*d*e*g*l
*z^2 - 81*a*b^5*c^4*d*e*h*k*z^2 - 81*a*b^5*c^4*d*e*g*l*z^2 + 81*a*b^5*c^4*d
*e*f*m*z^2 - 81*a*b^4*c^5*d*e*f*j*z^2 + 81*a*b^4*c^5*d*e*g*h*z^2 + 648*a^5*
b^2*c^3*j*k*l*m*z^2 - 567*a^4*b^4*c^2*j*k*l*m*z^2 - 1944*a^4*b^3*c^3*f*k*l*
m*z^2 + 729*a^3*b^5*c^2*f*k*l*m*z^2 + 648*a^4*b^3*c^3*h*j*k*m*z^2 + 648*a^4
*b^3*c^3*g*j*l*m*z^2 - 81*a^3*b^5*c^2*h*j*k*m*z^2 - 81*a^3*b^5*c^2*g*j*l*m*
z^2 + 1944*a^4*b^2*c^4*f*j*k*l*z^2 - 729*a^3*b^4*c^3*f*j*k*l*z^2 + 648*a^4*
b^2*c^4*e*j*k*m*z^2 + 648*a^4*b^2*c^4*d*j*l*m*z^2 - 81*a^3*b^4*c^3*e*j*k*m*
z^2 - 81*a^3*b^4*c^3*d*j*l*m*z^2 + 81*a^2*b^6*c^2*f*j*k*l*z^2 + 1296*a^4*b^
2*c^4*f*h*k*m*z^2 + 1296*a^4*b^2*c^4*f*g*l*m*z^2 + 648*a^4*b^2*c^4*g*h*j*m*
z^2 - 648*a^3*b^4*c^3*f*h*k*m*z^2 - 648*a^3*b^4*c^3*f*g*l*m*z^2 - 324*a^4*b
^2*c^4*g*h*k*l*z^2 - 324*a^4*b^2*c^4*e*h*l*m*z^2 + 81*a^3*b^4*c^3*g*h*k*l*z
^2 - 81*a^3*b^4*c^3*g*h*j*m*z^2 + 81*a^2*b^6*c^2*f*h*k*m*z^2 + 81*a^2*b^6*c
^2*f*g*l*m*z^2 - 1296*a^3*b^3*c^4*e*g*j*m*z^2 - 1296*a^3*b^3*c^4*d*h*j*m*z^
2 + 648*a^3*b^3*c^4*f*h*j*k*z^2 + 648*a^3*b^3*c^4*f*g*j*l*z^2 + 648*a^3*b^3
*c^4*e*f*k*m*z^2 + 648*a^3*b^3*c^4*d*f*l*m*z^2 + 486*a^3*b^3*c^4*e*g*k*l*z^
2 + 486*a^3*b^3*c^4*d*h*k*l*z^2 + 162*a^3*b^3*c^4*e*h*j*l*z^2 + 162*a^3*b^3
*c^4*d*g*k*m*z^2 + 162*a^2*b^5*c^3*e*g*j*m*z^2 + 162*a^2*b^5*c^3*d*h*j*m*z^
2 - 81*a^2*b^5*c^3*f*h*j*k*z^2 - 81*a^2*b^5*c^3*f*g*j*l*z^2 - 81*a^2*b^5*c^
3*e*g*k*l*z^2 - 81*a^2*b^5*c^3*e*f*k*m*z^2 - 81*a^2*b^5*c^3*d*h*k*l*z^2 - 8
1*a^2*b^5*c^3*d*f*l*m*z^2 + 648*a^3*b^3*c^4*f*g*h*m*z^2 - 81*a^2*b^5*c^3*f*
g*h*m*z^2 - 3240*a^3*b^2*c^5*d*e*j*m*z^2 + 1620*a^3*b^2*c^5*d*e*k*l*z^2 + 1
377*a^2*b^4*c^4*d*e*j*m*z^2 - 648*a^3*b^2*c^5*e*f*j*k*z^2 - 648*a^3*b^2*c^5
*d*f*j*l*z^2 - 648*a^2*b^4*c^4*d*e*k*l*z^2 - 324*a^3*b^2*c^5*d*g*j*k*z^2 +
81*a^2*b^4*c^4*e*f*j*k*z^2 + 81*a^2*b^4*c^4*d*f*j*l*z^2 + 972*a^3*b^2*c^5*e
*f*h*l*z^2 - 648*a^3*b^2*c^5*f*g*h*j*z^2 - 324*a^3*b^2*c^5*e*g*h*k*z^2 - 32
4*a^3*b^2*c^5*d*g*h*l*z^2 - 162*a^2*b^4*c^4*e*f*h*l*z^2 + 81*a^2*b^4*c^4*f*
g*h*j*z^2 + 81*a^2*b^4*c^4*e*g*h*k*z^2 + 81*a^2*b^4*c^4*d*g*h*l*z^2 - 648*a
^2*b^3*c^5*d*e*f*m*z^2 + 486*a^2*b^3*c^5*d*e*h*k*z^2 + 486*a^2*b^3*c^5*d*e*
g*l*z^2 + 162*a^2*b^3*c^5*d*f*g*k*z^2 + 648*a^2*b^2*c^6*d*e*f*j*z^2 - 324*a
^2*b^2*c^6*d*e*g*h*z^2 - 1296*a^6*b*c^3*k*l*m^2*z^2 - 81*a^4*b^5*c*k*l*m^2*
z^2 - 1296*a^5*b*c^4*j^2*k*l*z^2 - 324*a^5*b*c^4*h^2*l*m*z^2 + 324*a^5*b*c^
4*h*k^2*l*z^2 - 324*a^5*b*c^4*g*k^2*m*z^2 + 972*a^5*b*c^4*h*j*l^2*z^2 + 324
*a^5*b*c^4*g*k*l^2*z^2 - 324*a^5*b*c^4*e*l^2*m*z^2 - 324*a^4*b*c^5*e^2*l*m*
z^2 - 1944*a^5*b*c^4*f*j*m^2*z^2 + 1296*a^5*b*c^4*e*k*m^2*z^2 + 1296*a^5*b*

c^4*d*l*m^2*z^2 + 648*a^4*b*c^5*f^2*j*m*z^2 + 81*a^2*b^7*c*f*j*m^2*z^2 + 12
96*a^5*b*c^4*g*h*m^2*z^2 - 324*a^4*b*c^5*g^2*j*k*z^2 + 324*a^4*b*c^5*g^2*h*
l*z^2 + 972*a^4*b*c^5*f*h^2*l*z^2 + 324*a^4*b*c^5*g*h^2*k*z^2 - 324*a^4*b*c
^5*e*h^2*m*z^2 - 324*a^4*b*c^5*d*j*k^2*z^2 - 324*a^3*b*c^6*d^2*j*k*z^2 + 97
2*a^4*b*c^5*f*g*k^2*z^2 + 972*a^3*b*c^6*d^2*g*m*z^2 + 324*a^4*b*c^5*e*h*k^2
*z^2 + 324*a^3*b*c^6*d^2*h*l*z^2 + 81*a*b^5*c^4*d^2*g*m*z^2 + 972*a^4*b*c^5
*e*f*l^2*z^2 + 324*a^4*b*c^5*d*g*l^2*z^2 - 324*a^3*b*c^6*e^2*h*j*z^2 + 324*
a^3*b*c^6*e^2*g*k*z^2 - 324*a^3*b*c^6*e^2*f*l*z^2 - 1296*a^4*b*c^5*d*e*m^2*
z^2 + 81*a*b^7*c^2*d*e*m^2*z^2 - 324*a^3*b*c^6*d*g^2*j*z^2 - 81*a*b^4*c^5*d
^2*g*j*z^2 + 81*a*b^4*c^5*d^2*e*l*z^2 + 324*a^3*b*c^6*e*g^2*h*z^2 + 81*a*b^
4*c^5*d*e^2*k*z^2 + 1296*a^3*b*c^6*d*e*j^2*z^2 - 324*a^3*b*c^6*e*f*h^2*z^2
+ 324*a^3*b*c^6*d*g*h^2*z^2 + 81*a*b^5*c^4*d*e*j^2*z^2 - 324*a^2*b*c^7*d^2*
f*g*z^2 + 324*a^2*b*c^7*d^2*e*h*z^2 + 81*a*b^3*c^6*d^2*f*g*z^2 - 81*a*b^3*c
^6*d^2*e*h*z^2 + 324*a^2*b*c^7*d*e^2*g*z^2 - 81*a*b^3*c^6*d*e^2*g*z^2 + 129
6*a^6*c^4*j*k*l*m*z^2 - 1296*a^5*c^5*f*j*k*l*z^2 - 1296*a^5*c^5*e*j*k*m*z^2
 - 1296*a^5*c^5*d*j*l*m*z^2 - 1296*a^5*c^5*g*h*j*m*z^2 + 1296*a^5*c^5*e*h*l
*m*z^2 + 1296*a^4*c^6*e*f*j*k*z^2 + 1296*a^4*c^6*d*g*j*k*z^2 + 1296*a^4*c^6
*d*f*j*l*z^2 - 1296*a^4*c^6*d*e*k*l*z^2 + 1296*a^4*c^6*d*e*j*m*z^2 + 1296*a
^4*c^6*f*g*h*j*z^2 - 1296*a^4*c^6*e*f*h*l*z^2 - 1296*a^3*c^7*d*e*f*j*z^2 +
648*a^5*b^3*c^2*k*l*m^2*z^2 + 648*a^4*b^3*c^3*j^2*k*l*z^2 + 486*a^5*b^2*c^3
*h*l^2*m*z^2 - 81*a^4*b^4*c^2*h*l^2*m*z^2 + 81*a^4*b^3*c^3*h^2*l*m*z^2 - 81
*a^3*b^5*c^2*j^2*k*l*z^2 - 162*a^4*b^2*c^4*g^2*k*m*z^2 - 81*a^4*b^3*c^3*h*k
^2*l*z^2 + 81*a^4*b^3*c^3*g*k^2*m*z^2 - 567*a^4*b^3*c^3*h*j*l^2*z^2 + 486*a
^4*b^2*c^4*h^2*j*l*z^2 - 81*a^4*b^3*c^3*g*k*l^2*z^2 + 81*a^4*b^3*c^3*e*l^2*
m*z^2 + 81*a^3*b^5*c^2*h*j*l^2*z^2 - 81*a^3*b^4*c^3*h^2*j*l*z^2 + 81*a^3*b^
3*c^4*e^2*l*m*z^2 + 2430*a^4*b^3*c^3*f*j*m^2*z^2 - 2268*a^4*b^2*c^4*f*j^2*m
*z^2 - 810*a^3*b^5*c^2*f*j*m^2*z^2 + 810*a^3*b^4*c^3*f*j^2*m*z^2 - 648*a^4*
b^3*c^3*e*k*m^2*z^2 - 648*a^4*b^3*c^3*d*l*m^2*z^2 - 648*a^4*b^2*c^4*h*j^2*k
*z^2 - 648*a^4*b^2*c^4*g*j^2*l*z^2 - 162*a^3*b^3*c^4*f^2*j*m*z^2 + 81*a^3*b
^5*c^2*e*k*m^2*z^2 + 81*a^3*b^5*c^2*d*l*m^2*z^2 + 81*a^3*b^4*c^3*h*j^2*k*z^
2 + 81*a^3*b^4*c^3*g*j^2*l*z^2 - 81*a^2*b^6*c^2*f*j^2*m*z^2 - 648*a^4*b^3*c
^3*g*h*m^2*z^2 + 486*a^4*b^2*c^4*g*j*k^2*z^2 - 486*a^4*b^2*c^4*e*k^2*l*z^2
+ 486*a^3*b^2*c^5*d^2*k*m*z^2 - 162*a^4*b^2*c^4*d*k^2*m*z^2 + 81*a^3*b^5*c^
2*g*h*m^2*z^2 - 81*a^3*b^4*c^3*g*j*k^2*z^2 + 81*a^3*b^4*c^3*e*k^2*l*z^2 + 8
1*a^3*b^3*c^4*g^2*j*k*z^2 - 81*a^2*b^4*c^4*d^2*k*m*z^2 + 486*a^4*b^2*c^4*e*
j*l^2*z^2 - 486*a^4*b^2*c^4*d*k*l^2*z^2 - 162*a^3*b^2*c^5*e^2*j*l*z^2 - 81*
a^3*b^4*c^3*e*j*l^2*z^2 + 81*a^3*b^4*c^3*d*k*l^2*z^2 - 81*a^3*b^3*c^4*g^2*h
*l*z^2 - 1458*a^4*b^2*c^4*f*h*l^2*z^2 + 648*a^3*b^4*c^3*f*h*l^2*z^2 - 567*a
^3*b^3*c^4*f*h^2*l*z^2 + 486*a^3*b^2*c^5*e^2*h*m*z^2 - 81*a^3*b^3*c^4*g*h^2
*k*z^2 + 81*a^3*b^3*c^4*e*h^2*m*z^2 - 81*a^2*b^6*c^2*f*h*l^2*z^2 + 81*a^2*b
^5*c^3*f*h^2*l*z^2 - 81*a^2*b^4*c^4*e^2*h*m*z^2 - 1296*a^4*b^2*c^4*e*g*m^2*
z^2 - 1296*a^4*b^2*c^4*d*h*m^2*z^2 + 648*a^3*b^4*c^3*e*g*m^2*z^2 + 648*a^3*
b^4*c^3*d*h*m^2*z^2 + 81*a^3*b^3*c^4*d*j*k^2*z^2 - 81*a^2*b^6*c^2*e*g*m^2*z
^2 - 81*a^2*b^6*c^2*d*h*m^2*z^2 + 81*a^2*b^3*c^5*d^2*j*k*z^2 - 567*a^3*b^3*
c^4*f*g*k^2*z^2 - 567*a^2*b^3*c^5*d^2*g*m*z^2 + 486*a^3*b^2*c^5*f*g^2*k*z^2

- 486*a^3*b^2*c^5*e*g^2*l*z^2 + 486*a^3*b^2*c^5*d*g^2*m*z^2 - 81*a^3*b^3*c
^4*e*h*k^2*z^2 + 81*a^2*b^5*c^3*f*g*k^2*z^2 - 81*a^2*b^4*c^4*f*g^2*k*z^2 +
81*a^2*b^4*c^4*e*g^2*l*z^2 - 81*a^2*b^4*c^4*d*g^2*m*z^2 - 81*a^2*b^3*c^5*d^
2*h*l*z^2 - 567*a^3*b^3*c^4*e*f*l^2*z^2 - 486*a^3*b^2*c^5*d*h^2*k*z^2 - 162
*a^3*b^2*c^5*e*h^2*j*z^2 - 81*a^3*b^3*c^4*d*g*l^2*z^2 + 81*a^2*b^5*c^3*e*f*
l^2*z^2 + 81*a^2*b^4*c^4*d*h^2*k*z^2 + 81*a^2*b^3*c^5*e^2*h*j*z^2 - 81*a^2*
b^3*c^5*e^2*g*k*z^2 + 81*a^2*b^3*c^5*e^2*f*l*z^2 + 1944*a^3*b^3*c^4*d*e*m^2
*z^2 - 729*a^2*b^5*c^3*d*e*m^2*z^2 + 648*a^3*b^2*c^5*e*g*j^2*z^2 + 648*a^3*
b^2*c^5*d*h*j^2*z^2 - 81*a^2*b^4*c^4*e*g*j^2*z^2 - 81*a^2*b^4*c^4*d*h*j^2*z
^2 + 486*a^3*b^2*c^5*d*f*k^2*z^2 + 486*a^2*b^2*c^6*d^2*g*j*z^2 - 486*a^2*b^
2*c^6*d^2*e*l*z^2 - 162*a^2*b^2*c^6*d^2*f*k*z^2 - 81*a^2*b^4*c^4*d*f*k^2*z^
2 + 81*a^2*b^3*c^5*d*g^2*j*z^2 - 486*a^2*b^2*c^6*d*e^2*k*z^2 - 81*a^2*b^3*c
^5*e*g^2*h*z^2 - 648*a^2*b^3*c^5*d*e*j^2*z^2 - 162*a^2*b^2*c^6*e^2*f*h*z^2
+ 81*a^2*b^3*c^5*e*f*h^2*z^2 - 81*a^2*b^3*c^5*d*g*h^2*z^2 - 162*a^2*b^2*c^6
*d*f*g^2*z^2 - 189*a^5*b^3*c^2*l^3*m*z^2 + 162*a^5*b^2*c^3*k^3*m*z^2 - 27*a
^4*b^4*c^2*k^3*m*z^2 - 702*a^4*b^3*c^3*j^3*m*z^2 - 81*a^3*b^6*c*j^2*m^2*z^2
+ 81*a^3*b^5*c^2*j^3*m*z^2 - 54*a^5*b^3*c^2*j*m^3*z^2 - 486*a^5*b^2*c^3*j*
l^3*z^2 + 216*a^4*b^4*c^2*j*l^3*z^2 - 189*a^4*b^3*c^3*j*k^3*z^2 - 54*a^4*b^
2*c^4*h^3*m*z^2 + 27*a^3*b^5*c^2*j*k^3*z^2 + 27*a^3*b^3*c^4*g^3*m*z^2 - 810
*a^4*b^4*c^2*f*m^3*z^2 + 540*a^5*b^2*c^3*f*m^3*z^2 - 324*a^3*b^2*c^5*f^3*m*
z^2 + 54*a^2*b^4*c^4*f^3*m*z^2 + 675*a^4*b^3*c^3*f*l^3*z^2 - 243*a^3*b^5*c^
2*f*l^3*z^2 - 189*a^2*b^3*c^5*e^3*m*z^2 + 27*a^3*b^3*c^4*h^3*j*z^2 - 486*a^
4*b^2*c^4*f*k^3*z^2 - 486*a^2*b^2*c^6*d^3*m*z^2 + 216*a^3*b^4*c^3*f*k^3*z^2
- 54*a^3*b^2*c^5*g^3*j*z^2 - 27*a^2*b^6*c^2*f*k^3*z^2 - 270*a^3*b^3*c^4*f*
j^3*z^2 - 54*a^2*b^3*c^5*f^3*j*z^2 + 27*a^2*b^5*c^3*f*j^3*z^2 + 162*a^2*b^2
*c^6*e^3*j*z^2 + 162*a^3*b^2*c^5*f*h^3*z^2 - 27*a^2*b^4*c^4*f*h^3*z^2 + 27*
a^2*b^3*c^5*f*g^3*z^2 + 81*a*b^2*c^7*d^2*e^2*z^2 - 648*a^6*c^4*h*l^2*m*z^2
+ 648*a^5*c^5*g^2*k*m*z^2 - 648*a^5*c^5*h^2*j*l*z^2 + 1296*a^5*c^5*h*j^2*k*
z^2 + 1296*a^5*c^5*g*j^2*l*z^2 + 1296*a^5*c^5*f*j^2*m*z^2 - 648*a^5*c^5*g*j
*k^2*z^2 + 648*a^5*c^5*e*k^2*l*z^2 + 648*a^5*c^5*d*k^2*m*z^2 - 648*a^4*c^6*
d^2*k*m*z^2 - 648*a^5*c^5*e*j*l^2*z^2 + 648*a^5*c^5*d*k*l^2*z^2 + 648*a^4*c
^6*e^2*j*l*z^2 + 324*a^6*b*c^3*l^3*m*z^2 + 27*a^4*b^5*c*l^3*m*z^2 + 648*a^5
*c^5*f*h*l^2*z^2 - 648*a^4*c^6*e^2*h*m*z^2 + 1512*a^5*b*c^4*j^3*m*z^2 + 108
0*a^6*b*c^3*j*m^3*z^2 - 162*a^4*b^5*c*j*m^3*z^2 - 648*a^4*c^6*f*g^2*k*z^2 +
648*a^4*c^6*e*g^2*l*z^2 - 648*a^4*c^6*d*g^2*m*z^2 - 27*a^3*b^6*c*j*l^3*z^2
+ 648*a^4*c^6*e*h^2*j*z^2 + 648*a^4*c^6*d*h^2*k*z^2 + 324*a^5*b*c^4*j*k^3*
z^2 - 1296*a^4*c^6*e*g*j^2*z^2 - 1296*a^4*c^6*d*h*j^2*z^2 - 108*a^4*b*c^5*g
^3*m*z^2 - 648*a^4*c^6*d*f*k^2*z^2 - 648*a^3*c^7*d^2*g*j*z^2 + 648*a^3*c^7*
d^2*f*k*z^2 + 648*a^3*c^7*d^2*e*l*z^2 + 270*a^3*b^6*c*f*m^3*z^2 + 648*a^3*c
^7*d*e^2*k*z^2 - 540*a^5*b*c^4*f*l^3*z^2 + 324*a^3*b*c^6*e^3*m*z^2 - 108*a^
4*b*c^5*h^3*j*z^2 + 27*a^2*b^7*c*f*l^3*z^2 + 27*a*b^5*c^4*e^3*m*z^2 + 648*a
^3*c^7*e^2*f*h*z^2 + 216*a*b^4*c^5*d^3*m*z^2 + 648*a^4*b*c^5*f*j^3*z^2 + 21
6*a^3*b*c^6*f^3*j*z^2 + 648*a^3*c^7*d*f*g^2*z^2 - 27*a*b^4*c^5*e^3*j*z^2 +
324*a^2*b*c^7*d^3*j*z^2 - 189*a*b^3*c^6*d^3*j*z^2 - 108*a^3*b*c^6*f*g^3*z^2
- 108*a^2*b*c^7*e^3*f*z^2 + 27*a*b^3*c^6*e^3*f*z^2 + 162*a*b^2*c^7*d^3*f*z

^2 - 1134*a^5*b^2*c^3*j^2*m^2*z^2 + 648*a^4*b^4*c^2*j^2*m^2*z^2 + 81*a^5*b^
2*c^3*k^2*l^2*z^2 + 162*a^4*b^2*c^4*f^2*m^2*z^2 + 81*a^4*b^2*c^4*h^2*k^2*z^
2 + 81*a^4*b^2*c^4*g^2*l^2*z^2 + 162*a^3*b^2*c^5*f^2*j^2*z^2 + 81*a^3*b^2*c
^5*e^2*k^2*z^2 + 81*a^3*b^2*c^5*d^2*l^2*z^2 + 81*a^3*b^2*c^5*g^2*h^2*z^2 +
81*a^2*b^2*c^6*e^2*g^2*z^2 + 81*a^2*b^2*c^6*d^2*h^2*z^2 - 216*a^6*c^4*k^3*m
*z^2 + 216*a^6*c^4*j*l^3*z^2 + 27*a^3*b^7*j*m^3*z^2 + 216*a^5*c^5*h^3*m*z^2
 + 432*a^6*c^4*f*m^3*z^2 + 432*a^4*c^6*f^3*m*z^2 - 27*b^6*c^4*d^3*m*z^2 - 2
7*a^2*b^8*f*m^3*z^2 + 216*a^5*c^5*f*k^3*z^2 + 216*a^4*c^6*g^3*j*z^2 + 216*a
^3*c^7*d^3*m*z^2 + 216*a^5*b^4*c*m^4*z^2 - 216*a^3*c^7*e^3*j*z^2 + 27*b^5*c
^5*d^3*j*z^2 - 216*a^4*c^6*f*h^3*z^2 - 27*b^4*c^6*d^3*f*z^2 - 216*a^2*c^8*d
^3*f*z^2 - 648*a^6*c^4*j^2*m^2*z^2 - 324*a^6*c^4*k^2*l^2*z^2 - 648*a^5*c^5*
f^2*m^2*z^2 - 324*a^5*c^5*h^2*k^2*z^2 - 324*a^5*c^5*g^2*l^2*z^2 - 648*a^4*c
^6*f^2*j^2*z^2 - 324*a^4*c^6*e^2*k^2*z^2 - 324*a^4*c^6*d^2*l^2*z^2 - 405*a^
6*b^2*c^2*m^4*z^2 - 324*a^4*c^6*g^2*h^2*z^2 - 324*a^3*c^7*e^2*g^2*z^2 - 324
*a^3*c^7*d^2*h^2*z^2 + 243*a^4*b^2*c^4*j^4*z^2 - 27*a^3*b^4*c^3*j^4*z^2 - 3
24*a^2*c^8*d^2*e^2*z^2 + 27*a^2*b^2*c^6*f^4*z^2 - 108*a^7*c^3*m^4*z^2 - 27*
a^4*b^6*m^4*z^2 - 540*a^5*c^5*j^4*z^2 - 108*a^3*c^7*f^4*z^2 - 216*a^5*b*c^3
*f*j*k*l*m*z - 54*a^3*b^5*c*f*j*k*l*m*z + 27*a^3*b^5*c*g*h*k*l*m*z - 27*a^2
*b^6*c*e*g*k*l*m*z - 27*a^2*b^6*c*d*h*k*l*m*z + 432*a^4*b*c^4*d*g*j*k*m*z -
 432*a^4*b*c^4*d*e*k*l*m*z + 216*a^4*b*c^4*e*g*j*k*l*z + 216*a^4*b*c^4*e*f*
j*k*m*z + 216*a^4*b*c^4*d*h*j*k*l*z + 216*a^4*b*c^4*d*f*j*l*m*z + 216*a^4*b
*c^4*f*g*h*j*m*z - 27*a*b^6*c^2*d*e*j*k*l*z - 27*a*b^6*c^2*d*e*h*k*m*z - 27
*a*b^6*c^2*d*e*g*l*m*z + 216*a^3*b*c^5*d*e*h*j*k*z + 216*a^3*b*c^5*d*e*g*j*
l*z - 216*a^3*b*c^5*d*e*f*j*m*z + 27*a*b^5*c^3*d*e*h*j*k*z + 27*a*b^5*c^3*d
*e*g*j*l*z + 27*a*b^5*c^3*d*e*g*h*m*z - 27*a*b^4*c^4*d*e*g*h*j*z + 27*a*b^7
*c*d*e*k*l*m*z + 270*a^4*b^3*c^2*f*j*k*l*m*z - 108*a^4*b^3*c^2*g*h*k*l*m*z
- 216*a^4*b^2*c^3*f*h*j*k*m*z - 216*a^4*b^2*c^3*f*g*j*l*m*z - 216*a^4*b^2*c
^3*e*g*k*l*m*z - 216*a^4*b^2*c^3*d*h*k*l*m*z + 162*a^3*b^4*c^2*e*g*k*l*m*z
+ 162*a^3*b^4*c^2*d*h*k*l*m*z + 108*a^4*b^2*c^3*g*h*j*k*l*z + 108*a^4*b^2*c
^3*e*h*j*l*m*z + 54*a^3*b^4*c^2*f*h*j*k*m*z + 54*a^3*b^4*c^2*f*g*j*l*m*z -
27*a^3*b^4*c^2*g*h*j*k*l*z + 540*a^3*b^3*c^3*d*e*k*l*m*z - 216*a^2*b^5*c^2*
d*e*k*l*m*z - 162*a^3*b^3*c^3*e*g*j*k*l*z - 162*a^3*b^3*c^3*d*h*j*k*l*z - 1
08*a^3*b^3*c^3*d*g*j*k*m*z - 54*a^3*b^3*c^3*e*f*j*k*m*z - 54*a^3*b^3*c^3*d*
f*j*l*m*z + 27*a^2*b^5*c^2*e*g*j*k*l*z + 27*a^2*b^5*c^2*d*h*j*k*l*z - 108*a
^3*b^3*c^3*e*g*h*k*m*z - 108*a^3*b^3*c^3*d*g*h*l*m*z - 54*a^3*b^3*c^3*f*g*h
*j*m*z + 27*a^2*b^5*c^2*e*g*h*k*m*z + 27*a^2*b^5*c^2*d*g*h*l*m*z - 540*a^3*
b^2*c^4*d*e*j*k*l*z + 216*a^2*b^4*c^3*d*e*j*k*l*z - 216*a^3*b^2*c^4*d*e*h*k
*m*z - 216*a^3*b^2*c^4*d*e*g*l*m*z + 162*a^2*b^4*c^3*d*e*h*k*m*z + 162*a^2*
b^4*c^3*d*e*g*l*m*z + 108*a^3*b^2*c^4*e*g*h*j*k*z - 108*a^3*b^2*c^4*e*f*h*j
*l*z + 108*a^3*b^2*c^4*d*g*h*j*l*z + 108*a^3*b^2*c^4*d*f*g*k*m*z - 27*a^2*b
^4*c^3*e*g*h*j*k*z - 27*a^2*b^4*c^3*d*g*h*j*l*z - 162*a^2*b^3*c^4*d*e*h*j*k
*z - 162*a^2*b^3*c^4*d*e*g*j*l*z + 54*a^2*b^3*c^4*d*e*f*j*m*z - 108*a^2*b^3
*c^4*d*e*g*h*m*z + 108*a^2*b^2*c^5*d*e*g*h*j*z + 324*a^6*b*c^2*j*k*l*m^2*z
- 81*a^5*b^3*c*j*k*l*m^2*z + 27*a^4*b^4*c*j^2*k*l*m*z - 27*a^4*b^4*c*h*k^2*
l*m*z - 27*a^4*b^4*c*g*k*l^2*m*z + 216*a^5*b*c^3*h*j^2*k*m*z + 216*a^5*b*c^

3*g*j^2*l*m*z + 54*a^4*b^4*c*f*k*l*m^2*z + 27*a^4*b^4*c*h*j*k*m^2*z + 27*a^
4*b^4*c*g*j*l*m^2*z + 27*a^2*b^6*c*f^2*k*l*m*z + 216*a^5*b*c^3*e*k^2*l*m*z
- 108*a^5*b*c^3*h*j*k^2*l*z + 27*a^3*b^5*c*e*k^2*l*m*z + 216*a^5*b*c^3*d*k*
l^2*m*z + 216*a^4*b*c^4*e^2*j*l*m*z - 108*a^5*b*c^3*g*j*k*l^2*z + 27*a^3*b^
5*c*d*k*l^2*m*z - 324*a^5*b*c^3*e*j*k*m^2*z - 324*a^5*b*c^3*d*j*l*m^2*z - 2
16*a^5*b*c^3*f*h*l^2*m*z - 108*a^4*b*c^4*f^2*j*k*l*z - 27*a^3*b^5*c*e*j*k*m
^2*z - 27*a^3*b^5*c*d*j*l*m^2*z - 324*a^5*b*c^3*g*h*j*m^2*z + 216*a^5*b*c^3
*f*h*k*m^2*z + 216*a^5*b*c^3*f*g*l*m^2*z + 216*a^5*b*c^3*e*h*l*m^2*z - 216*
a^4*b*c^4*f^2*h*k*m*z - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5*c*g*h*j*m^2*
z + 216*a^4*b*c^4*e*g^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 216*a^4*b*c^4*f
*h^2*j*l*z + 216*a^4*b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k*m*z - 108*a^
4*b*c^4*g*h^2*j*k*z - 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^4*d*h*j^2*m*z
 + 216*a^4*b*c^4*f*h*j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*a^2*b^6*c*e*g
*j*m^2*z + 27*a^2*b^6*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*z - 216*a^4*b
*c^4*f*g*j*k^2*z + 216*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d^2*e*l*m*z -
108*a^4*b*c^4*e*h*j*k^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^3*b*c^5*d^2*h
*j*l*z + 108*a^3*b*c^5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z + 27*a*b^5*c^
3*d^2*g*k*l*z + 27*a*b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j*l^2*z + 216*
a^3*b*c^5*d*e^2*k*m*z - 108*a^4*b*c^4*d*g*j*l^2*z - 108*a^3*b*c^5*e^2*g*j*k
*z + 27*a*b^5*c^3*d*e^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^2*z + 216*a^3*b*c^5*e
^2*f*h*m*z - 108*a^4*b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5*e^2*g*h*l*z + 108*a^
3*b*c^5*e*f^2*j*k*z + 108*a^3*b*c^5*d*f^2*j*l*z + 27*a*b^6*c^2*d*e*j^2*m*z
- 216*a^3*b*c^5*e*f^2*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*z - 27*a*b^4*c^4*d^2*
e*j*l*z + 216*a^3*b*c^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*g^2*h*j*z + 54*a*b^4*
c^4*d^2*f*g*m*z - 27*a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*c^4*d^2*e*h*m*z - 27*
a*b^4*c^4*d*e^2*j*k*z - 108*a^3*b*c^5*d*g*h^2*j*z + 54*a*b^4*c^4*d*e^2*h*l*
z + 27*a*b^6*c^2*d*e*h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*z - 27*a*b^4*c^4*d*e^
2*g*m*z - 27*a*b^4*c^4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2*f*g*j*z - 108*a^3*b*
c^5*d*e*g*k^2*z - 108*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*b*c^6*d^2*e*g*k*z - 5
4*a*b^3*c^5*d^2*f*g*j*z - 27*a*b^5*c^3*d*e*g*k^2*z + 27*a*b^4*c^4*d*e*g^2*k
*z + 27*a*b^3*c^5*d^2*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k*z - 108*a^2*b*c^6*d*
e^2*g*j*z + 27*a*b^3*c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d*e*f^2*j*z + 27*a*b^3
*c^5*d*e*f^2*j*z - 432*a^5*c^4*e*h*j*l*m*z + 432*a^4*c^5*d*e*j*k*l*z + 432*
a^4*c^5*e*f*h*j*l*z - 432*a^4*c^5*d*f*g*k*m*z - 27*a*b^7*c*d*e*j*m^2*z - 54
*a^5*b^2*c^2*j^2*k*l*m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z + 108*a^5*b^2*c^2*g*
k*l^2*m*z - 54*a^5*b^2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c^3*f^2*k*l*m*z - 270*
a^5*b^2*c^2*f*k*l*m^2*z - 189*a^3*b^4*c^2*f^2*k*l*m*z - 108*a^5*b^2*c^2*h*j
*k*m^2*z - 108*a^5*b^2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^2*h*j^2*k*m*z - 54*a^
4*b^3*c^2*g*j^2*l*m*z - 162*a^4*b^3*c^2*e*k^2*l*m*z + 54*a^4*b^2*c^3*g^2*j*
k*m*z + 27*a^4*b^3*c^2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d*k*l^2*m*z + 108*a^4*
b^2*c^3*g^2*h*l*m*z - 54*a^3*b^3*c^3*e^2*j*l*m*z + 27*a^4*b^3*c^2*g*j*k*l^2
*z - 27*a^3*b^4*c^2*g^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^2*k*l*z + 189*a^4*b^3
*c^2*e*j*k*m^2*z + 189*a^4*b^3*c^2*d*j*l*m^2*z - 162*a^4*b^2*c^3*e*j^2*k*m*
z - 162*a^4*b^2*c^3*d*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*j*k*l*z + 108*a^4*b^2
*c^3*g*h^2*k*m*z + 54*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4*b^2*c^3*f*h^2*l*m*z

+ 54*a^3*b^4*c^2*f*j^2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*m*z + 27*a^3*b^4*c^2*
e*j^2*k*m*z + 27*a^3*b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*c^2*f^2*j*k*l*z - 270
*a^3*b^2*c^4*d^2*j*k*m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z - 162*a^4*b^2*c^3*g*
h*j^2*m*z + 162*a^4*b^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*c^3*f^2*h*k*m*z + 162
*a^3*b^3*c^3*f^2*g*l*m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z - 54*a^4*b^3*c^2*f*g*
l*m^2*z - 54*a^4*b^3*c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*d*j*k^2*m*z + 54*a^2*
b^4*c^3*d^2*j*k*m*z + 27*a^3*b^4*c^2*g*h*j^2*m*z - 27*a^3*b^4*c^2*e*j*k^2*l
*z - 27*a^2*b^5*c^2*f^2*h*k*m*z - 27*a^2*b^5*c^2*f^2*g*l*m*z + 162*a^4*b^2*
c^3*d*j*k*l^2*z - 162*a^3*b^3*c^3*e*g^2*l*m*z + 108*a^4*b^2*c^3*e*h*k^2*m*z
 + 108*a^3*b^2*c^4*d^2*h*l*m*z - 54*a^4*b^2*c^3*f*g*k^2*m*z - 27*a^3*b^4*c^
2*e*h*k^2*m*z - 27*a^3*b^4*c^2*d*j*k*l^2*z + 27*a^3*b^3*c^3*g^2*h*j*l*z + 2
7*a^2*b^5*c^2*e*g^2*l*m*z - 27*a^2*b^4*c^3*d^2*h*l*m*z + 270*a^4*b^2*c^3*f*
h*j*l^2*z - 270*a^3*b^2*c^4*e^2*h*j*m*z - 162*a^4*b^2*c^3*e*h*k*l^2*z - 162
*a^3*b^3*c^3*d*h^2*k*m*z + 162*a^3*b^2*c^4*e^2*h*k*l*z + 108*a^4*b^2*c^3*d*
g*l^2*m*z + 108*a^3*b^2*c^4*e^2*g*k*m*z - 54*a^4*b^2*c^3*e*f*l^2*m*z - 54*a
^3*b^4*c^2*f*h*j*l^2*z + 54*a^3*b^3*c^3*f*h^2*j*l*z - 54*a^3*b^3*c^3*e*h^2*
j*m*z + 54*a^3*b^2*c^4*e^2*f*l*m*z + 54*a^2*b^4*c^3*e^2*h*j*m*z + 27*a^3*b^
4*c^2*e*h*k*l^2*z - 27*a^3*b^4*c^2*d*g*l^2*m*z + 27*a^3*b^3*c^3*g*h^2*j*k*z
 + 27*a^2*b^5*c^2*d*h^2*k*m*z - 27*a^2*b^4*c^3*e^2*h*k*l*z - 27*a^2*b^4*c^3
*e^2*g*k*m*z + 432*a^4*b^2*c^3*e*g*j*m^2*z + 432*a^4*b^2*c^3*d*h*j*m^2*z -
270*a^4*b^2*c^3*d*g*k*m^2*z - 216*a^3*b^4*c^2*e*g*j*m^2*z - 216*a^3*b^4*c^2
*d*h*j*m^2*z + 216*a^3*b^3*c^3*e*g*j^2*m*z + 216*a^3*b^3*c^3*d*h*j^2*m*z -
162*a^3*b^2*c^4*e*f^2*k*m*z - 162*a^3*b^2*c^4*d*f^2*l*m*z - 108*a^3*b^2*c^4
*f^2*h*j*k*z - 108*a^3*b^2*c^4*f^2*g*j*l*z + 54*a^4*b^2*c^3*e*f*k*m^2*z + 5
4*a^4*b^2*c^3*d*f*l*m^2*z + 54*a^3*b^4*c^2*d*g*k*m^2*z - 54*a^3*b^3*c^3*f*h
*j^2*k*z - 54*a^3*b^3*c^3*f*g*j^2*l*z - 27*a^2*b^5*c^2*e*g*j^2*m*z - 27*a^2
*b^5*c^2*d*h*j^2*m*z + 27*a^2*b^4*c^3*f^2*h*j*k*z + 27*a^2*b^4*c^3*f^2*g*j*
l*z + 27*a^2*b^4*c^3*e*f^2*k*m*z + 27*a^2*b^4*c^3*d*f^2*l*m*z + 324*a^2*b^3
*c^4*d^2*g*j*m*z - 270*a^3*b^2*c^4*d*g^2*j*m*z - 162*a^3*b^2*c^4*f^2*g*h*m*
z + 162*a^3*b^2*c^4*e*g^2*j*l*z - 162*a^2*b^3*c^4*d^2*e*l*m*z - 135*a^2*b^3
*c^4*d^2*g*k*l*z + 108*a^3*b^2*c^4*d*g^2*k*l*z + 54*a^4*b^2*c^3*f*g*h*m^2*z
 + 54*a^3*b^3*c^3*f*g*j*k^2*z - 54*a^3*b^2*c^4*f*g^2*j*k*z + 54*a^2*b^4*c^3
*d*g^2*j*m*z - 54*a^2*b^3*c^4*d^2*f*k*m*z + 27*a^3*b^3*c^3*e*h*j*k^2*z + 27
*a^3*b^3*c^3*d*g*k^2*l*z + 27*a^2*b^4*c^3*f^2*g*h*m*z - 27*a^2*b^4*c^3*e*g^
2*j*l*z - 27*a^2*b^4*c^3*d*g^2*k*l*z + 27*a^2*b^3*c^4*d^2*h*j*l*z + 162*a^3
*b^2*c^4*d*h^2*j*k*z - 162*a^2*b^3*c^4*d*e^2*k*m*z + 108*a^3*b^2*c^4*e*g^2*
h*m*z + 54*a^3*b^3*c^3*e*f*j*l^2*z + 27*a^3*b^3*c^3*d*g*j*l^2*z - 27*a^2*b^
4*c^3*e*g^2*h*m*z - 27*a^2*b^4*c^3*d*h^2*j*k*z + 27*a^2*b^3*c^4*e^2*g*j*k*z
 - 621*a^3*b^3*c^3*d*e*j*m^2*z + 594*a^3*b^2*c^4*d*e*j^2*m*z + 243*a^2*b^5*
c^2*d*e*j*m^2*z - 243*a^2*b^4*c^3*d*e*j^2*m*z + 135*a^3*b^3*c^3*e*g*h*l^2*z
 - 108*a^3*b^2*c^4*e*g*h^2*l*z + 108*a^3*b^2*c^4*d*g*h^2*m*z + 54*a^3*b^2*c
^4*e*f*j^2*k*z + 54*a^3*b^2*c^4*e*f*h^2*m*z + 54*a^3*b^2*c^4*d*g*j^2*k*z +
54*a^3*b^2*c^4*d*f*j^2*l*z - 54*a^2*b^3*c^4*e^2*f*h*m*z - 27*a^2*b^5*c^2*e*
g*h*l^2*z + 27*a^2*b^4*c^3*e*g*h^2*l*z - 27*a^2*b^4*c^3*d*g*h^2*m*z - 27*a^
2*b^3*c^4*e^2*g*h*l*z - 27*a^2*b^3*c^4*e*f^2*j*k*z - 27*a^2*b^3*c^4*d*f^2*j

*l*z + 162*a^2*b^2*c^5*d^2*e*j*l*z + 54*a^3*b^2*c^4*f*g*h*j^2*z - 54*a^3*b^
2*c^4*d*f*j*k^2*z + 54*a^2*b^3*c^4*e*f^2*h*l*z + 54*a^2*b^2*c^5*d^2*f*j*k*z
 - 27*a^2*b^3*c^4*f^2*g*h*j*z - 270*a^2*b^2*c^5*d^2*f*g*m*z - 162*a^3*b^2*c
^4*d*g*h*k^2*z + 162*a^2*b^2*c^5*d^2*g*h*k*z + 162*a^2*b^2*c^5*d*e^2*j*k*z
+ 108*a^2*b^2*c^5*d^2*e*h*m*z - 54*a^2*b^3*c^4*d*f*g^2*m*z + 27*a^2*b^4*c^3
*d*g*h*k^2*z + 27*a^2*b^3*c^4*e*g^2*h*j*z + 270*a^3*b^2*c^4*d*e*h*l^2*z - 2
70*a^2*b^2*c^5*d*e^2*h*l*z - 162*a^2*b^4*c^3*d*e*h*l^2*z + 108*a^2*b^3*c^4*
d*e*h^2*l*z + 108*a^2*b^2*c^5*d*e^2*g*m*z + 54*a^2*b^2*c^5*e^2*f*h*j*z + 27
*a^2*b^3*c^4*d*g*h^2*j*z + 162*a^2*b^2*c^5*d*e*f^2*m*z - 54*a^3*b^2*c^4*d*e
*f*m^2*z - 54*a^2*b^2*c^5*d*f^2*g*k*z + 135*a^2*b^3*c^4*d*e*g*k^2*z - 108*a
^2*b^2*c^5*d*e*g^2*k*z + 54*a^2*b^2*c^5*d*f*g^2*j*z - 54*a^2*b^2*c^5*d*e*f*
j^2*z - 9*a*b^7*c*d*e*l^3*z - 36*a*b*c^7*d^3*e*g*z - 108*a^6*b*c^2*k^2*l^2*
m*z + 27*a^5*b^3*c*k^2*l^2*m*z - 18*a^5*b^2*c^2*j*k^3*m*z - 27*a^4*b^3*c^2*
j^3*k*l*z - 108*a^5*b*c^3*h^2*k^2*m*z - 108*a^5*b*c^3*g^2*l^2*m*z + 108*a^5
*b*c^3*h^2*k*l^2*z + 108*a^5*b*c^3*g^2*k*m^2*z + 90*a^5*b^2*c^2*f*l^3*m*z -
 18*a^5*b^2*c^2*h*k*l^3*z + 18*a^4*b^2*c^3*h^3*k*l*z + 18*a^4*b^2*c^3*h^3*j
*m*z - 108*a^5*b*c^3*h*j^2*l^2*z + 18*a^4*b^3*c^2*f*k^3*m*z - 18*a^3*b^3*c^
3*g^3*j*m*z - 9*a^4*b^3*c^2*g*k^3*l*z + 9*a^3*b^3*c^3*g^3*k*l*z + 252*a^4*b
^2*c^3*f*j^3*m*z + 216*a^5*b*c^3*f*j^2*m^2*z + 180*a^3*b^2*c^4*f^3*j*m*z -
108*a^4*b*c^4*e^2*k^2*m*z - 108*a^4*b*c^4*d^2*l^2*m*z + 90*a^5*b^2*c^2*e*k*
m^3*z + 90*a^5*b^2*c^2*d*l*m^3*z - 90*a^3*b^2*c^4*f^3*k*l*z + 54*a^3*b^5*c*
f*j^2*m^2*z - 54*a^3*b^4*c^2*f*j^3*m*z + 36*a^5*b^2*c^2*f*j*m^3*z + 36*a^4*
b^2*c^3*h*j^3*k*z + 36*a^4*b^2*c^3*g*j^3*l*z - 36*a^2*b^4*c^3*f^3*j*m*z - 2
7*a^2*b^6*c*f^2*j*m^2*z + 18*a^2*b^4*c^3*f^3*k*l*z - 216*a^4*b*c^4*d^2*k*m^
2*z + 108*a^5*b*c^3*d*k^2*m^2*z - 108*a^4*b^3*c^2*f*j*l^3*z - 108*a^4*b*c^4
*g^2*h^2*m*z + 108*a^2*b^3*c^4*e^3*j*m*z + 90*a^5*b^2*c^2*g*h*m^3*z + 54*a^
4*b^3*c^2*e*k*l^3*z - 54*a^2*b^3*c^4*e^3*k*l*z + 234*a^2*b^2*c^5*d^3*j*m*z
- 144*a^2*b^2*c^5*d^3*k*l*z + 90*a^4*b^2*c^3*f*j*k^3*z - 72*a^4*b^2*c^3*d*k
^3*l*z + 27*a^4*b^3*c^2*g*h*l^3*z - 27*a^3*b^3*c^3*g*h^3*l*z - 18*a^3*b^4*c
^2*f*j*k^3*z + 9*a^3*b^4*c^2*d*k^3*l*z + 216*a^4*b*c^4*f^2*h*l^2*z - 216*a^
4*b*c^4*e^2*h*m^2*z + 108*a^4*b*c^4*g^2*h*k^2*z - 18*a^4*b^2*c^3*g*h*k^3*z
+ 18*a^3*b^2*c^4*g^3*h*k*z + 18*a^3*b^2*c^4*f*g^3*m*z + 9*a^3*b^4*c^2*g*h*k
^3*z - 9*a^3*b^3*c^3*e*j^3*k*z - 9*a^3*b^3*c^3*d*j^3*l*z - 144*a^4*b^3*c^2*
e*g*m^3*z - 144*a^4*b^3*c^2*d*h*m^3*z - 108*a^3*b*c^5*e^2*g^2*m*z + 108*a^3
*b*c^5*d^2*j^2*k*z - 108*a^3*b*c^5*d^2*h^2*m*z - 18*a^2*b^3*c^4*f^3*h*k*z -
 18*a^2*b^3*c^4*f^3*g*l*z - 9*a^3*b^3*c^3*g*h*j^3*z - 216*a^4*b*c^4*d*g^2*m
^2*z + 144*a^4*b^2*c^3*e*g*l^3*z - 126*a^3*b^2*c^4*d*h^3*l*z - 108*a^4*b*c^
4*d*h^2*l^2*z - 108*a^3*b*c^5*f^2*g^2*k*z - 108*a^3*b*c^5*e^2*h^2*k*z - 90*
a^2*b^2*c^5*e^3*f*m*z + 72*a^2*b^2*c^5*e^3*g*l*z - 63*a^3*b^4*c^2*e*g*l^3*z
 - 36*a^3*b^4*c^2*d*h*l^3*z + 27*a^2*b^4*c^3*d*h^3*l*z + 27*a*b^6*c^2*d^2*g
*m^2*z - 18*a^4*b^2*c^3*d*h*l^3*z - 18*a^3*b^2*c^4*f*h^3*j*z - 18*a^3*b^2*c
^4*e*h^3*k*z + 18*a^2*b^2*c^5*e^3*h*k*z + 108*a^3*b*c^5*e^2*h*j^2*z + 54*a^
3*b^3*c^3*d*h*k^3*z + 27*a^3*b^3*c^3*e*g*k^3*z - 27*a^2*b^3*c^4*e*g^3*k*z +
 27*a^2*b^3*c^4*d*g^3*l*z - 27*a*b^4*c^4*d^2*g^2*l*z - 9*a^2*b^5*c^2*e*g*k^
3*z - 9*a^2*b^5*c^2*d*h*k^3*z + 207*a^3*b^4*c^2*d*e*m^3*z - 108*a^2*b*c^6*d

^2*e^2*m*z - 90*a^4*b^2*c^3*d*e*m^3*z - 72*a^3*b^2*c^4*e*g*j^3*z - 72*a^3*b
^2*c^4*d*h*j^3*z + 27*a*b^3*c^5*d^2*e^2*m*z + 18*a^2*b^2*c^5*e*f^3*k*z + 18
*a^2*b^2*c^5*d*f^3*l*z + 9*a^2*b^4*c^3*e*g*j^3*z + 9*a^2*b^4*c^3*d*h*j^3*z
- 216*a^3*b*c^5*d*e^2*l^2*z - 198*a^3*b^3*c^3*d*e*l^3*z + 108*a^3*b*c^5*d*g
^2*j^2*z - 108*a^3*b*c^5*d*f^2*k^2*z + 72*a^2*b^5*c^2*d*e*l^3*z - 27*a*b^5*
c^3*d*e^2*l^2*z + 27*a*b^4*c^4*d^2*g*j^2*z + 18*a^2*b^2*c^5*f^3*g*h*z + 144
*a^3*b^2*c^4*d*e*k^3*z - 63*a^2*b^4*c^3*d*e*k^3*z + 27*a*b^4*c^4*d^2*e*k^2*
z - 9*a^2*b^3*c^4*e*g*h^3*z - 108*a^2*b*c^6*d^2*g^2*h*z + 81*a^2*b^3*c^4*d*
e*j^3*z + 27*a*b^3*c^5*d^2*g^2*h*z - 27*a*b^2*c^6*d^2*e^2*j*z - 18*a^2*b^2*
c^5*d*g^3*h*z + 108*a^2*b*c^6*d*e^2*h^2*z - 27*a*b^3*c^5*d*e^2*h^2*z + 27*a
*b^2*c^6*d^2*f^2*g*z - 18*a^2*b^2*c^5*d*e*h^3*z - 216*a^6*c^3*j^2*k*l*m*z +
216*a^6*c^3*h*j*l^2*m*z + 216*a^6*c^3*f*k*l*m^2*z - 216*a^5*c^4*f^2*k*l*m*
z - 216*a^5*c^4*g^2*j*k*m*z + 216*a^5*c^4*f*j^2*k*l*z + 216*a^5*c^4*f*h^2*l
*m*z + 216*a^5*c^4*e*j^2*k*m*z + 216*a^5*c^4*d*j^2*l*m*z + 216*a^5*c^4*g*h*
j^2*m*z - 216*a^5*c^4*e*j*k^2*l*z - 216*a^5*c^4*d*j*k^2*m*z + 216*a^4*c^5*d
^2*j*k*m*z - 18*a^6*b^2*c*k*l*m^3*z + 216*a^5*c^4*f*g*k^2*m*z - 216*a^5*c^4
*d*j*k*l^2*z - 72*a^6*b*c^2*j*l^3*m*z + 18*a^5*b^3*c*j*l^3*m*z - 216*a^5*c^
4*f*h*j*l^2*z + 216*a^5*c^4*e*h*k*l^2*z + 216*a^5*c^4*e*f*l^2*m*z - 216*a^4
*c^5*e^2*h*k*l*z + 216*a^4*c^5*e^2*h*j*m*z - 216*a^4*c^5*e^2*f*l*m*z - 216*
a^5*c^4*e*f*k*m^2*z + 216*a^5*c^4*d*g*k*m^2*z - 216*a^5*c^4*d*f*l*m^2*z + 2
16*a^4*c^5*e*f^2*k*m*z + 216*a^4*c^5*d*f^2*l*m*z + 108*a^5*b*c^3*j^3*k*l*z
- 216*a^5*c^4*f*g*h*m^2*z + 216*a^4*c^5*f^2*g*h*m*z + 216*a^4*c^5*f*g^2*j*k
*z - 216*a^4*c^5*e*g^2*j*l*z + 216*a^4*c^5*d*g^2*j*m*z - 72*a^6*b*c^2*h*k*m
^3*z - 72*a^6*b*c^2*g*l*m^3*z + 54*a^5*b^3*c*h*k*m^3*z + 54*a^5*b^3*c*g*l*m
^3*z - 216*a^4*c^5*d*h^2*j*k*z - 18*a^4*b^4*c*f*l^3*m*z + 9*a^4*b^4*c*h*k*l
^3*z - 216*a^4*c^5*e*f*j^2*k*z - 216*a^4*c^5*e*f*h^2*m*z - 216*a^4*c^5*d*g*
j^2*k*z - 216*a^4*c^5*d*f*j^2*l*z - 216*a^4*c^5*d*e*j^2*m*z - 72*a^5*b*c^3*
f*k^3*m*z + 72*a^4*b*c^4*g^3*j*m*z + 36*a^5*b*c^3*g*k^3*l*z - 36*a^4*b*c^4*
g^3*k*l*z - 216*a^4*c^5*f*g*h*j^2*z + 216*a^4*c^5*d*f*j*k^2*z - 216*a^3*c^6
*d^2*f*j*k*z - 216*a^3*c^6*d^2*e*j*l*z + 72*a^4*b^4*c*f*j*m^3*z - 63*a^4*b^
4*c*e*k*m^3*z - 63*a^4*b^4*c*d*l*m^3*z + 216*a^4*c^5*d*g*h*k^2*z - 216*a^3*
c^6*d^2*g*h*k*z + 216*a^3*c^6*d^2*f*g*m*z - 216*a^3*c^6*d*e^2*j*k*z + 144*a
^5*b*c^3*f*j*l^3*z - 144*a^3*b*c^5*e^3*j*m*z - 72*a^5*b*c^3*e*k*l^3*z + 72*
a^3*b*c^5*e^3*k*l*z - 63*a^4*b^4*c*g*h*m^3*z + 18*a^3*b^5*c*f*j*l^3*z - 18*
a*b^5*c^3*e^3*j*m*z - 9*a^3*b^5*c*e*k*l^3*z + 9*a*b^5*c^3*e^3*k*l*z - 216*a
^4*c^5*d*e*h*l^2*z - 216*a^3*c^6*e^2*f*h*j*z + 216*a^3*c^6*d*e^2*h*l*z - 12
6*a*b^4*c^4*d^3*j*m*z + 108*a^4*b*c^4*g*h^3*l*z + 63*a*b^4*c^4*d^3*k*l*z +
36*a^5*b*c^3*g*h*l^3*z - 9*a^3*b^5*c*g*h*l^3*z + 216*a^4*c^5*d*e*f*m^2*z +
216*a^3*c^6*d*f^2*g*k*z - 216*a^3*c^6*d*e*f^2*m*z + 36*a^4*b*c^4*e*j^3*k*z
+ 36*a^4*b*c^4*d*j^3*l*z - 216*a^3*c^6*d*f*g^2*j*z + 72*a^3*b^5*c*e*g*m^3*z
+ 72*a^3*b^5*c*d*h*m^3*z + 72*a^3*b*c^5*f^3*h*k*z + 72*a^3*b*c^5*f^3*g*l*z
+ 36*a^4*b*c^4*g*h*j^3*z + 18*a*b^4*c^4*e^3*f*m*z + 9*a^2*b^6*c*e*g*l^3*z
+ 9*a^2*b^6*c*d*h*l^3*z - 9*a*b^4*c^4*e^3*h*k*z - 9*a*b^4*c^4*e^3*g*l*z + 2
16*a^3*c^6*d*e*f*j^2*z - 144*a^2*b*c^6*d^3*f*m*z + 108*a^3*b*c^5*e*g^3*k*z
- 108*a^3*b*c^5*d*g^3*l*z + 108*a*b^3*c^5*d^3*f*m*z - 72*a^4*b*c^4*d*h*k^3*

z + 72*a^2*b*c^6*d^3*h*k*z - 54*a*b^3*c^5*d^3*h*k*z + 36*a^4*b*c^4*e*g*k^3*
z - 36*a^2*b*c^6*d^3*g*l*z - 27*a*b^3*c^5*d^3*g*l*z - 81*a^2*b^6*c*d*e*m^3*
z + 216*a^4*b*c^4*d*e*l^3*z + 72*a^2*b*c^6*e^3*f*j*z + 72*a^2*b*c^6*d*e^3*l
*z - 18*a*b^3*c^5*e^3*f*j*z - 18*a*b^3*c^5*d*e^3*l*z - 90*a*b^2*c^6*d^3*f*j
*z + 72*a*b^2*c^6*d^3*e*k*z + 36*a^3*b*c^5*e*g*h^3*z - 36*a^2*b*c^6*e^3*g*h
*z + 9*a*b^6*c^2*d*e*k^3*z + 9*a*b^3*c^5*e^3*g*h*z - 180*a^3*b*c^5*d*e*j^3*
z + 18*a*b^2*c^6*d^3*g*h*z - 9*a*b^5*c^3*d*e*j^3*z + 18*a*b^2*c^6*d*e^3*h*z
 + 9*a*b^4*c^4*d*e*h^3*z + 36*a^2*b*c^6*d*e*g^3*z - 9*a*b^3*c^5*d*e*g^3*z -
 18*a*b^2*c^6*d*e*f^3*z + 27*a^5*b^2*c^2*h^2*l*m^2*z - 27*a^5*b^2*c^2*j*k^2
*l^2*z + 27*a^4*b^3*c^2*h^2*k^2*m*z + 27*a^4*b^3*c^2*g^2*l^2*m*z + 27*a^5*b
^2*c^2*g*k^2*m^2*z - 27*a^4*b^3*c^2*h^2*k*l^2*z - 27*a^4*b^3*c^2*g^2*k*m^2*
z - 135*a^4*b^2*c^3*e^2*l*m^2*z + 27*a^5*b^2*c^2*e*l^2*m^2*z + 27*a^4*b^3*c
^2*h*j^2*l^2*z - 27*a^4*b^2*c^3*h^2*j^2*l*z + 27*a^3*b^4*c^2*e^2*l*m^2*z -
270*a^4*b^3*c^2*f*j^2*m^2*z - 270*a^4*b^2*c^3*f^2*j*m^2*z + 162*a^3*b^4*c^2
*f^2*j*m^2*z - 108*a^3*b^3*c^3*f^2*j^2*m*z - 27*a^4*b^2*c^3*h^2*j*k^2*z - 2
7*a^4*b^2*c^3*g^2*j*l^2*z + 27*a^3*b^3*c^3*e^2*k^2*m*z + 27*a^3*b^3*c^3*d^2
*l^2*m*z + 27*a^2*b^5*c^2*f^2*j^2*m*z + 162*a^3*b^3*c^3*d^2*k*m^2*z - 27*a^
4*b^3*c^2*d*k^2*m^2*z - 27*a^4*b^2*c^3*g*j^2*k^2*z + 27*a^3*b^3*c^3*g^2*h^2
*m*z - 27*a^2*b^5*c^2*d^2*k*m^2*z + 162*a^3*b^2*c^4*d^2*k^2*l*z - 108*a^4*b
^2*c^3*g*h^2*l^2*z - 27*a^4*b^2*c^3*e*j^2*l^2*z + 27*a^3*b^4*c^2*g*h^2*l^2*
z + 27*a^3*b^2*c^4*e^2*j^2*l*z - 27*a^2*b^4*c^3*d^2*k^2*l*z - 162*a^3*b^3*c
^3*f^2*h*l^2*z + 162*a^3*b^3*c^3*e^2*h*m^2*z - 135*a^4*b^2*c^3*e*h^2*m^2*z
+ 135*a^3*b^2*c^4*f^2*h^2*l*z + 27*a^3*b^4*c^2*e*h^2*m^2*z - 27*a^3*b^3*c^3
*g^2*h*k^2*z - 27*a^3*b^2*c^4*e^2*j*k^2*z - 27*a^3*b^2*c^4*d^2*j*l^2*z + 27
*a^2*b^5*c^2*f^2*h*l^2*z - 27*a^2*b^5*c^2*e^2*h*m^2*z - 27*a^2*b^4*c^3*f^2*
h^2*l*z - 27*a^3*b^2*c^4*g^2*h^2*j*z + 27*a^2*b^3*c^4*e^2*g^2*m*z - 27*a^2*
b^3*c^4*d^2*j^2*k*z + 27*a^2*b^3*c^4*d^2*h^2*m*z + 351*a^3*b^2*c^4*d^2*g*m^
2*z - 189*a^2*b^4*c^3*d^2*g*m^2*z + 162*a^3*b^3*c^3*d*g^2*m^2*z - 162*a^3*b
^2*c^4*e^2*g*l^2*z + 135*a^3*b^3*c^3*d*h^2*l^2*z + 135*a^3*b^2*c^4*f^2*g*k^
2*z - 27*a^2*b^5*c^2*d*h^2*l^2*z - 27*a^2*b^5*c^2*d*g^2*m^2*z - 27*a^2*b^4*
c^3*f^2*g*k^2*z + 27*a^2*b^4*c^3*e^2*g*l^2*z + 27*a^2*b^3*c^4*f^2*g^2*k*z +
 27*a^2*b^3*c^4*e^2*h^2*k*z + 135*a^3*b^2*c^4*e*f^2*l^2*z - 108*a^3*b^2*c^4
*e*g^2*k^2*z + 108*a^2*b^2*c^5*d^2*g^2*l*z + 27*a^3*b^2*c^4*e*h^2*j^2*z + 2
7*a^2*b^4*c^3*e*g^2*k^2*z - 27*a^2*b^4*c^3*e*f^2*l^2*z - 27*a^2*b^3*c^4*e^2
*h*j^2*z - 27*a^2*b^2*c^5*e^2*f^2*l*z - 27*a^2*b^2*c^5*e^2*g^2*j*z - 27*a^2
*b^2*c^5*d^2*h^2*j*z + 162*a^2*b^3*c^4*d*e^2*l^2*z - 135*a^2*b^2*c^5*d^2*g*
j^2*z - 27*a^2*b^3*c^4*d*g^2*j^2*z + 27*a^2*b^3*c^4*d*f^2*k^2*z - 162*a^2*b
^2*c^5*d^2*e*k^2*z - 27*a^2*b^2*c^5*e*f^2*h^2*z - 72*a^7*c^2*k*l*m^3*z + 9*
a^5*b^4*k*l*m^3*z + 72*a^6*c^3*j*k^3*m*z - 72*a^6*c^3*h*k*l^3*z - 72*a^6*c^
3*f*l^3*m*z - 72*a^5*c^4*h^3*k*l*z - 72*a^5*c^4*h^3*j*m*z - 9*a^4*b^5*h*k*m
^3*z - 9*a^4*b^5*g*l*m^3*z - 144*a^6*c^3*f*j*m^3*z - 144*a^5*c^4*h*j^3*k*z
- 144*a^5*c^4*g*j^3*l*z - 144*a^5*c^4*f*j^3*m*z - 144*a^4*c^5*f^3*j*m*z + 7
2*a^6*c^3*e*k*m^3*z + 72*a^6*c^3*d*l*m^3*z + 72*a^4*c^5*f^3*k*l*z + 72*a^6*
c^3*g*h*m^3*z + 18*b^6*c^3*d^3*j*m*z - 18*a^3*b^6*f*j*m^3*z - 9*b^6*c^3*d^3
*k*l*z + 9*a^3*b^6*e*k*m^3*z + 9*a^3*b^6*d*l*m^3*z + 144*a^5*c^4*d*k^3*l*z

```
+ 144*a^3*c^6*d^3*k*l*z - 72*a^5*c^4*f*j*k^3*z - 72*a^3*c^6*d^3*j*m*z + 9*a
^3*b^6*g*h*m^3*z - 72*a^5*c^4*g*h*k^3*z - 72*a^4*c^5*g^3*h*k*z - 72*a^4*c^5
*f*g^3*m*z - 108*a^5*b*c^3*j^4*m*z + 63*a^6*b^2*c*j*m^4*z + 36*a^6*b*c^2*k*
l^4*z - 9*a^5*b^3*c*k*l^4*z - 144*a^5*c^4*e*g*l^3*z - 144*a^3*c^6*e^3*g*l*z
 + 72*a^5*c^4*d*h*l^3*z + 72*a^4*c^5*f*h^3*j*z + 72*a^4*c^5*e*h^3*k*z + 72*
a^4*c^5*d*h^3*l*z + 72*a^3*c^6*e^3*h*k*z + 72*a^3*c^6*e^3*f*m*z - 18*b^5*c^
4*d^3*f*m*z + 9*b^5*c^4*d^3*h*k*z + 9*b^5*c^4*d^3*g*l*z - 9*a^2*b^7*e*g*m^3
*z - 9*a^2*b^7*d*h*m^3*z + 144*a^4*c^5*e*g*j^3*z + 144*a^4*c^5*d*h*j^3*z -
72*a^5*c^4*d*e*m^3*z - 72*a^3*c^6*e*f^3*k*z - 72*a^3*c^6*d*f^3*l*z + 144*a^
6*b*c^2*f*m^4*z - 108*a^5*b^3*c*f*m^4*z - 72*a^3*c^6*f^3*g*h*z + 36*a^5*b*c
^3*h*k^4*z - 36*a^3*b*c^5*f^4*m*z + 18*b^4*c^5*d^3*f*j*z - 9*b^4*c^5*d^3*e*
k*z + 9*a^4*b^4*c*g*l^4*z - 144*a^4*c^5*d*e*k^3*z - 144*a^2*c^7*d^3*e*k*z +
 72*a^2*c^7*d^3*f*j*z - 9*b^4*c^5*d^3*g*h*z + 72*a^3*c^6*d*g^3*h*z + 72*a^2
*c^7*d^3*g*h*z - 72*a^5*b*c^3*d*l^4*z - 72*a^4*b*c^4*f*j^4*z + 45*a*b^2*c^6
*d^4*l*z - 36*a^2*b*c^6*e^4*k*z - 9*a^3*b^5*c*d*l^4*z + 9*a*b^3*c^5*e^4*k*z
 - 72*a^3*c^6*d*e*h^3*z - 72*a^2*c^7*d*e^3*h*z + 9*b^3*c^6*d^3*e*g*z + 72*a
^2*c^7*d*e*f^3*z + 36*a^3*b*c^5*d*h^4*z - 9*a*b^2*c^6*e^4*g*z + 36*a*b*c^7*
d^3*f^2*z + 90*a^5*b^2*c^2*j^3*m^2*z + 45*a^5*b^2*c^2*j^2*l^3*z + 9*a^4*b^3
*c^2*j^2*k^3*z - 9*a^4*b^3*c^2*h^3*m^2*z - 45*a^4*b^2*c^3*g^3*m^2*z + 9*a^3
*b^4*c^2*g^3*m^2*z + 198*a^4*b^3*c^2*f^2*m^3*z - 108*a^3*b^3*c^3*f^3*m^2*z
+ 18*a^2*b^5*c^2*f^3*m^2*z - 117*a^4*b^2*c^3*f^2*l^3*z + 117*a^3*b^2*c^4*e^
3*m^2*z + 63*a^3*b^4*c^2*f^2*l^3*z - 63*a^2*b^4*c^3*e^3*m^2*z - 171*a^2*b^3
*c^4*d^3*m^2*z - 54*a^3*b^3*c^3*f^2*k^3*z + 9*a^3*b^2*c^4*g^3*j^2*z + 9*a^2
*b^5*c^2*f^2*k^3*z + 18*a^3*b^2*c^4*f^2*j^3*z + 18*a^2*b^3*c^4*f^3*j^2*z -
9*a^2*b^4*c^3*f^2*j^3*z - 45*a^2*b^2*c^5*e^3*j^2*z + 9*a^2*b^3*c^4*f^2*h^3*
z - 9*a^2*b^2*c^5*f^2*g^3*z + 9*a*b^8*d*e*m^3*z - 36*a*b*c^7*d^4*h*z - 108*
a^6*c^3*h^2*l*m^2*z + 108*a^6*c^3*j*k^2*l^2*z - 108*a^6*c^3*g*k^2*m^2*z - 1
08*a^6*c^3*e*l^2*m^2*z + 108*a^5*c^4*h^2*j^2*l*z + 108*a^5*c^4*e^2*l*m^2*z
+ 216*a^5*c^4*f^2*j*m^2*z + 108*a^5*c^4*h^2*j*k^2*z + 108*a^5*c^4*g^2*j*l^2
*z + 108*a^5*c^4*g*j^2*k^2*z - 216*a^4*c^5*d^2*k^2*l*z + 108*a^5*c^4*e*j^2*
l^2*z - 108*a^4*c^5*e^2*j^2*l*z - 9*a^6*b^2*c*l^3*m^2*z + 108*a^5*c^4*e*h^2
*m^2*z - 108*a^4*c^5*f^2*h^2*l*z + 108*a^4*c^5*e^2*j*k^2*z + 108*a^4*c^5*d^
2*j*l^2*z - 144*a^6*b*c^2*j^2*m^3*z + 108*a^4*c^5*g^2*h^2*j*z - 27*a^4*b^4*
c*j^3*m^2*z + 27*a^4*b^3*c^2*j^4*m*z + 9*a^5*b^2*c^2*k^4*l*z + 216*a^4*c^5*
e^2*g*l^2*z - 108*a^4*c^5*f^2*g*k^2*z - 108*a^4*c^5*d^2*g*m^2*z - 9*a^4*b^4
*c*j^2*l^3*z - 108*a^4*c^5*e*h^2*j^2*z - 108*a^4*c^5*e*f^2*l^2*z + 108*a^3*
c^6*e^2*f^2*l*z - 36*a^5*b*c^3*j^2*k^3*z + 36*a^5*b*c^3*h^3*m^2*z + 108*a^3
*c^6*e^2*g^2*j*z + 108*a^3*c^6*d^2*h^2*j*z - 216*a^5*b*c^3*f^2*m^3*z + 144*
a^4*b*c^4*f^3*m^2*z + 108*a^3*c^6*d^2*g*j^2*z - 72*a^3*b^5*c*f^2*m^3*z - 45
*a^5*b^2*c^2*g*l^4*z - 9*a^4*b^3*c^2*h*k^4*z - 9*a^3*b^2*c^4*g^4*l*z + 9*a^
2*b^3*c^4*f^4*m*z + 216*a^3*c^6*d^2*e*k^2*z - 9*a^2*b^6*c*f^2*l^3*z + 9*a*b
^6*c^2*e^3*m^2*z + 108*a^3*c^6*e*f^2*h^2*z + 108*a^3*b*c^5*d^3*m^2*z + 108*
a^2*c^7*d^2*e^2*j*z + 72*a^4*b*c^4*f^2*k^3*z + 72*a*b^5*c^3*d^3*m^2*z - 72*
a^3*b*c^5*f^3*j^2*z + 54*a^4*b^3*c^2*d*l^4*z - 45*a^4*b^2*c^3*e*k^4*z + 18*
a^3*b^3*c^3*f*j^4*z + 9*a^3*b^4*c^2*e*k^4*z - 9*a^2*b^2*c^5*f^4*j*z - 108*a
```

^2*c^7*d^2*f^2*g*z + 9*a^3*b^2*c^4*g*h^4*z + 9*a*b^4*c^4*e^3*j^2*z - 72*a^2
*b*c^6*d^3*j^2*z + 54*a*b^3*c^5*d^3*j^2*z - 36*a^3*b*c^5*f^2*h^3*z - 9*a^2*
b^3*c^4*d*h^4*z + 9*a^2*b^2*c^5*e*g^4*z + 9*a*b^2*c^6*e^3*f^2*z + 36*a^7*c^
2*l^3*m^2*z + 72*a^6*c^3*j^3*m^2*z - 36*a^6*c^3*j^2*l^3*z + 9*a^4*b^5*j^2*m
^3*z + 36*a^5*c^4*g^3*m^2*z + 36*a^5*c^4*f^2*l^3*z - 36*a^4*c^5*e^3*m^2*z -
 9*b^7*c^2*d^3*m^2*z + 9*a^2*b^7*f^2*m^3*z - 36*a^4*c^5*g^3*j^2*z + 72*a^4*
c^5*f^2*j^3*z + 36*a^3*c^6*e^3*j^2*z - 9*b^5*c^4*d^3*j^2*z + 36*a^3*c^6*f^2
*g^3*z - 9*a^4*b^2*c^3*j^5*z - 36*a^2*c^7*e^3*f^2*z - 9*b^3*c^6*d^3*f^2*z +
 36*a^7*c^2*j*m^4*z - 36*a^6*c^3*k^4*l*z - 18*a^5*b^4*j*m^4*z + 36*a^6*c^3*
g*l^4*z + 36*a^4*c^5*g^4*l*z + 18*a^4*b^5*f*m^4*z - 9*b^4*c^5*d^4*l*z + 36*
a^5*c^4*e*k^4*z + 36*a^3*c^6*f^4*j*z - 36*a^2*c^7*d^4*l*z - 36*a^4*c^5*g*h^
4*z + 9*b^3*c^6*d^4*h*z - 36*a^3*c^6*e*g^4*z + 36*a^2*c^7*e^4*g*z - 9*b^2*c
^7*d^4*e*z - 36*a^7*b*c*m^5*z + 36*a*c^8*d^4*e*z + 9*a^6*b^3*m^5*z + 36*a^5
*c^4*j^5*z + 9*a^4*b^3*c*g*h*j*k*l*m - 9*a^3*b^4*c*e*g*j*k*l*m - 9*a^3*b^4*
c*d*h*j*k*l*m - 9*a^3*b^4*c*f*g*h*k*l*m + 36*a^4*b*c^3*d*e*j*k*l*m + 9*a^2*
b^5*c*d*e*j*k*l*m + 36*a^4*b*c^3*e*f*h*j*l*m + 36*a^4*b*c^3*e*f*g*k*l*m + 3
6*a^4*b*c^3*d*f*h*k*l*m + 9*a^2*b^5*c*e*f*g*k*l*m + 9*a^2*b^5*c*d*f*h*k*l*m
 + 36*a^3*b*c^4*d*e*f*j*k*l + 9*a*b^5*c^2*d*e*f*j*k*l + 36*a^3*b*c^4*d*e*g*
h*k*l + 36*a^3*b*c^4*d*e*f*h*k*m + 36*a^3*b*c^4*d*e*f*g*l*m + 9*a*b^5*c^2*d
*e*f*h*k*m + 9*a*b^5*c^2*d*e*f*g*l*m - 9*a*b^4*c^3*d*e*f*h*j*k - 9*a*b^4*c^
3*d*e*f*g*j*l - 9*a*b^4*c^3*d*e*f*g*h*m + 9*a*b^3*c^4*d*e*f*g*h*j - 9*a*b^6
*c*d*e*f*k*l*m + 18*a^4*b^2*c^2*e*g*j*k*l*m + 18*a^4*b^2*c^2*d*h*j*k*l*m +
18*a^4*b^2*c^2*f*g*h*k*l*m - 36*a^3*b^3*c^2*d*e*j*k*l*m - 36*a^3*b^3*c^2*e*
f*g*k*l*m - 36*a^3*b^3*c^2*d*f*h*k*l*m + 9*a^3*b^3*c^2*f*g*h*j*k*l + 9*a^3*
b^3*c^2*e*g*h*j*k*m + 9*a^3*b^3*c^2*d*g*h*j*l*m - 108*a^3*b^2*c^3*d*e*f*k*l
*m + 54*a^2*b^4*c^2*d*e*f*k*l*m - 36*a^3*b^2*c^3*d*f*g*j*k*m + 18*a^3*b^2*c
^3*e*f*g*j*k*l + 18*a^3*b^2*c^3*d*f*h*j*k*l + 18*a^3*b^2*c^3*d*e*h*j*k*m +
18*a^3*b^2*c^3*d*e*g*j*l*m - 9*a^2*b^4*c^2*e*f*g*j*k*l - 9*a^2*b^4*c^2*d*f*
h*j*k*l - 9*a^2*b^4*c^2*d*e*h*j*k*m - 9*a^2*b^4*c^2*d*e*g*j*l*m + 18*a^3*b^
2*c^3*e*f*g*h*k*m + 18*a^3*b^2*c^3*d*f*g*h*l*m - 9*a^2*b^4*c^2*e*f*g*h*k*m
- 9*a^2*b^4*c^2*d*f*g*h*l*m - 36*a^2*b^3*c^3*d*e*f*j*k*l - 36*a^2*b^3*c^3*d
*e*f*h*k*m - 36*a^2*b^3*c^3*d*e*f*g*l*m + 9*a^2*b^3*c^3*e*f*g*h*j*k + 9*a^2
*b^3*c^3*d*f*g*h*j*l + 9*a^2*b^3*c^3*d*e*g*h*j*m + 18*a^2*b^2*c^4*d*e*f*h*j
*k + 18*a^2*b^2*c^4*d*e*f*g*j*l + 18*a^2*b^2*c^4*d*e*f*g*h*m - 9*a^5*b^2*c*
h*j*k^2*l*m - 9*a^5*b^2*c*g*j*k*l^2*m + 27*a^5*b^2*c*f*j*k*l*m^2 - 9*a^4*b^
3*c*f*j^2*k*l*m + 9*a^3*b^4*c*f^2*j*k*l*m - 18*a^5*b*c^2*e*j*k^2*l*m - 9*a^
5*b^2*c*g*h*k*l*m^2 + 9*a^4*b^3*c*e*j*k^2*l*m - 18*a^5*b*c^2*f*h*k^2*l*m -
18*a^5*b*c^2*d*j*k*l^2*m + 9*a^4*b^3*c*f*h*k^2*l*m + 9*a^4*b^3*c*d*j*k*l^2*
m + 36*a^5*b*c^2*e*h*k*l^2*m - 36*a^4*b*c^3*e^2*h*k*l*m + 18*a^5*b*c^2*f*h*
j*l^2*m - 18*a^5*b*c^2*f*g*k*l^2*m - 18*a^4*b^3*c*e*h*k*l^2*m + 9*a^4*b^3*c
*f*g*k*l^2*m + 9*a^3*b^4*c*e*h^2*k*l*m - 9*a^2*b^5*c*e^2*h*k*l*m - 54*a^5*b
*c^2*e*h*j*l*m^2 - 18*a^5*b*c^2*e*g*k*l*m^2 - 18*a^5*b*c^2*d*h*k*l*m^2 + 18
*a^4*b^3*c*e*h*j*l*m^2 - 9*a^4*b^3*c*f*h*j*k*m^2 - 9*a^4*b^3*c*f*g*j*l*m^2
+ 9*a^4*b^3*c*e*g*k*l*m^2 + 9*a^4*b^3*c*d*h*k*l*m^2 + 18*a^4*b*c^3*f*g^2*j*
k*m - 18*a^4*b*c^3*e*g^2*j*l*m + 18*a^3*b^4*c*d*g*k^2*l*m - 9*a^3*b^4*c*e*f

*k^2*l*m - 9*a^2*b^5*c*d*g^2*k*l*m - 18*a^4*b*c^3*f*g^2*h*l*m - 18*a^4*b*c^3*d*h^2*j*k*m - 9*a^3*b^4*c*d*f*k*l^2*m - 54*a^4*b*c^3*d*g*j^2*k*m - 18*a^4*b*c^3*f*g*h^2*k*m - 18*a^4*b*c^3*e*g*j^2*k*l - 18*a^4*b*c^3*d*h*j^2*k*l - 18*a^3*b^4*c*d*g*j*k*m^2 + 9*a^3*b^4*c*e*f*j*k*m^2 + 9*a^3*b^4*c*d*f*j*l*m^2 - 9*a^3*b^4*c*d*e*k*l*m^2 - 54*a^3*b*c^4*d^2*f*j*k*m + 36*a^4*b*c^3*d*g*j*k^2*l - 36*a^3*b*c^4*d^2*g*j*k*l - 18*a^4*b*c^3*e*f*j*k^2*l + 18*a^4*b*c^3*d*f*j*k^2*m - 18*a^3*b*c^4*d^2*e*j*l*m + 9*a^3*b^4*c*f*g*h*j*m^2 - 9*a*b^5*c^2*d^2*g*j*k*l + 36*a^4*b*c^3*d*g*h*k^2*m - 36*a^3*b*c^4*d^2*g*h*k*m + 18*a^4*b*c^3*e*g*h*k^2*l - 18*a^4*b*c^3*e*f*h*k^2*m - 18*a^4*b*c^3*d*f*j*k*l^2 - 18*a^3*b*c^4*d^2*f*h*l*m - 18*a^3*b*c^4*d*e^2*j*k*m - 9*a*b^5*c^2*d^2*g*h*k*m - 54*a^4*b*c^3*d*g*h*k*l^2 - 54*a^3*b*c^4*e^2*f*h*j*m - 18*a^4*b*c^3*d*f*g*l^2*m - 18*a^3*b*c^4*e^2*f*g*k*m - 54*a^4*b*c^3*d*f*g*k*m^2 - 36*a^4*b*c^3*e*f*g*j*m^2 - 36*a^4*b*c^3*d*f*h*j*m^2 + 36*a^3*b*c^4*e*f^2*g*j*m + 36*a^3*b*c^4*d*f^2*h*j*m - 18*a^4*b*c^3*d*e*h*k*m^2 - 18*a^4*b*c^3*d*e*g*l*m^2 + 18*a^3*b*c^4*e*f^2*h*j*l - 18*a^3*b*c^4*e*f^2*g*k*l - 18*a^3*b*c^4*d*f^2*h*k*l + 18*a^3*b*c^4*d*f^2*g*k*m - 9*a^2*b^5*c*e*f*g*j*m^2 - 9*a^2*b^5*c*d*f*h*j*m^2 - 54*a^3*b*c^4*d*f*g^2*j*m - 18*a^3*b*c^4*e*f*g^2*j*l - 18*a*b^4*c^3*d^2*f*g*j*m + 9*a*b^4*c^3*d^2*g*h*j*k + 9*a*b^4*c^3*d^2*f*g*k*l + 9*a*b^4*c^3*d^2*e*g*k*m - 9*a*b^4*c^3*d^2*e*f*l*m - 18*a^3*b*c^4*e*f*g^2*h*m - 18*a^3*b*c^4*d*f*h^2*j*k - 9*a*b^4*c^3*d*e^2*f*k*m + 18*a^3*b*c^4*d*f*g*j^2*k - 18*a^3*b*c^4*d*f*g*h^2*m - 18*a^3*b*c^4*d*e*h*j^2*k - 18*a^3*b*c^4*d*e*g*j^2*l + 18*a*b^4*c^3*d*e*f^2*j*m - 9*a*b^5*c^2*d*e*f*j^2*m - 9*a*b^4*c^3*d*e*f^2*k*l - 18*a^2*b*c^5*d^2*e*f*j*l - 9*a*b^3*c^4*d^2*e*g*j*k + 9*a*b^3*c^4*d^2*e*f*j*l - 54*a^2*b*c^5*d^2*e*g*h*l - 18*a^2*b*c^5*d^2*e*f*h*m - 18*a^2*b*c^5*d*e^2*f*j*k + 18*a*b^3*c^4*d^2*e*g*h*l - 9*a*b^3*c^4*d^2*f*g*h*k + 9*a*b^3*c^4*d^2*e*f*h*m + 9*a*b^3*c^4*d*e^2*f*j*k - 36*a^3*b*c^4*d*e*f*h*l^2 + 36*a^2*b*c^5*d*e^2*f*h*l + 18*a^2*b*c^5*d*e^2*g*h*k - 18*a^2*b*c^5*d*e^2*f*g*m - 18*a*b^3*c^4*d*e^2*f*h*l - 9*a*b^5*c^2*d*e*f*h*l^2 + 9*a*b^4*c^3*d*e*f*h^2*l + 9*a*b^3*c^4*d*e^2*f*g*m - 18*a^2*b*c^5*d*e*f^2*h*k - 18*a^2*b*c^5*d*e*f^2*g*l + 9*a*b^3*c^4*d*e*f^2*h*k + 9*a*b^3*c^4*d*e*f^2*g*l + 27*a*b^2*c^5*d^2*e*f*g*k + 9*a*b^4*c^3*d*e*f*g*k^2 - 9*a*b^3*c^4*d*e*f*g^2*k - 9*a*b^2*c^5*d^2*e*f*h*j - 9*a*b^2*c^5*d*e^2*f*g*j - 9*a*b^2*c^5*d*e*f^2*g*h + 72*a^4*c^4*d*f*g*j*k*m + 72*a^4*c^4*d*e*f*k*l*m + 9*a*b^6*c*d^2*g*k*l*m + 9*a*b^6*c*d*e*f*j*m^2 - 27*a^4*b^2*c^2*f^2*j*k*l*m - 9*a^4*b^2*c^2*g^2*h*j*l*m + 36*a^3*b^3*c^2*e^2*h*k*l*m - 18*a^4*b^2*c^2*e*h^2*k*l*m - 9*a^4*b^2*c^2*g*h^2*j*k*m + 18*a^4*b^2*c^2*f*h*j^2*k*m + 18*a^4*b^2*c^2*f*g*j^2*l*m - 18*a^4*b^2*c^2*e*h*j^2*l*m - 9*a^4*b^2*c^2*g*h*j^2*k*l - 9*a^3*b^3*c^2*f^2*h*j*k*m - 9*a^3*b^3*c^2*f^2*g*j*l*m - 63*a^4*b^2*c^2*d*g*k^2*l*m + 63*a^3*b^2*c^3*d^2*g*k*l*m - 45*a^2*b^4*c^2*d^2*g*k*l*m + 36*a^4*b^2*c^2*e*f*k^2*l*m + 27*a^3*b^3*c^2*d*g^2*k*l*m - 9*a^4*b^2*c^2*f*h*j*k^2*l - 9*a^4*b^2*c^2*e*h*j*k^2*m + 9*a^3*b^3*c^2*e*g^2*j*l*m - 9*a^3*b^2*c^3*d^2*h*j*l*m + 36*a^4*b^2*c^2*d*f*k*l^2*m + 27*a^4*b^2*c^2*e*h*j*k*l^2 - 27*a^3*b^2*c^3*e^2*h*j*k*l - 18*a^3*b^2*c^3*e^2*f*j*l*m - 9*a^4*b^2*c^2*f*g*j*k*l^2 - 9*a^4*b^2*c^2*d*g*j*l^2*m + 9*a^3*b^3*c^2*f*g^2*h*l*m - 9*a^3*b^3*c^2*e*h^2*j*k*l + 9*a^3*b^3*c^2*d*h^2*j*k*m - 9*a^3*b^2*c^3*e^2*g*j*k*m + 9*a^2*b^4*c^2

*e^2*h*j*k*l + 72*a^4*b^2*c^2*d*g*j*k*m^2 + 36*a^4*b^2*c^2*d*e*k*l*m^2 + 27
*a^4*b^2*c^2*e*g*h*l^2*m - 27*a^4*b^2*c^2*e*f*j*k*m^2 - 27*a^4*b^2*c^2*d*f*
j*l*m^2 - 27*a^3*b^2*c^3*e^2*g*h*l*m + 27*a^3*b^2*c^3*e*f^2*j*k*m + 27*a^3*
b^2*c^3*d*f^2*j*l*m + 18*a^3*b^3*c^2*d*g*j^2*k*m + 9*a^3*b^3*c^2*f*g*h^2*k*
m + 9*a^3*b^3*c^2*e*g*j^2*k*l - 9*a^3*b^3*c^2*e*g*h^2*l*m - 9*a^3*b^3*c^2*e
*f*j^2*k*m + 9*a^3*b^3*c^2*d*h*j^2*k*l - 9*a^3*b^3*c^2*d*f*j^2*l*m + 9*a^2*
b^4*c^2*e^2*g*h*l*m + 36*a^2*b^3*c^3*d^2*g*j*k*l - 27*a^4*b^2*c^2*f*g*h*j*m
^2 + 27*a^3*b^2*c^3*f^2*g*h*j*m - 18*a^4*b^2*c^2*e*f*h*l*m^2 - 18*a^3*b^3*c
^2*d*g*j*k^2*l - 18*a^3*b^2*c^3*d*g^2*j*k*l + 18*a^2*b^3*c^3*d^2*f*j*k*m -
9*a^4*b^2*c^2*e*g*h*k*m^2 - 9*a^4*b^2*c^2*d*g*h*l*m^2 - 9*a^3*b^3*c^2*f*g*h
*j^2*m + 9*a^3*b^3*c^2*e*f*j*k^2*l - 9*a^3*b^2*c^3*f^2*g*h*k*l + 9*a^2*b^4*
c^2*d*g^2*j*k*l + 9*a^2*b^3*c^3*d^2*e*j*l*m + 36*a^3*b^2*c^3*e*f*g^2*l*m +
36*a^2*b^3*c^3*d^2*g*h*k*m - 18*a^3*b^3*c^2*d*g*h*k^2*m - 18*a^3*b^2*c^3*d*
g^2*h*k*m + 9*a^3*b^3*c^2*e*f*h*k^2*m + 9*a^3*b^3*c^2*d*f*j*k*l^2 - 9*a^3*b
^2*c^3*f*g^2*h*j*l - 9*a^3*b^2*c^3*e*g^2*h*j*m - 9*a^2*b^4*c^2*e*f*g^2*l*m
+ 9*a^2*b^4*c^2*d*g^2*h*k*m + 9*a^2*b^3*c^3*d^2*f*h*l*m + 9*a^2*b^3*c^3*d*e
^2*j*k*m + 36*a^3*b^2*c^3*d*f*h^2*k*m + 36*a^3*b^2*c^3*d*e*j^2*k*l + 18*a^3
*b^3*c^2*d*g*h*k*l^2 + 18*a^3*b^2*c^3*e*g*h^2*j*l + 18*a^3*b^2*c^3*e*f*h^2*
k*l - 18*a^3*b^2*c^3*e*f*h^2*j*m - 18*a^3*b^2*c^3*d*g*h^2*k*l + 18*a^3*b^2*
c^3*d*e*h^2*l*m + 18*a^2*b^3*c^3*e^2*f*h*j*m - 9*a^3*b^3*c^2*e*g*h*j*l^2 -
9*a^3*b^3*c^2*e*f*h*k*l^2 + 9*a^3*b^3*c^2*d*f*g*l^2*m - 9*a^3*b^3*c^2*d*e*h
*l^2*m - 9*a^3*b^2*c^3*f*g*h^2*j*k - 9*a^3*b^2*c^3*d*g*h^2*j*m - 9*a^2*b^4*
c^2*d*f*h^2*k*m - 9*a^2*b^4*c^2*d*e*j^2*k*l - 9*a^2*b^3*c^3*e^2*g*h*j*l - 9
*a^2*b^3*c^3*e^2*f*h*k*l + 9*a^2*b^3*c^3*e^2*f*g*k*m - 9*a^2*b^3*c^3*d*e^2*
h*l*m + 36*a^3*b^3*c^2*e*f*g*j*m^2 + 36*a^3*b^3*c^2*d*f*h*j*m^2 + 18*a^3*b^
3*c^2*d*f*g*k*m^2 - 18*a^3*b^2*c^3*e*f*g*j^2*m - 18*a^3*b^2*c^3*d*f*h*j^2*m
 - 18*a^2*b^3*c^3*e*f^2*g*j*m - 18*a^2*b^3*c^3*d*f^2*h*j*m + 9*a^3*b^3*c^2*
d*e*h*k*m^2 + 9*a^3*b^3*c^2*d*e*g*l*m^2 - 9*a^3*b^2*c^3*e*g*h*j^2*k - 9*a^3
*b^2*c^3*d*g*h*j^2*l + 9*a^2*b^4*c^2*e*f*g*j^2*m + 9*a^2*b^4*c^2*d*f*h*j^2*
m + 9*a^2*b^3*c^3*e*f^2*g*k*l + 9*a^2*b^3*c^3*d*f^2*h*k*l + 72*a^2*b^2*c^4*
d^2*f*g*j*m + 36*a^2*b^2*c^4*d^2*e*f*l*m + 27*a^3*b^2*c^3*d*g*h*j*k^2 + 27*
a^3*b^2*c^3*d*f*g*k^2*l + 27*a^3*b^2*c^3*d*e*g*k^2*m - 27*a^2*b^2*c^4*d^2*g
*h*j*k - 27*a^2*b^2*c^4*d^2*f*g*k*l - 27*a^2*b^2*c^4*d^2*e*g*k*m + 18*a^2*b
^3*c^3*d*f*g^2*j*m - 18*a^2*b^2*c^4*d^2*e*h*k*l - 9*a^3*b^2*c^3*e*f*h*j*k^2
 + 9*a^2*b^3*c^3*e*f*g^2*j*l - 9*a^2*b^3*c^3*d*g^2*h*j*k - 9*a^2*b^3*c^3*d*
f*g^2*k*l - 9*a^2*b^3*c^3*d*e*g^2*k*m - 9*a^2*b^2*c^4*d^2*f*h*j*l - 9*a^2*b
^2*c^4*d^2*e*h*j*m + 36*a^2*b^2*c^4*d*e^2*f*k*m - 27*a^3*b^2*c^3*d*e*h*j*l^
2 + 27*a^2*b^2*c^4*d*e^2*h*j*l - 18*a^3*b^2*c^3*d*e*g*k*l^2 - 9*a^3*b^2*c^3
*d*f*g*j*l^2 + 9*a^2*b^4*c^2*d*e*h*j*l^2 + 9*a^2*b^3*c^3*e*f*g^2*h*m + 9*a^
2*b^3*c^3*d*f*h^2*j*k - 9*a^2*b^3*c^3*d*e*h^2*j*l - 9*a^2*b^2*c^4*e^2*f*g*j
*k - 9*a^2*b^2*c^4*d*e^2*g*j*m + 63*a^3*b^2*c^3*d*e*f*j*m^2 - 63*a^2*b^2*c^
4*d*e*f^2*j*m - 45*a^2*b^4*c^2*d*e*f*j*m^2 + 36*a^2*b^2*c^4*d*e*f^2*k*l - 2
7*a^3*b^2*c^3*e*f*g*h*l^2 + 27*a^2*b^3*c^3*d*e*f*j^2*m + 27*a^2*b^2*c^4*e^2
*f*g*h*l + 9*a^2*b^4*c^2*e*f*g*h*l^2 - 9*a^2*b^3*c^3*e*f*g*h^2*l + 9*a^2*b^
3*c^3*d*f*g*h^2*m + 9*a^2*b^3*c^3*d*e*h*j^2*k + 9*a^2*b^3*c^3*d*e*g*j^2*l +

18*a^2*b^2*c^4*d*e*g^2*j*k - 9*a^3*b^2*c^3*d*e*g*h*m^2 - 9*a^2*b^3*c^3*d*e
*g*j*k^2 - 9*a^2*b^2*c^4*e*f^2*g*h*k - 9*a^2*b^2*c^4*d*f^2*g*h*l + 18*a^2*b
^2*c^4*d*f*g^2*h*k - 18*a^2*b^2*c^4*d*e*g^2*h*l - 9*a^2*b^3*c^3*d*f*g*h*k^2
- 9*a^2*b^2*c^4*e*f*g^2*h*j + 36*a^2*b^3*c^3*d*e*f*h*l^2 - 18*a^2*b^2*c^4*
d*e*f*h^2*l - 9*a^2*b^2*c^4*d*f*g*h^2*j - 9*a^2*b^2*c^4*d*e*g*h*j^2 - 27*a^
2*b^2*c^4*d*e*f*g*k^2 + 18*a^2*b^2*c^4*d^2*f*h*k^2 - 9*a^2*b^3*c^3*e*f*g^2*
k^2 - 9*a^2*b^2*c^4*e^2*f*h*j^2 - 9*a^2*b^2*c^4*d*f^2*h^2*k + 45*a^2*b^3*c^
3*d*e*f^2*m^2 + 36*a^2*b^2*c^4*d^2*e*g*l^2 + 9*a^2*b^3*c^3*d*e*g^2*l^2 + 9*
a^2*b^2*c^4*e*f^2*g*j^2 + 9*a^2*b^2*c^4*d*f^2*h*j^2 - 9*a^2*b^2*c^4*d*e^2*h
*k^2 - 36*a^2*b^2*c^4*d*e^2*f*l^2 - 9*a^2*b^2*c^4*d*f*g^2*j^2 - 12*a^6*b*c*
h*k*l^3*m + 3*a*b^6*c*e^3*k*l*m + 3*a*b^6*c*d*e*f*l^3 - 12*a*b*c^6*d*e^3*f*
h + 9*a^5*b^2*c*h^2*k*l^2*m + 18*a^5*b*c^2*g^2*k^2*l*m - 9*a^5*b^2*c*h^2*j*
l*m^2 + 9*a^5*b*c^2*h^2*j^2*l*m - 9*a^4*b^3*c*g^2*k^2*l*m - 3*a^4*b^2*c^2*g
^3*k*l*m + 18*a^5*b*c^2*f^2*k*l*m^2 + 15*a^3*b^3*c^2*f^3*k*l*m + 9*a^5*b^2*
c*h*j^2*k*m^2 + 9*a^5*b^2*c*g*j^2*l*m^2 - 9*a^5*b^2*c*f*k^2*l^2*m + 9*a^5*b
*c^2*h^2*j*k^2*m + 9*a^5*b*c^2*g^2*j*l^2*m - 9*a^4*b^3*c*f^2*k*l*m^2 + 36*a
^3*b^2*c^3*e^3*k*l*m - 27*a^5*b*c^2*g^2*j*k*m^2 - 18*a^5*b*c^2*h^2*j*k*l^2
- 18*a^2*b^4*c^2*e^3*k*l*m - 9*a^5*b^2*c*g*j*k^2*m^2 - 9*a^5*b^2*c*e*k^2*l*
m^2 + 9*a^5*b*c^2*h*j^2*k^2*l + 9*a^5*b*c^2*g*j^2*k^2*m + 9*a^4*b^3*c*g^2*j
*k*m^2 + 9*a^3*b^4*c*e^2*k*l^2*m + 3*a^4*b^2*c^2*h^3*j*k*l - 54*a^4*b*c^3*d
^2*k^2*l*m - 51*a^2*b^3*c^3*d^3*k*l*m - 27*a^4*b*c^3*e^2*j^2*l*m - 18*a^5*b
*c^2*g*h^2*l^2*m - 9*a^5*b^2*c*e*j*l^2*m^2 - 9*a^5*b^2*c*d*k*l^2*m^2 + 9*a^
5*b*c^2*g^2*h*l*m^2 + 9*a^5*b*c^2*g*j^2*k*l^2 + 9*a^5*b*c^2*e*j^2*l^2*m - 9
*a^3*b^4*c*e^2*j*l*m^2 - 9*a^2*b^5*c*d^2*k^2*l*m + 3*a^4*b^2*c^2*g*h^3*l*m
- 3*a^3*b^3*c^2*g^3*j*k*l + 18*a^5*b*c^2*e*j^2*k*m^2 + 18*a^5*b*c^2*d*j^2*l
*m^2 + 18*a^4*b*c^3*f^2*j^2*k*l + 9*a^5*b*c^2*g*h^2*k*m^2 + 9*a^5*b*c^2*f*h
^2*l*m^2 + 9*a^5*b*c^2*f*j*k^2*l^2 - 9*a^4*b^3*c*e*j^2*k*m^2 - 9*a^4*b^3*c*
d*j^2*l*m^2 + 9*a^4*b^2*c^2*f*j^3*k*l + 9*a^4*b^2*c^2*e*j^3*k*m + 9*a^4*b^2
*c^2*d*j^3*l*m + 9*a^4*b*c^3*f^2*h^2*l*m + 9*a^4*b*c^3*e^2*j*k^2*m + 9*a^4*
b*c^3*d^2*j*l^2*m - 3*a^3*b^3*c^2*g^3*h*k*m - 3*a^3*b^2*c^3*f^3*j*k*l + 3*a
^2*b^4*c^2*f^3*j*k*l + 45*a^4*b*c^3*d^2*j*k*m^2 - 27*a^5*b*c^2*d*j*k^2*m^2
+ 18*a^5*b*c^2*g*h*j^2*m^2 + 18*a^4*b*c^3*e^2*j*k*l^2 + 15*a^2*b^3*c^3*e^3*
j*k*l - 12*a^3*b^2*c^3*f^3*h*k*m - 12*a^3*b^2*c^3*f^3*g*l*m + 9*a^5*b*c^2*g
*h*k^2*l^2 - 9*a^4*b^3*c*g*h*j^2*m^2 + 9*a^4*b^3*c*d*j*k^2*m^2 + 9*a^4*b^2*
c^2*g*h*j^3*m + 9*a^4*b*c^3*g^2*h^2*k*l + 9*a^4*b*c^3*g^2*h^2*j*m + 9*a^2*b
^5*c*d^2*j*k*m^2 + 3*a^2*b^4*c^2*f^3*h*k*m + 3*a^2*b^4*c^2*f^3*g*l*m + 36*a
^2*b^2*c^4*d^3*j*k*l + 18*a^4*b*c^3*e^2*g*l^2*m + 15*a^2*b^3*c^3*e^3*g*l*m
+ 12*a^4*b^2*c^2*d*j*k^3*l + 9*a^5*b*c^2*f*g*k^2*m^2 + 9*a^5*b*c^2*e*h*k^2*
m^2 + 9*a^4*b*c^3*g^2*h*j^2*l + 9*a^4*b*c^3*f^2*h*k^2*l + 9*a^4*b*c^3*f^2*g
*k^2*m + 9*a^4*b*c^3*d^2*h*l*m^2 - 9*a^3*b^3*c^2*e*h^3*k*m + 6*a^2*b^3*c^3*
e^3*h*k*m + 45*a^4*b*c^3*e^2*h*j*m^2 + 36*a^2*b^2*c^4*d^3*h*k*m - 33*a^3*b^
2*c^3*d*g^3*l*m - 27*a^4*b*c^3*f^2*h*j*l^2 - 27*a^4*b*c^3*e^2*f*l*m^2 - 27*
a^4*b*c^3*e*h^2*j^2*m - 18*a^4*b*c^3*g^2*h*j*k^2 - 18*a^4*b*c^3*f*g^2*k^2*l
- 18*a^4*b*c^3*e*g^2*k^2*m - 18*a^3*b*c^4*d^2*g^2*l*m + 12*a^4*b^2*c^2*d*h
*k^3*m + 9*a^5*b*c^2*e*f*l^2*m^2 + 9*a^5*b*c^2*d*g*l^2*m^2 + 9*a^4*b*c^3*f^

2*g*k*l^2 + 9*a^4*b*c^3*e^2*g*k*m^2 + 9*a^4*b*c^3*g*h^2*j^2*k + 9*a^4*b*c^3
*f*h^2*j^2*l + 9*a^4*b*c^3*e*f^2*l^2*m - 9*a^3*b^4*c*e*h^2*j*m^2 + 9*a^3*b*
c^4*e^2*f^2*l*m + 9*a^2*b^5*c*e^2*h*j*m^2 + 9*a^2*b^4*c^2*d*g^3*l*m - 9*a^2
*b^2*c^4*d^3*g*l*m - 9*a*b^5*c^2*d^2*g^2*l*m - 6*a^4*b^2*c^2*e*h*k^3*l - 6*
a^3*b^2*c^3*f*g^3*j*m + 3*a^4*b^2*c^2*g*h*j*k^3 + 3*a^4*b^2*c^2*f*g*k^3*l +
3*a^4*b^2*c^2*e*g*k^3*m + 3*a^3*b^2*c^3*g^3*h*j*k + 3*a^3*b^2*c^3*f*g^3*k*
l + 3*a^3*b^2*c^3*e*g^3*k*m - 27*a^3*b*c^4*d^2*h^2*k*l + 18*a^4*b*c^3*e*f^2
*k*m^2 + 18*a^4*b*c^3*d*f^2*l*m^2 + 9*a^4*b*c^3*f*h^2*j*k^2 + 9*a^4*b*c^3*f
*g^2*j*l^2 + 9*a^4*b*c^3*e*g^2*k*l^2 + 9*a^4*b*c^3*d*h^2*k^2*l + 9*a^3*b^4*
c*e*g*j^2*m^2 + 9*a^3*b^4*c*d*h*j^2*m^2 - 9*a^3*b^3*c^2*e*g*j^3*m - 9*a^3*b
^3*c^2*d*h*j^3*m + 9*a^3*b*c^4*e^2*g^2*k*l + 9*a^3*b*c^4*e^2*g^2*j*m + 9*a^
3*b*c^4*d^2*h^2*j*m - 3*a^2*b^3*c^3*f^3*h*j*k - 3*a^2*b^3*c^3*f^3*g*j*l - 3
*a^2*b^3*c^3*e*f^3*k*m - 3*a^2*b^3*c^3*d*f^3*l*m + 45*a^4*b*c^3*d*g^2*j*m^2
+ 45*a^3*b*c^4*d^2*g*j^2*m + 24*a^4*b^2*c^2*d*g*k*l^3 + 24*a^2*b^2*c^4*e^3
*f*j*m + 18*a^4*b*c^3*f^2*g*h*m^2 + 18*a^4*b*c^3*d*h^2*j*l^2 + 18*a^3*b*c^4
*e^2*h^2*j*k - 12*a^4*b^2*c^2*e*g*j*l^3 - 12*a^4*b^2*c^2*e*f*k*l^3 - 12*a^4
*b^2*c^2*d*e*l^3*m - 12*a^2*b^2*c^4*e^3*g*j*l - 12*a^2*b^2*c^4*e^3*f*k*l -
12*a^2*b^2*c^4*d*e^3*l*m + 9*a^4*b*c^3*f*g*j^2*k^2 + 9*a^4*b*c^3*e*h*j^2*k^
2 + 9*a^3*b^2*c^3*e*h^3*j*k + 9*a^3*b^2*c^3*d*h^3*j*l + 9*a^3*b*c^4*f^2*g^2
*j*k + 9*a^3*b*c^4*d^2*h*j^2*l + 9*a^2*b^5*c*d*g^2*j*m^2 + 9*a*b^5*c^2*d^2*
g*j^2*m - 3*a^4*b^2*c^2*d*h*j*l^3 - 3*a^2*b^3*c^3*f^3*g*h*m - 3*a^2*b^2*c^4
*e^3*h*j*k + 18*a^4*b*c^3*f*g*h^2*l^2 + 18*a^3*b*c^4*e^2*g*h^2*m + 18*a^3*b
*c^4*d^2*h*j*k^2 + 18*a^3*b*c^4*d^2*f*k^2*l + 18*a^3*b*c^4*d^2*e*k^2*m + 9*
a^4*b*c^3*e*g^2*h*m^2 + 9*a^4*b*c^3*e*f*j^2*l^2 + 9*a^4*b*c^3*d*g*j^2*l^2 +
9*a^3*b^2*c^3*f*g*h^3*l + 9*a^3*b^2*c^3*e*g*h^3*m + 9*a^3*b*c^4*f^2*g^2*h*
l + 9*a^3*b*c^4*e^2*g*j^2*k + 9*a^3*b*c^4*e^2*f*j^2*l - 9*a^2*b^3*c^3*d*g^3
*j*l + 9*a*b^4*c^3*d^2*g^2*j*l - 3*a^4*b^2*c^2*f*g*h*l^3 - 3*a^3*b^3*c^2*e*
g*j*k^3 - 3*a^3*b^3*c^2*d*h*j*k^3 - 3*a^3*b^3*c^2*d*f*k^3*l - 3*a^3*b^3*c^2
*d*e*k^3*m - 3*a^2*b^2*c^4*e^3*g*h*m - 33*a^3*b^2*c^3*d*e*j^3*m - 27*a^4*b*
c^3*e*f*h^2*m^2 - 27*a^3*b*c^4*d^2*e*k*l^2 - 18*a^4*b*c^3*d*e*j^2*m^2 - 18*
a^3*b*c^4*e*f^2*j^2*k - 18*a^3*b*c^4*d*f^2*j^2*l - 9*a^4*b^2*c^2*d*e*j*m^3
+ 9*a^4*b*c^3*d*g*h^2*m^2 + 9*a^4*b*c^3*d*e*k^2*l^2 + 9*a^3*b*c^4*f^2*g*h^2
*k + 9*a^3*b*c^4*e^2*f*j*k^2 + 9*a^3*b*c^4*d^2*f*j*l^2 + 9*a^3*b*c^4*e*f^2*
h^2*m + 9*a^3*b*c^4*d*e^2*k^2*l - 9*a^2*b^5*c*d*e*j^2*m^2 + 9*a^2*b^4*c^2*d
*e*j^3*m - 9*a^2*b^3*c^3*d*g^3*h*m + 9*a^2*b*c^5*d^2*e^2*k*l + 9*a^2*b*c^5*
d^2*e^2*j*m + 9*a*b^4*c^3*d^2*g^2*h*m - 6*a^3*b^2*c^3*d*g*j^3*k - 3*a^3*b^3
*c^2*f*g*h*k^3 + 3*a^3*b^2*c^3*e*f*j^3*k + 3*a^3*b^2*c^3*d*f*j^3*l + 3*a^2*
b^2*c^4*e*f^3*j*k + 3*a^2*b^2*c^4*d*f^3*j*l + 45*a^3*b*c^4*d^2*g*h*l^2 + 36
*a^4*b^2*c^2*e*f*g*m^3 + 36*a^4*b^2*c^2*d*f*h*m^3 - 27*a^3*b*c^4*e^2*g*h*k^
2 - 27*a^3*b*c^4*d*g^2*h^2*l - 18*a^3*b*c^4*f^2*g*h*j^2 + 18*a^3*b*c^4*d*e^
2*j*l^2 + 15*a^3*b^3*c^2*d*e*j*l^3 + 12*a^2*b^2*c^4*e*f^3*g*m + 12*a^2*b^2*
c^4*d*f^3*h*m + 9*a^3*b*c^4*f*g^2*h^2*j + 9*a^3*b*c^4*e*g^2*h^2*k + 9*a^3*b
*c^4*d*f^2*j*k^2 + 9*a^2*b*c^5*d^2*f^2*j*k + 9*a*b^5*c^2*d^2*g*h*l^2 - 9*a*
b^4*c^3*d^2*g*h^2*l - 6*a^2*b^2*c^4*e*f^3*h*l + 3*a^3*b^2*c^3*f*g*h*j^3 + 3
*a^2*b^2*c^4*f^3*g*h*j + 45*a^3*b*c^4*d^2*f*g*m^2 - 27*a^2*b*c^5*d^2*f^2*g*

m + 18*a^3*b*c^4*e^2*f*g*l^2 + 15*a^3*b^3*c^2*e*f*g*l^3 - 12*a^3*b^2*c^3*d*
e*j*k^3 + 9*a^3*b*c^4*d^2*e*h*m^2 + 9*a^3*b*c^4*e*g^2*h*j^2 + 9*a^3*b*c^4*e
*f^2*h*k^2 - 9*a^2*b^3*c^3*d*f*h^3*l + 9*a^2*b*c^5*d^2*f^2*h*l + 9*a*b^5*c^
2*d^2*f*g*m^2 + 9*a*b^3*c^4*d^2*f^2*g*m + 6*a^3*b^3*c^2*d*f*h*l^3 + 3*a^2*b
^4*c^2*d*e*j*k^3 + 18*a^3*b*c^4*e*f*g^2*k^2 + 18*a^2*b*c^5*d^2*g^2*h*j + 18
*a^2*b*c^5*d^2*f*g^2*l + 18*a^2*b*c^5*d^2*e*g^2*m - 12*a^3*b^2*c^3*d*f*h*k^
3 + 9*a^3*b*c^4*e*f*h^2*j^2 + 9*a^3*b*c^4*d*f^2*g*l^2 + 9*a^3*b*c^4*d*e^2*g
*m^2 + 9*a^3*b*c^4*d*g*h^2*j^2 + 9*a^2*b^2*c^4*e*f*g^3*k + 9*a^2*b^2*c^4*d*
g^3*h*j + 9*a^2*b^2*c^4*d*f*g^3*l + 9*a^2*b^2*c^4*d*e*g^3*m + 9*a^2*b*c^5*e
^2*f^2*h*j + 9*a^2*b*c^5*e^2*f^2*g*k - 9*a*b^3*c^4*d^2*g^2*h*j - 9*a*b^3*c^
4*d^2*f*g^2*l - 9*a*b^3*c^4*d^2*e*g^2*m - 3*a^3*b^2*c^3*e*f*g*k^3 + 3*a^2*b
^4*c^2*e*f*g*k^3 + 3*a^2*b^4*c^2*d*f*h*k^3 - 54*a^3*b*c^4*d*e*f^2*m^2 - 51*
a^3*b^3*c^2*d*e*f*m^3 - 27*a^3*b*c^4*d*e*g^2*l^2 + 9*a^3*b*c^4*d*e*h^2*k^2
+ 9*a^2*b*c^5*e^2*f*g^2*j + 9*a^2*b*c^5*d^2*f*h^2*j + 9*a^2*b*c^5*d^2*e*h^2
*k + 9*a^2*b*c^5*d*e^2*g^2*l - 9*a*b^5*c^2*d*e*f^2*m^2 - 9*a*b^4*c^3*d^2*e*
g*l^2 - 9*a*b^2*c^5*d^2*e^2*g*l - 9*a*b^2*c^5*d^2*e^2*f*m - 3*a^2*b^3*c^3*e
*f*g*j^3 - 3*a^2*b^3*c^3*d*f*h*j^3 + 36*a^3*b^2*c^3*d*e*f*l^3 - 27*a^2*b*c^
5*d^2*f*g*j^2 - 18*a^2*b^4*c^2*d*e*f*l^3 - 18*a^2*b*c^5*d*e^2*h^2*j + 9*a^2
*b*c^5*d^2*e*h*j^2 + 9*a^2*b*c^5*d*f^2*g^2*j + 9*a*b^4*c^3*d*e^2*f*l^2 + 9*
a*b^3*c^4*d^2*f*g*j^2 - 9*a*b^2*c^5*d^2*f^2*g*j - 9*a*b^2*c^5*d^2*e*f^2*l +
3*a^2*b^2*c^4*d*e*h^3*j - 18*a^2*b*c^5*e^2*f*g*h^2 + 18*a^2*b*c^5*d^2*e*f*
k^2 + 15*a^2*b^3*c^3*d*e*f*k^3 + 9*a^2*b*c^5*e*f^2*g^2*h + 9*a^2*b*c^5*d*e^
2*g*j^2 - 9*a*b^3*c^4*d^2*e*f*k^2 + 9*a*b^2*c^5*d^2*e*g^2*j - 9*a*b^2*c^5*d
*e^2*f^2*k + 3*a^2*b^2*c^4*e*f*g*h^3 + 18*a^2*b*c^5*d*e*f^2*j^2 + 9*a^2*b*c
^5*d*f^2*g*h^2 - 9*a*b^3*c^4*d*e*f^2*j^2 + 9*a*b^2*c^5*d^2*f*g^2*h - 3*a^2*
b^2*c^4*d*e*f*j^3 + 9*a^2*b*c^5*d*e*g^2*h^2 - 9*a*b^2*c^5*d^2*e*g*h^2 + 9*a
*b^2*c^5*d*e^2*f*h^2 - 36*a^6*c^2*f*j*k*l*m^2 + 36*a^5*c^3*f^2*j*k*l*m - 36
*a^5*c^3*f*h^2*j*l*m + 36*a^5*c^3*e*h*j^2*l*m - 18*a^6*b*c*j^2*k*l*m^2 + 9*
a^6*b*c*j*k^2*l^2*m + 3*a^5*b^2*c*j^3*k*l*m - 36*a^5*c^3*f*g*j*k^2*m - 36*a
^5*c^3*e*f*k^2*l*m + 36*a^5*c^3*d*g*k^2*l*m - 36*a^4*c^4*d^2*g*k*l*m - 36*a
^5*c^3*e*h*j*k*l^2 - 36*a^5*c^3*e*f*j*l^2*m - 36*a^5*c^3*d*f*k*l^2*m + 36*a
^4*c^4*e^2*h*j*k*l + 36*a^4*c^4*e^2*f*j*l*m + 9*a^6*b*c*h*k^2*l*m^2 - 3*a^4
*b^3*c*h^3*k*l*m - 36*a^5*c^3*e*g*h*l^2*m + 36*a^5*c^3*e*f*j*k*m^2 - 36*a^5
*c^3*d*g*j*k*m^2 + 36*a^5*c^3*d*f*j*l*m^2 - 36*a^5*c^3*d*e*k*l*m^2 + 36*a^4
*c^4*e^2*g*h*l*m - 36*a^4*c^4*e*f^2*j*k*m - 36*a^4*c^4*d*f^2*j*l*m + 9*a^6*
b*c*h*j*l^2*m^2 + 9*a^6*b*c*g*k*l^2*m^2 + 9*a^5*b^2*c*g*k^3*l*m + 3*a^3*b^4
*c*g^3*k*l*m + 36*a^5*c^3*f*g*h*j*m^2 + 36*a^5*c^3*e*f*h*l*m^2 - 36*a^4*c^4
*f^2*g*h*j*m - 36*a^4*c^4*e*f^2*h*l*m - 24*a^4*b*c^3*f^3*k*l*m - 12*a^5*b*c
^2*h*j^3*k*m - 12*a^5*b*c^2*g*j^3*l*m - 3*a^2*b^5*c*f^3*k*l*m - 36*a^4*c^4*
e*g^2*h*k*l - 36*a^4*c^4*e*f*g^2*l*m + 12*a^5*b^2*c*e*k*l^3*m - 6*a^5*b^2*c
*f*j*l^3*m + 3*a^5*b^2*c*h*j*k*l^3 + 48*a^3*b*c^4*d^3*k*l*m + 36*a^4*c^4*e*
f*h^2*j*m + 36*a^4*c^4*d*g*h^2*k*l - 36*a^4*c^4*d*f*h^2*k*m - 36*a^4*c^4*d*
e*j^2*k*l + 24*a^5*b*c^2*d*k^3*l*m + 21*a*b^5*c^2*d^3*k*l*m - 12*a^5*b*c^2*
g*j*k^3*l - 9*a^4*b^3*c*d*k^3*l*m + 6*a^5*b*c^2*f*j*k^3*m + 3*a^5*b^2*c*g*h
*l^3*m - 36*a^4*c^4*e*f*h*j^2*l - 12*a^5*b*c^2*g*h*k^3*m - 3*a^5*b^2*c*e*j*

k*m^3 - 3*a^5*b^2*c*d*j*l*m^3 - 36*a^4*c^4*d*g*h*j*k^2 - 36*a^4*c^4*d*f*g*k
^2*l - 36*a^4*c^4*d*e*h*k^2*l - 36*a^4*c^4*d*e*g*k^2*m + 36*a^3*c^5*d^2*g*h
*j*k + 36*a^3*c^5*d^2*f*g*k*l - 36*a^3*c^5*d^2*f*g*j*m + 36*a^3*c^5*d^2*e*h
*k*l + 36*a^3*c^5*d^2*e*g*k*m - 36*a^3*c^5*d^2*e*f*l*m + 24*a^5*b^2*c*e*h*l
*m^3 - 24*a^3*b*c^4*e^3*j*k*l - 12*a^5*b^2*c*f*h*k*m^3 - 12*a^5*b^2*c*f*g*l
*m^3 - 3*a^5*b^2*c*g*h*j*m^3 - 3*a^4*b^3*c*e*j*k*l^3 - 3*a*b^5*c^2*e^3*j*k*
l + 36*a^4*c^4*d*e*h*j*l^2 + 36*a^4*c^4*d*e*g*k*l^2 - 36*a^3*c^5*d*e^2*h*j*
l - 36*a^3*c^5*d*e^2*g*k*l - 36*a^3*c^5*d*e^2*f*k*m + 24*a^4*b*c^3*e*h^3*k*
m - 24*a^3*b*c^4*e^3*g*l*m - 18*a*b^4*c^3*d^3*j*k*l - 12*a^4*b*c^3*g*h^3*j*
l - 12*a^4*b*c^3*f*h^3*k*l - 12*a^4*b*c^3*d*h^3*l*m + 12*a^3*b*c^4*e^3*h*k*
m + 6*a^4*b*c^3*f*h^3*j*m - 3*a^4*b^3*c*g*h*j*l^3 - 3*a^4*b^3*c*f*h*k*l^3 -
3*a^4*b^3*c*e*g*l^3*m - 3*a^4*b^3*c*d*h*l^3*m - 3*a*b^5*c^2*e^3*h*k*m - 3*
a*b^5*c^2*e^3*g*l*m + 36*a^4*c^4*e*f*g*h*l^2 - 36*a^4*c^4*d*e*f*j*m^2 - 36*
a^3*c^5*e^2*f*g*h*l - 36*a^3*c^5*d*f^2*g*j*k - 36*a^3*c^5*d*e*f^2*k*l + 36*
a^3*c^5*d*e*f^2*j*m - 18*a*b^4*c^3*d^3*h*k*m - 9*a*b^4*c^3*d^3*g*l*m + 30*a
^5*b*c^2*d*g*k*m^3 - 30*a^4*b^3*c*d*g*k*m^3 - 24*a^5*b*c^2*e*f*k*m^3 - 24*a
^5*b*c^2*d*f*l*m^3 + 24*a^4*b*c^3*e*g*j^3*m + 24*a^4*b*c^3*d*h*j^3*m + 15*a
^4*b^3*c*e*f*k*m^3 + 15*a^4*b^3*c*d*f*l*m^3 + 12*a^5*b*c^2*e*g*j*m^3 + 12*a
^5*b*c^2*d*h*j*m^3 - 12*a^4*b*c^3*f*h*j^3*k - 12*a^4*b*c^3*f*g*j^3*l + 6*a^
4*b^3*c*e*g*j*m^3 + 6*a^4*b^3*c*d*h*j*m^3 + 6*a^4*b*c^3*e*h*j^3*l + 36*a^3*
c^5*d*e*g^2*h*l - 24*a^5*b*c^2*f*g*h*m^3 + 15*a^4*b^3*c*f*g*h*m^3 - 9*a*b^6
*c*d^2*g*j*m^2 - 6*a^3*b^4*c*d*g*k*l^3 - 6*a*b^4*c^3*e^3*f*j*m + 3*a^3*b^4*
c*e*g*j*l^3 + 3*a^3*b^4*c*e*f*k*l^3 + 3*a^3*b^4*c*d*h*j*l^3 + 3*a^3*b^4*c*d
*e*l^3*m + 3*a*b^4*c^3*e^3*h*j*k + 3*a*b^4*c^3*e^3*g*j*l + 3*a*b^4*c^3*e^3*
f*k*l + 3*a*b^4*c^3*d*e^3*l*m - 36*a^3*c^5*d*e*g*h^2*k + 30*a^2*b*c^5*d^3*f
*j*m - 30*a*b^3*c^4*d^3*f*j*m + 24*a^3*b*c^4*d*g^3*j*l - 24*a^2*b*c^5*d^3*h
*j*k - 24*a^2*b*c^5*d^3*f*k*l - 24*a^2*b*c^5*d^3*e*k*m + 15*a*b^3*c^4*d^3*h
*j*k + 15*a*b^3*c^4*d^3*f*k*l + 15*a*b^3*c^4*d^3*e*k*m - 12*a^3*b*c^4*e*g^3
*j*k + 12*a^2*b*c^5*d^3*g*j*l + 6*a*b^3*c^4*d^3*g*j*l + 3*a^3*b^4*c*f*g*h*l
^3 + 3*a*b^4*c^3*e^3*g*h*m + 24*a^3*b*c^4*d*g^3*h*m - 12*a^3*b*c^4*f*g^3*h*
k + 12*a^2*b*c^5*d^3*g*h*m - 9*a^3*b^4*c*d*e*j*m^3 + 6*a^3*b*c^4*e*g^3*h*l
+ 6*a*b^3*c^4*d^3*g*h*m + 36*a^3*c^5*d*e*f*g*k^2 - 36*a^2*c^6*d^2*e*f*g*k -
24*a^4*b*c^3*d*e*j*l^3 - 18*a^3*b^4*c*e*f*g*m^3 - 18*a^3*b^4*c*d*f*h*m^3 -
3*a^2*b^5*c*d*e*j*l^3 - 3*a*b^3*c^4*d*e^3*j*l - 24*a^4*b*c^3*e*f*g*l^3 + 2
4*a^3*b*c^4*d*f*h^3*l + 12*a^4*b*c^3*d*f*h*l^3 - 12*a^3*b*c^4*e*g*h^3*j - 1
2*a^3*b*c^4*e*f*h^3*k - 12*a^3*b*c^4*d*e*h^3*m - 12*a*b^2*c^5*d^3*e*j*k + 6
*a^3*b*c^4*d*g*h^3*k - 3*a^2*b^5*c*e*f*g*l^3 - 3*a^2*b^5*c*d*f*h*l^3 - 3*a*
b^3*c^4*e^3*g*h*j - 3*a*b^3*c^4*e^3*f*h*k - 3*a*b^3*c^4*e^3*f*g*l - 3*a*b^3
*c^4*d*e^3*h*m + 24*a*b^2*c^5*d^3*e*h*l - 12*a*b^2*c^5*d^3*f*h*k - 3*a*b^2*
c^5*d^3*g*h*j - 3*a*b^2*c^5*d^3*f*g*l - 3*a*b^2*c^5*d^3*e*g*m + 48*a^4*b*c^
3*d*e*f*m^3 + 24*a^2*b*c^5*d*e*f^3*m + 21*a^2*b^5*c*d*e*f*m^3 - 12*a^2*b*c^
5*e*f^3*g*j - 12*a^2*b*c^5*d*f^3*h*j - 9*a*b^3*c^4*d*e*f^3*m + 6*a^2*b*c^5*
d*f^3*g*k + 12*a*b^2*c^5*d*e^3*f*l - 6*a*b^2*c^5*d*e^3*g*k + 3*a*b^2*c^5*d*
e^3*h*j - 24*a^3*b*c^4*d*e*f*k^3 - 12*a^2*b*c^5*d*e*g^3*j - 3*a*b^5*c^2*d*e
*f*k^3 + 3*a*b^2*c^5*e^3*f*g*h - 12*a^2*b*c^5*d*f*g^3*h + 9*a*b^2*c^5*d*e*f

^3*j + 9*a*b*c^6*d^2*e^2*f*j + 3*a*b^4*c^3*d*e*f*j^3 + 9*a*b*c^6*d^2*e^2*g*
h + 9*a*b*c^6*d^2*e*f^2*h - 3*a*b^3*c^4*d*e*f*h^3 - 18*a*b*c^6*d^2*e*f*g^2
+ 9*a*b*c^6*d*e^2*f^2*g + 3*a*b^2*c^5*d*e*f*g^3 - 36*a^4*b^2*c^2*e^2*k*l^2*
m - 9*a^4*b^2*c^2*g^2*j^2*k*m + 45*a^3*b^3*c^2*d^2*k^2*l*m + 36*a^4*b^2*c^2
*e^2*j*l*m^2 + 9*a^4*b^2*c^2*g^2*j*k^2*l + 9*a^3*b^3*c^2*e^2*j^2*l*m + 9*a^
4*b^2*c^2*g^2*h*k^2*m - 9*a^4*b^2*c^2*f^2*h*l^2*m - 9*a^3*b^3*c^2*f^2*j^2*k
*l - 45*a^3*b^3*c^2*d^2*j*k*m^2 + 36*a^3*b^2*c^3*d^2*j^2*k*m + 18*a^4*b^2*c
^2*f^2*h*k*m^2 + 18*a^4*b^2*c^2*f^2*g*l*m^2 - 9*a^4*b^2*c^2*g^2*h*k*l^2 - 9
*a^4*b^2*c^2*f*h^2*k^2*m - 9*a^4*b^2*c^2*f*g^2*l^2*m - 9*a^4*b^2*c^2*e*j^2*
k^2*l - 9*a^4*b^2*c^2*d*j^2*k^2*m - 9*a^3*b^3*c^2*e^2*j*k*l^2 - 9*a^2*b^4*c
^2*d^2*j^2*k*m - 36*a^3*b^2*c^3*d^2*j*k^2*l - 27*a^3*b^2*c^3*e^2*h^2*k*m +
9*a^4*b^2*c^2*g*h^2*j*l^2 + 9*a^4*b^2*c^2*f*h^2*k*l^2 - 9*a^4*b^2*c^2*f*g^2
*k*m^2 - 9*a^4*b^2*c^2*e*g^2*l*m^2 - 9*a^4*b^2*c^2*d*j^2*k*l^2 + 9*a^4*b^2*
c^2*d*h^2*l^2*m - 9*a^3*b^3*c^2*e^2*g*l^2*m + 9*a^2*b^4*c^2*e^2*h^2*k*m + 9
*a^2*b^4*c^2*d^2*j*k^2*l - 45*a^3*b^3*c^2*e^2*h*j*m^2 + 36*a^4*b^2*c^2*e*h^
2*j*m^2 + 36*a^3*b^2*c^3*e^2*h*j^2*m - 36*a^3*b^2*c^3*d^2*h*k^2*m + 36*a^2*
b^3*c^3*d^2*g^2*l*m - 9*a^4*b^2*c^2*f*h*j^2*l^2 - 9*a^4*b^2*c^2*d*h^2*k*m^2
 + 9*a^3*b^3*c^2*f^2*h*j*l^2 + 9*a^3*b^3*c^2*e^2*f*l*m^2 + 9*a^3*b^3*c^2*e*
h^2*j^2*m - 9*a^3*b^2*c^3*f^2*h^2*j*l - 9*a^2*b^4*c^2*e^2*h*j^2*m + 9*a^2*b
^4*c^2*d^2*h*k^2*m + 36*a^3*b^2*c^3*d^2*h*k*l^2 - 27*a^4*b^2*c^2*e*g*j^2*m^
2 - 27*a^4*b^2*c^2*d*h*j^2*m^2 - 9*a^4*b^2*c^2*d*h*k^2*l^2 - 9*a^3*b^3*c^2*
e*f^2*k*m^2 - 9*a^3*b^3*c^2*d*f^2*l*m^2 + 9*a^3*b^2*c^3*f^2*h*j^2*k + 9*a^3
*b^2*c^3*f^2*g*j^2*l - 9*a^3*b^2*c^3*e^2*g*k^2*l - 9*a^3*b^2*c^3*e^2*f*k^2*
m - 9*a^3*b^2*c^3*d^2*f*l^2*m - 9*a^2*b^4*c^2*d^2*h*k*l^2 + 9*a^2*b^3*c^3*d
^2*h^2*k*l - 81*a^3*b^2*c^3*d^2*g*j*m^2 + 54*a^2*b^4*c^2*d^2*g*j*m^2 - 45*a
^3*b^3*c^2*d*g^2*j*m^2 - 45*a^2*b^3*c^3*d^2*g*j^2*m + 36*a^3*b^2*c^3*d^2*f*
k*m^2 + 36*a^3*b^2*c^3*d*g^2*j^2*m + 18*a^3*b^2*c^3*e^2*g*j*l^2 + 18*a^3*b^
2*c^3*e^2*f*k*l^2 + 18*a^3*b^2*c^3*d*e^2*l^2*m - 9*a^4*b^2*c^2*d*f*k^2*m^2
- 9*a^3*b^3*c^2*f^2*g*h*m^2 - 9*a^3*b^3*c^2*d*h^2*j*l^2 - 9*a^3*b^2*c^3*f^2
*g*j*k^2 - 9*a^3*b^2*c^3*d^2*e*l*m^2 - 9*a^3*b^2*c^3*f*g^2*h^2*m - 9*a^3*b^
2*c^3*e*g^2*j^2*l - 9*a^3*b^2*c^3*e*f^2*k^2*l - 9*a^2*b^4*c^2*d^2*f*k*m^2 -
 9*a^2*b^4*c^2*d*g^2*j^2*m - 9*a^2*b^3*c^3*e^2*h^2*j*k - 9*a^2*b^2*c^4*d^2*
f^2*k*m - 27*a^2*b^2*c^4*d^2*g^2*j*l - 9*a^3*b^3*c^2*f*g*h^2*l^2 + 9*a^3*b^
2*c^3*e*g^2*j*k^2 - 9*a^3*b^2*c^3*e*f^2*j*l^2 - 9*a^3*b^2*c^3*d*h^2*j^2*k -
 9*a^3*b^2*c^3*d*f^2*k*l^2 - 9*a^3*b^2*c^3*d*e^2*k*m^2 - 9*a^2*b^3*c^3*e^2*
g*h^2*m - 9*a^2*b^3*c^3*d^2*h*j*k^2 - 9*a^2*b^3*c^3*d^2*f*k^2*l - 9*a^2*b^3
*c^3*d^2*e*k^2*m + 36*a^3*b^3*c^2*d*e*j^2*m^2 + 36*a^3*b^2*c^3*e^2*f*h*m^2
- 27*a^2*b^2*c^4*d^2*g^2*h*m + 9*a^3*b^3*c^2*e*f*h^2*m^2 + 9*a^3*b^2*c^3*f*
g^2*h*k^2 - 9*a^2*b^4*c^2*e^2*f*h*m^2 + 9*a^2*b^3*c^3*d^2*e*k*l^2 - 9*a^2*b
^2*c^4*e^2*f^2*h*m - 45*a^2*b^3*c^3*d^2*g*h*l^2 - 36*a^3*b^2*c^3*e*f^2*g*m^
2 + 36*a^3*b^2*c^3*d*g^2*h*l^2 - 36*a^3*b^2*c^3*d*f^2*h*m^2 + 36*a^2*b^2*c^
4*d^2*g*h^2*l - 9*a^3*b^2*c^3*e*g*h^2*k^2 + 9*a^2*b^4*c^2*e*f^2*g*m^2 - 9*a
^2*b^4*c^2*d*g^2*h*l^2 + 9*a^2*b^4*c^2*d*f^2*h*m^2 + 9*a^2*b^3*c^3*e^2*g*h*
k^2 + 9*a^2*b^3*c^3*d*g^2*h^2*l - 9*a^2*b^3*c^3*d*e^2*j*l^2 - 9*a^2*b^2*c^4
*e^2*g^2*h*k - 9*a^2*b^2*c^4*e^2*f*g^2*m - 9*a^2*b^2*c^4*d^2*f*j^2*k - 9*a^

2*b^2*c^4*d^2*f*h^2*m - 9*a^2*b^2*c^4*d^2*e*j^2*l - 45*a^2*b^3*c^3*d^2*f*g*
m^2 + 36*a^3*b^2*c^3*d*f*g^2*m^2 - 27*a^3*b^2*c^3*d*f*h^2*l^2 + 18*a^2*b^2*
c^4*d^2*e*j*k^2 + 9*a^2*b^4*c^2*d*f*h^2*l^2 - 9*a^2*b^4*c^2*d*f*g^2*m^2 - 9
*a^2*b^3*c^3*e^2*f*g*l^2 + 9*a^2*b^2*c^4*e^2*g*h^2*j + 9*a^2*b^2*c^4*e^2*f*
h^2*k - 9*a^2*b^2*c^4*e*f^2*g^2*l - 9*a^2*b^2*c^4*d*f^2*g^2*m - 9*a^2*b^2*c
^4*d*e^2*j^2*k + 9*a^2*b^2*c^4*d*e^2*h^2*m + 18*a^4*b^2*c^2*f^2*j^2*m^2 + 1
8*a^3*b^2*c^3*e^2*h^2*l^2 - 9*a^2*b^4*c^2*e^2*h^2*l^2 + 18*a^2*b^2*c^4*d^2*
g^2*k^2 + 12*a^6*c^2*j^3*k*l*m + 3*a^6*b^2*j*k*l*m^3 - 12*a^6*c^2*g*k^3*l*m
 - 12*a^5*c^3*g^3*k*l*m - 24*a^6*c^2*e*k*l^3*m - 24*a^4*c^4*e^3*k*l*m + 12*
a^6*c^2*h*j*k*l^3 + 12*a^6*c^2*f*j*l^3*m + 12*a^5*c^3*h^3*j*k*l - 3*a^5*b^3
*h*j*k*m^3 - 3*a^5*b^3*g*j*l*m^3 - 3*a^5*b^3*f*k*l*m^3 + 12*a^6*c^2*g*h*l^3
*m + 12*a^5*c^3*g*h^3*l*m - 12*a^6*c^2*e*j*k*m^3 - 12*a^6*c^2*d*j*l*m^3 - 1
2*a^5*c^3*f*j^3*k*l - 12*a^5*c^3*e*j^3*k*m - 12*a^5*c^3*d*j^3*l*m - 12*a^4*
c^4*f^3*j*k*l + 24*a^6*c^2*f*h*k*m^3 + 24*a^6*c^2*f*g*l*m^3 + 24*a^4*c^4*f^
3*h*k*m + 24*a^4*c^4*f^3*g*l*m - 12*a^6*c^2*g*h*j*m^3 - 12*a^6*c^2*e*h*l*m^
3 - 12*a^5*c^3*g*h*j^3*m + 3*b^6*c^2*d^3*j*k*l + 3*a^4*b^4*e*j*k*m^3 + 3*a^
4*b^4*d*j*l*m^3 - 24*a^5*c^3*d*j*k^3*l - 24*a^3*c^5*d^3*j*k*l - 6*a^4*b^4*e
*h*l*m^3 + 3*b^6*c^2*d^3*h*k*m + 3*b^6*c^2*d^3*g*l*m + 3*a^6*b*c*j^2*l^3*m
+ 3*a^4*b^4*g*h*j*m^3 + 3*a^4*b^4*f*h*k*m^3 + 3*a^4*b^4*f*g*l*m^3 - 24*a^5*
c^3*d*h*k^3*m - 24*a^3*c^5*d^3*h*k*m + 12*a^5*c^3*g*h*j*k^3 + 12*a^5*c^3*f*
g*k^3*l + 12*a^5*c^3*e*h*k^3*l + 12*a^5*c^3*e*g*k^3*m + 12*a^4*c^4*g^3*h*j*
k + 12*a^4*c^4*f*g^3*k*l + 12*a^4*c^4*f*g^3*j*m + 12*a^4*c^4*e*g^3*k*m + 12
*a^4*c^4*d*g^3*l*m + 12*a^3*c^5*d^3*g*l*m + 3*a^6*b*c*j*k^3*m^2 - 9*a^6*b*c
*h^2*l*m^3 - 3*a^5*b*c^2*j^4*k*l + 24*a^5*c^3*e*g*j*l^3 + 24*a^5*c^3*e*f*k*
l^3 + 24*a^5*c^3*d*e*l^3*m + 24*a^3*c^5*e^3*g*j*l + 24*a^3*c^5*e^3*f*k*l +
24*a^3*c^5*d*e^3*l*m - 12*a^5*c^3*d*h*j*l^3 - 12*a^5*c^3*d*g*k*l^3 - 12*a^4
*c^4*e*h^3*j*k - 12*a^4*c^4*d*h^3*j*l - 12*a^3*c^5*e^3*h*j*k - 12*a^3*c^5*e
^3*f*j*m + 9*a^4*b*c^3*g^4*l*m + 6*b^5*c^3*d^3*f*j*m + 6*a^3*b^5*d*g*k*m^3
- 3*b^5*c^3*d^3*h*j*k - 3*b^5*c^3*d^3*g*j*l - 3*b^5*c^3*d^3*f*k*l - 3*b^5*c
^3*d^3*e*k*m - 3*a^3*b^5*e*g*j*m^3 - 3*a^3*b^5*e*f*k*m^3 - 3*a^3*b^5*d*h*j*
m^3 - 3*a^3*b^5*d*f*l*m^3 - 12*a^5*c^3*f*g*h*l^3 - 12*a^4*c^4*f*g*h^3*l - 1
2*a^4*c^4*e*g*h^3*m - 12*a^3*c^5*e^3*g*h*m - 9*a^6*b*c*g*k^2*m^3 - 3*b^5*c^
3*d^3*g*h*m + 3*a^6*b*c*f*l^3*m^2 - 3*a^3*b^5*f*g*h*m^3 + 12*a^5*c^3*d*e*j*
m^3 + 12*a^4*c^4*e*f*j^3*k + 12*a^4*c^4*d*g*j^3*k + 12*a^4*c^4*d*f*j^3*l +
12*a^4*c^4*d*e*j^3*m + 12*a^3*c^5*e*f^3*j*k + 12*a^3*c^5*d*f^3*j*l - 9*a^6*
b*c*e*l^2*m^3 - 24*a^5*c^3*e*f*g*m^3 - 24*a^5*c^3*d*f*h*m^3 - 24*a^3*c^5*e*
f^3*g*m - 24*a^3*c^5*d*f^3*h*m - 15*a^2*b*c^5*d^4*l*m + 15*a*b^3*c^4*d^4*l*
m + 12*a^4*c^4*f*g*h*j^3 + 12*a^3*c^5*f^3*g*h*j + 12*a^3*c^5*e*f^3*h*l + 9*
a^3*b*c^4*f^4*k*l - 9*a^3*b*c^4*f^4*j*m + 3*b^4*c^4*d^3*e*j*k + 3*a^5*b^2*c
*g*j*l^4 + 3*a^5*b^2*c*f*k*l^4 + 3*a^5*b^2*c*d*l^4*m - 3*a^5*b*c^2*h*j*k^4
- 3*a^5*b*c^2*f*k^4*l - 3*a^5*b*c^2*e*k^4*m - 3*a^4*b*c^3*h^4*j*k + 3*a^2*b
^6*d*e*j*m^3 + 3*a*b^4*c^3*e^4*k*m + 24*a^4*c^4*d*e*j*k^3 + 24*a^2*c^6*d^3*
e*j*k - 6*b^4*c^4*d^3*e*h*l + 3*b^4*c^4*d^3*g*h*j + 3*b^4*c^4*d^3*f*h*k + 3
*b^4*c^4*d^3*f*g*l + 3*b^4*c^4*d^3*e*g*m - 3*a^4*b*c^3*g*h^4*m + 3*a^2*b^6*
e*f*g*m^3 + 3*a^2*b^6*d*f*h*m^3 - 3*a*b^6*c*e^3*j*m^2 + 24*a^4*c^4*d*f*h*k^

3 + 24*a^2*c^6*d^3*f*h*k - 12*a^4*c^4*e*f*g*k^3 - 12*a^3*c^5*e*f*g^3*k - 12
*a^3*c^5*d*g^3*h*j - 12*a^3*c^5*d*f*g^3*l - 12*a^3*c^5*d*e*g^3*m - 12*a^2*c
^6*d^3*g*h*j - 12*a^2*c^6*d^3*f*g*l - 12*a^2*c^6*d^3*e*h*l - 12*a^2*c^6*d^3
*e*g*m - 12*a*b^2*c^5*d^4*j*l + 9*a^5*b*c^2*d*j*l^4 + 9*a^2*b*c^5*e^4*j*k -
 3*a^4*b^3*c*d*j*l^4 - 3*a^4*b*c^3*e*j^4*k - 3*a^4*b*c^3*d*j^4*l - 3*a*b^3*
c^4*e^4*j*k - 24*a^4*c^4*d*e*f*l^3 - 24*a^2*c^6*d*e^3*f*l - 12*a^5*b^2*c*e*
g*m^4 - 12*a^5*b^2*c*d*h*m^4 + 12*a^3*c^5*d*e*h^3*j + 12*a^2*c^6*d*e^3*h*j
+ 12*a^2*c^6*d*e^3*g*k - 12*a*b^2*c^5*d^4*h*m + 9*a^5*b*c^2*f*g*l^4 - 9*a^5
*b*c^2*e*h*l^4 - 9*a^2*b*c^5*e^4*h*l + 9*a^2*b*c^5*e^4*g*m + 6*a^4*b^3*c*e*
h*l^4 + 6*a*b^3*c^4*e^4*h*l - 3*b^3*c^5*d^3*e*g*j - 3*b^3*c^5*d^3*e*f*k - 3
*a^4*b^3*c*f*g*l^4 - 3*a^4*b*c^3*g*h*j^4 - 3*a^3*b*c^4*g^4*h*j - 3*a^3*b*c^
4*f*g^4*l - 3*a^3*b*c^4*e*g^4*m - 3*a*b^3*c^4*e^4*g*m + 12*a^3*c^5*e*f*g*h^
3 + 12*a^2*c^6*e^3*f*g*h - 3*b^3*c^5*d^3*f*g*h - 12*a^3*c^5*d*e*f*j^3 - 12*
a^2*c^6*d*e*f^3*j - 3*a*b^6*c*d^2*g*l^3 - 15*a^5*b*c^2*d*e*m^4 + 15*a^4*b^3
*c*d*e*m^4 + 9*a^4*b*c^3*e*f*k^4 - 9*a^4*b*c^3*d*g*k^4 + 3*a^3*b^4*c*d*f*l^
4 - 3*a^3*b*c^4*d*h^4*j - 3*a^2*b*c^5*e*f^4*k - 3*a^2*b*c^5*d*f^4*l + 3*a*b
^2*c^5*e^4*g*j + 3*a*b^2*c^5*e^4*f*k + 3*a*b^2*c^5*d*e^4*m - 9*a*b*c^6*d^3*
e^2*l + 3*b^2*c^6*d^3*e*f*g - 3*a^3*b*c^4*f*g*h^4 - 3*a^2*b*c^5*f^4*g*h + 1
2*a^2*c^6*d*e*f*g^3 - 9*a*b*c^6*d^3*f^2*j + 3*a*b*c^6*d^2*e^3*k + 9*a^3*b*c
^4*d*e*j^4 - 3*a^2*b*c^5*e*f*g^4 - 9*a*b*c^6*d^3*e*h^2 + 3*a*b*c^6*d^2*f^3*
g + 3*a*b*c^6*d*e^3*g^2 - 3*a^4*b^2*c^2*h^3*j^2*m + 12*a^4*b^2*c^2*g^3*j*m^
2 - 3*a^4*b^2*c^2*f^2*k^3*m + 3*a^3*b^3*c^2*g^3*j^2*m - 9*a^3*b^4*c*f^2*j^2
*m^2 + 9*a^3*b^3*c^2*f^2*j^3*m - 6*a^3*b^3*c^2*f^3*j*m^2 - 6*a^3*b^2*c^3*f^
3*j^2*m - 3*a^2*b^4*c^2*f^3*j^2*m - 27*a^4*b^2*c^2*d^2*k*m^3 - 27*a^3*b^2*c
^3*e^3*j*m^2 + 18*a^2*b^4*c^2*e^3*j*m^2 - 15*a^2*b^3*c^3*e^3*j^2*m + 12*a^4
*b^2*c^2*f^2*j*l^3 + 3*a^3*b^3*c^2*e^2*k^3*l + 42*a^2*b^3*c^3*d^3*j*m^2 - 2
7*a^2*b^2*c^4*d^3*j^2*m - 15*a^3*b^3*c^2*d^2*k*l^3 - 3*a^4*b^2*c^2*f*j^2*k^
3 - 3*a^4*b^2*c^2*f*h^3*m^2 + 3*a^3*b^3*c^2*g^3*h*l^2 + 3*a^3*b^3*c^2*f^2*j
*k^3 - 3*a^3*b^2*c^3*g^3*h^2*l - 3*a^3*b^2*c^3*e^2*j^3*l - 27*a^4*b^2*c^2*e
^2*h*m^3 + 12*a^3*b^2*c^3*f^3*h*l^2 + 3*a^3*b^3*c^2*f*g^3*m^2 - 3*a^2*b^4*c
^2*f^3*h*l^2 + 3*a^2*b^3*c^3*f^3*h^2*l + 9*a^3*b^3*c^2*e*h^3*l^2 + 9*a^2*b^
3*c^3*e^2*h^3*l - 6*a^4*b^2*c^2*e*h^2*l^3 - 6*a^3*b^3*c^2*e^2*h*l^3 - 6*a^2
*b^3*c^3*e^3*h*l^2 - 6*a^2*b^2*c^4*e^3*h^2*l + 3*a^2*b^3*c^3*d^2*j^3*k + 42
*a^3*b^3*c^2*d^2*g*m^3 - 27*a^4*b^2*c^2*d*g^2*m^3 - 27*a^2*b^2*c^4*d^3*h*l^
2 - 15*a^2*b^3*c^3*e^3*f*m^2 + 12*a^3*b^2*c^3*e^2*h*k^3 + 3*a^3*b^3*c^2*e*h
^2*k^3 - 3*a^3*b^2*c^3*e*g^3*l^2 - 3*a^2*b^4*c^2*e^2*h*k^3 + 3*a^2*b^3*c^3*
f^3*g*k^2 - 3*a^2*b^2*c^4*f^3*g^2*k - 27*a^3*b^2*c^3*d^2*g*l^3 - 27*a^2*b^2
*c^4*d^3*f*m^2 + 18*a^2*b^4*c^2*d^2*g*l^3 - 15*a^3*b^3*c^2*d*g^2*l^3 + 12*a
^2*b^2*c^4*e^3*g*k^2 - 3*a^3*b^2*c^3*e*h^2*j^3 + 3*a^2*b^3*c^3*e^2*h*j^3 +
3*a^2*b^3*c^3*e*f^3*l^2 - 3*a^2*b^2*c^4*d^2*h^3*k + 9*a^2*b^3*c^3*d*g^3*k^2
 - 9*a*b^4*c^3*d^2*g^2*k^2 - 6*a^3*b^2*c^3*d*g^2*k^3 - 6*a^2*b^3*c^3*d^2*g*
k^3 - 3*a^2*b^4*c^2*d*g^2*k^3 + 12*a^2*b^2*c^4*d^2*g*j^3 + 3*a^2*b^3*c^3*d*
g^2*j^3 - 3*a^2*b^2*c^4*d*f^3*k^2 - 3*a^2*b^2*c^4*d*g^2*h^3 + 12*a^7*c*j*k*
l*m^3 - 3*b^7*c*d^3*k*l*m - 3*a^6*b*c*k^4*l*m - 3*a^6*b*c*j*k*l^4 - 3*a^6*b
*c*g*l^4*m - 9*a^6*b*c*f*j*m^4 + 9*a^6*b*c*e*k*m^4 + 9*a^6*b*c*d*l*m^4 + 9*

a^6*b*c*g*h*m^4 - 3*a*b^7*d*e*f*m^3 + 9*a*b*c^6*d^4*h*j - 9*a*b*c^6*d^4*g*k
 + 9*a*b*c^6*d^4*f*l + 9*a*b*c^6*d^4*e*m + 12*a*c^7*d^3*e*f*g - 3*a*b*c^6*d
*e^4*j - 3*a*b*c^6*e^4*f*g - 3*a*b*c^6*d*e*f^4 + 18*a^6*c^2*h^2*j*l*m^2 - 1
8*a^6*c^2*h*j^2*l^2*m + 18*a^6*c^2*f*k^2*l^2*m + 36*a^5*c^3*e^2*k*l^2*m + 1
8*a^6*c^2*g*j*k^2*m^2 + 18*a^6*c^2*e*k^2*l*m^2 + 18*a^5*c^3*g^2*j^2*k*m + 1
8*a^6*c^2*e*j*l^2*m^2 + 18*a^6*c^2*d*k*l^2*m^2 - 18*a^5*c^3*e^2*j*l*m^2 - 1
8*a^6*c^2*f*h*l^2*m^2 + 18*a^5*c^3*f^2*h*l^2*m - 36*a^5*c^3*f^2*h*k*m^2 - 3
6*a^5*c^3*f^2*g*l*m^2 + 18*a^5*c^3*g^2*h*k*l^2 - 18*a^5*c^3*g*h^2*k^2*l + 1
8*a^5*c^3*f*h^2*k^2*m + 18*a^5*c^3*f*g^2*l^2*m + 18*a^5*c^3*e*j^2*k^2*l + 1
8*a^5*c^3*d*j^2*k^2*m - 18*a^4*c^4*d^2*j^2*k*m + 36*a^4*c^4*d^2*j*k^2*l + 1
8*a^5*c^3*f*g^2*k*m^2 + 18*a^5*c^3*e*g^2*l*m^2 + 18*a^5*c^3*d*j^2*k*l^2 - 1
8*a^4*c^4*f^2*g^2*k*m + 36*a^4*c^4*d^2*h*k^2*m + 18*a^5*c^3*f*h*j^2*l^2 - 1
8*a^5*c^3*e*h^2*j*m^2 + 18*a^5*c^3*d*h^2*k*m^2 + 18*a^4*c^4*f^2*h^2*j*l - 1
8*a^4*c^4*e^2*h*j^2*m - 18*a^5*c^3*e*g*k^2*l^2 + 18*a^5*c^3*d*h*k^2*l^2 + 1
8*a^4*c^4*e^2*g*k^2*l + 18*a^4*c^4*e^2*f*k^2*m - 18*a^4*c^4*d^2*h*k*l^2 + 1
8*a^4*c^4*d^2*f*l^2*m - 36*a^4*c^4*e^2*g*j*l^2 - 36*a^4*c^4*e^2*f*k*l^2 - 3
6*a^4*c^4*d*e^2*l^2*m + 18*a^5*c^3*d*f*k^2*m^2 + 18*a^4*c^4*f^2*g*j*k^2 + 1
8*a^4*c^4*d^2*g*j*m^2 - 18*a^4*c^4*d^2*f*k*m^2 + 18*a^4*c^4*d^2*e*l*m^2 - 1
8*a^4*c^4*f*g^2*j^2*k + 18*a^4*c^4*f*g^2*h^2*m + 18*a^4*c^4*e*g^2*j^2*l + 1
8*a^4*c^4*e*f^2*k^2*l - 18*a^4*c^4*d*g^2*j^2*m - 18*a^4*c^4*d*f^2*k^2*m + 1
8*a^3*c^5*d^2*f^2*k*m + 3*a^4*b^2*c^2*h^4*k*m - 3*a^3*b^3*c^2*g^4*l*m + 18*
a^4*c^4*e*f^2*j*l^2 + 18*a^4*c^4*d*h^2*j^2*k + 18*a^4*c^4*d*f^2*k*l^2 + 18*
a^4*c^4*d*e^2*k*m^2 - 18*a^3*c^5*e^2*f^2*j*l + 12*a^5*b^2*c*g^2*k*m^3 - 9*a
^5*b*c^2*h^3*j*m^2 - 9*a^5*b*c^2*f^2*l^3*m + 3*a^5*b*c^2*h^2*k^3*l + 3*a^4*
b^3*c*h^3*j*m^2 + 3*a^4*b^3*c*f^2*l^3*m - 18*a^4*c^4*e^2*f*h*m^2 + 18*a^3*c
^5*e^2*f^2*h*m + 15*a^5*b*c^2*e^2*l*m^3 - 15*a^4*b^3*c*e^2*l*m^3 - 9*a^5*b*
c^2*g^2*k*l^3 - 9*a^4*b*c^3*g^3*j^2*m - 3*a^5*b^2*c*g*k^2*l^3 + 3*a^5*b*c^2
*h*j^3*l^2 + 3*a^4*b^3*c*g^2*k*l^3 - 3*a^3*b^4*c*g^3*j*m^2 + 36*a^4*c^4*e*f
^2*g*m^2 + 36*a^4*c^4*d*f^2*h*m^2 + 18*a^4*c^4*e*g*h^2*k^2 - 18*a^4*c^4*d*g
^2*h*l^2 - 18*a^4*c^4*d*f*j^2*k^2 + 18*a^3*c^5*e^2*g^2*h*k + 18*a^3*c^5*e^2
*f*g^2*m - 18*a^3*c^5*d^2*g*h^2*l + 18*a^3*c^5*d^2*f*j^2*k + 18*a^3*c^5*d^2
*f*h^2*m + 18*a^3*c^5*d^2*e*j^2*l - 12*a^2*b^2*c^4*e^4*k*m + 9*a^4*b^3*c*f*
j^3*m^2 - 9*a^4*b^2*c^2*f*j^4*m - 6*a^5*b^2*c*f*j^2*m^3 + 6*a^5*b*c^2*f^2*j
*m^3 - 6*a^5*b*c^2*f*j^3*m^2 - 6*a^4*b^3*c*f^2*j*m^3 + 6*a^4*b*c^3*f^3*j*m^
2 - 6*a^4*b*c^3*f^2*j^3*m + 6*a^2*b^3*c^3*f^4*j*m + 3*a^3*b^2*c^3*g^4*j*l +
 3*a^2*b^5*c*f^3*j*m^2 - 3*a^2*b^3*c^3*f^4*k*l - 36*a^3*c^5*d^2*e*j*k^2 - 1
8*a^4*c^4*d*f*g^2*m^2 + 18*a^3*c^5*e*f^2*g^2*l + 18*a^3*c^5*d*f^2*g^2*m + 1
8*a^3*c^5*d*e^2*j^2*k + 18*a^3*b^4*c*d^2*k*m^3 + 15*a^3*b*c^4*e^3*j^2*m + 1
2*a^5*b^2*c*d*k^2*m^3 - 9*a^5*b*c^2*f*j^2*l^3 - 9*a^4*b*c^3*e^2*k^3*l + 3*a
^5*b*c^2*e*k^3*l^2 + 3*a^4*b^3*c*f*j^2*l^3 + 3*a^4*b*c^3*g^2*j^3*k - 3*a^3*
b^4*c*f^2*j*l^3 + 3*a^3*b^2*c^3*g^4*h*m + 3*a*b^5*c^2*e^3*j^2*m - 36*a^3*c^
5*d^2*f*h*k^2 - 21*a^3*b*c^4*d^3*j*m^2 - 21*a*b^5*c^2*d^3*j*m^2 + 18*a^3*c^
5*e^2*f*h*j^2 - 18*a^3*c^5*e*f^2*h^2*j + 18*a^3*c^5*d*f^2*h^2*k + 18*a*b^4*
c^3*d^3*j^2*m + 15*a^4*b*c^3*d^2*k*l^3 - 9*a^5*b*c^2*d*k^2*l^3 - 9*a^4*b*c^
3*g^3*h*l^2 - 9*a^4*b*c^3*f^2*j*k^3 + 3*a^4*b^3*c*d*k^2*l^3 + 3*a^2*b^5*c*d

^2*k*l^3 - 18*a^3*c^5*d^2*e*g*l^2 + 18*a^3*c^5*d*e^2*h*k^2 + 18*a^3*b^4*c*e
^2*h*m^3 - 18*a^2*c^6*d^2*e^2*h*k + 18*a^2*c^6*d^2*e^2*g*l + 18*a^2*c^6*d^2
*e^2*f*m + 15*a^5*b*c^2*e*h^2*m^3 - 15*a^4*b^3*c*e*h^2*m^3 - 9*a^4*b*c^3*f*
g^3*m^2 - 9*a^3*b*c^4*f^3*h^2*l + 3*a^4*b^2*c^2*e*j*k^4 + 3*a^4*b*c^3*g*h^3
*k^2 + 3*a^3*b*c^4*f^2*g^3*m + 36*a^3*c^5*d*e^2*f*l^2 + 18*a^3*c^5*d*f*g^2*
j^2 + 18*a^2*c^6*d^2*f^2*g*j + 18*a^2*c^6*d^2*e*f^2*l - 9*a^3*b^2*c^3*e*h^4
*l - 9*a^3*b*c^4*d^2*j^3*k + 6*a^4*b*c^3*e^2*h*l^3 - 6*a^4*b*c^3*e*h^3*l^2
+ 6*a^3*b*c^4*e^3*h*l^2 - 6*a^3*b*c^4*e^2*h^3*l + 3*a^4*b^2*c^2*f*h*k^4 + 3
*a^4*b*c^3*d*j^3*k^2 - 3*a^3*b^4*c*e*h^2*l^3 + 3*a^2*b^5*c*e^2*h*l^3 + 3*a^
2*b^2*c^4*f^4*h*k + 3*a^2*b^2*c^4*f^4*g*l + 3*a*b^5*c^2*e^3*h*l^2 - 3*a*b^4
*c^3*e^3*h^2*l - 21*a^4*b*c^3*d^2*g*m^3 - 21*a^2*b^5*c*d^2*g*m^3 + 18*a^3*b
^4*c*d*g^2*m^3 + 18*a^2*c^6*d*e^2*f^2*k + 18*a*b^4*c^3*d^3*h*l^2 + 15*a^3*b
*c^4*e^3*f*m^2 + 15*a^2*b*c^5*d^3*h^2*l - 15*a*b^3*c^4*d^3*h^2*l - 9*a^4*b*
c^3*e*h^2*k^3 - 9*a^3*b*c^4*f^3*g*k^2 - 9*a^2*b*c^5*e^3*f^2*m + 3*a^3*b*c^4
*f^2*h^3*j + 3*a*b^5*c^2*e^3*f*m^2 + 3*a*b^3*c^4*e^3*f^2*m + 18*a*b^4*c^3*d
^3*f*m^2 + 15*a^4*b*c^3*d*g^2*l^3 + 12*a*b^2*c^5*d^3*f^2*m - 9*a^3*b*c^4*e^
2*h*j^3 - 9*a^3*b*c^4*e*f^3*l^2 - 9*a^2*b*c^5*e^3*g^2*k + 3*a^3*b*c^4*f*g^3
*j^2 + 3*a^2*b^5*c*d*g^2*l^3 + 3*a^2*b*c^5*e^2*f^3*l - 3*a*b^4*c^3*e^3*g*k^
2 + 3*a*b^3*c^4*e^3*g^2*k + 18*a^2*c^6*d^2*e*g*h^2 - 18*a^2*c^6*d*e^2*g^2*h
 - 12*a^4*b^2*c^2*d*f*l^4 - 9*a^2*b^2*c^4*d*g^4*k + 9*a*b^3*c^4*d^2*g^3*k +
 6*a^3*b^3*c^2*d*g*k^4 + 6*a^3*b*c^4*d^2*g*k^3 - 6*a^3*b*c^4*d*g^3*k^2 + 6*
a^2*b*c^5*d^3*g*k^2 - 6*a^2*b*c^5*d^2*g^3*k - 6*a*b^3*c^4*d^3*g*k^2 - 6*a*b
^2*c^5*d^3*g^2*k - 3*a^3*b^3*c^2*e*f*k^4 + 3*a^3*b^2*c^3*e*g*j^4 + 3*a^3*b^
2*c^3*d*h*j^4 + 3*a*b^5*c^2*d^2*g*k^3 + 15*a^2*b*c^5*d^3*e*l^2 - 15*a*b^3*c
^4*d^3*e*l^2 - 9*a^3*b*c^4*d*g^2*j^3 - 9*a^2*b*c^5*e^3*f*j^2 - 3*a*b^4*c^3*
d^2*g*j^3 + 3*a*b^3*c^4*e^3*f*j^2 - 3*a*b^2*c^5*e^3*f^2*j + 12*a*b^2*c^5*d^
3*f*j^2 - 9*a^2*b*c^5*d*e^3*k^2 + 3*a^2*b*c^5*e^2*g^3*h + 3*a*b^3*c^4*d*e^3
*k^2 - 9*a^2*b*c^5*d^2*g*h^3 - 3*a^2*b^3*c^3*d*e*j^4 + 3*a^2*b*c^5*e*f^3*h^
2 + 3*a*b^3*c^4*d^2*g*h^3 + 3*a^2*b^2*c^4*d*f*h^4 - 9*a^7*c*k^2*l^2*m^2 - 6
*a^6*c^2*j^2*k^3*m - 3*a^6*b^2*h*l^2*m^3 + 3*a^5*b^3*h^2*l*m^3 - 6*a^6*c^2*
g^2*k*m^3 - 6*a^6*c^2*h*k^3*l^2 + 6*a^5*c^3*h^3*j^2*m + 6*a^6*c^2*g*k^2*l^3
 - 6*a^6*c^2*f*k^3*m^2 - 6*a^5*c^3*h^2*j^3*l - 6*a^5*c^3*g^3*j*m^2 + 6*a^5*
c^3*f^2*k^3*m + 3*a^5*b^3*g*k^2*m^3 - 3*a^4*b^4*g^2*k*m^3 + 12*a^6*c^2*f*j^
2*m^3 + 12*a^4*c^4*f^3*j^2*m + 3*a^5*b^3*e*l^2*m^3 + 3*a^3*b^5*e^2*l*m^3 -
6*a^6*c^2*d*k^2*m^3 - 6*a^5*c^3*f^2*j*l^3 + 6*a^5*c^3*d^2*k*m^3 - 6*a^5*c^3
*g*j^3*k^2 + 6*a^4*c^4*e^3*j*m^2 - 3*b^6*c^2*d^3*j^2*m - 3*a^4*b^4*f*j^2*m^
3 + 3*a^3*b^5*f^2*j*m^3 + 6*a^5*c^3*f*j^2*k^3 + 6*a^5*c^3*f*h^3*m^2 - 6*a^5
*c^3*e*j^3*l^2 + 6*a^4*c^4*g^3*h^2*l - 6*a^4*c^4*f^2*h^3*m + 6*a^4*c^4*e^2*
j^3*l + 6*a^3*c^5*d^3*j^2*m - 3*a^4*b^4*d*k^2*m^3 - 3*a^2*b^6*d^2*k*m^3 + 6
*a^5*c^3*e^2*h*m^3 - 6*a^4*c^4*g^2*h^3*k - 6*a^4*c^4*f^3*h*l^2 + 12*a^5*c^3
*e*h^2*l^3 + 12*a^3*c^5*e^3*h^2*l - 3*b^6*c^2*d^3*h*l^2 + 3*b^5*c^3*d^3*h^2
*l - 3*a^5*b^2*c*j^4*m^2 + 3*a^3*b^5*e*h^2*m^3 - 3*a^2*b^6*e^2*h*m^3 + 6*a^
5*c^3*d*g^2*m^3 - 6*a^4*c^4*e^2*h*k^3 - 6*a^4*c^4*f*h^3*j^2 + 6*a^4*c^4*e*g
^3*l^2 + 6*a^3*c^5*f^3*g^2*k - 6*a^3*c^5*e^2*g^3*l + 6*a^3*c^5*d^3*h*l^2 -
3*b^6*c^2*d^3*f*m^2 - 3*b^4*c^4*d^3*f^2*m + 6*a^4*c^4*d^2*g*l^3 + 6*a^4*c^4

*e*h^2*j^3 - 6*a^4*c^4*d*h^3*k^2 - 6*a^3*c^5*f^2*g^3*j - 6*a^3*c^5*e^3*g*k^
2 + 6*a^3*c^5*d^3*f*m^2 + 6*a^3*c^5*d^2*h^3*k - 6*a^2*c^6*d^3*f^2*m + 4*a^5
*b^2*c*h^3*m^3 + 3*b^5*c^3*d^3*g*k^2 - 3*b^4*c^4*d^3*g^2*k - 3*a^2*b^6*d*g^
2*m^3 + a^5*b*c^2*j^3*k^3 + 12*a^4*c^4*d*g^2*k^3 + 12*a^2*c^6*d^3*g^2*k + 6
*a^5*b*c^2*h^3*l^3 + 5*a^5*b*c^2*g^3*m^3 - 5*a^4*b^3*c*g^3*m^3 + 3*b^5*c^3*
d^3*e*l^2 + 3*b^3*c^5*d^3*e^2*l - 3*a^5*b^2*c*h^2*l^4 + a^4*b^3*c*h^3*l^3 +
12*a^5*b^2*c*f^2*m^4 - 6*a^3*c^5*d^2*g*j^3 + 6*a^3*c^5*d*f^3*k^2 + 6*a^3*b
^4*c*f^3*m^3 + 6*a^2*c^6*e^3*f^2*j - 6*a^2*c^6*d^2*f^3*k - 3*b^4*c^4*d^3*f*
j^2 + 3*b^3*c^5*d^3*f^2*j - 3*a^2*b^2*c^4*f^5*m - 7*a^4*b*c^3*e^3*m^3 - 7*a
^2*b^5*c*e^3*m^3 + 6*a^4*b*c^3*g^3*k^3 - 6*a^3*c^5*e*g^3*h^2 - 6*a^2*c^6*d^
3*f*j^2 + 5*a^4*b*c^3*f^3*l^3 + a^4*b*c^3*h^3*j^3 + a^2*b^5*c*f^3*l^3 + 6*a
^3*c^5*d*g^2*h^3 - 6*a^2*c^6*e^2*f^3*h - 3*a^3*b^4*c*e^2*l^4 - 3*a*b^4*c^3*
e^4*l^2 - 7*a^3*b*c^4*d^3*l^3 - 7*a*b^5*c^2*d^3*l^3 + 6*a^3*b*c^4*f^3*j^3 +
5*a^3*b*c^4*e^3*k^3 + 3*b^3*c^5*d^3*e*h^2 - 3*b^2*c^6*d^3*e^2*h + a*b^5*c^
2*e^3*k^3 + 12*a*b^2*c^5*d^4*k^2 - 6*a^2*c^6*d*f^3*g^2 + 6*a*b^4*c^3*d^3*k^
3 - 3*a^4*b^2*c^2*d*k^5 + a^3*b*c^4*g^3*h^3 + 5*a^2*b*c^5*d^3*j^3 - 5*a*b^3
*c^4*d^3*j^3 - 9*a*c^7*d^2*e^2*f^2 + 6*a^2*b*c^5*e^3*h^3 - 3*a*b^2*c^5*e^4*
h^2 + a^2*b*c^5*f^3*g^3 + a*b^3*c^4*e^3*h^3 + 4*a*b^2*c^5*d^3*h^3 - 3*a*b^2
*c^5*d^2*g^4 - 6*a^7*c*j*l^3*m^2 + 6*a^7*c*h*l^2*m^3 + 6*a^6*c^2*j*k^4*l +
6*a^6*c^2*h*k^4*m - 6*a^5*c^3*h^4*k*m + 3*a^6*b^2*h*k*m^4 + 3*a^6*b^2*g*l*m
^4 - 3*b^5*c^3*d^4*l*m - 6*a^6*c^2*g*j*l^4 - 6*a^6*c^2*f*k*l^4 - 6*a^6*c^2*
d*l^4*m + 6*a^5*c^3*h*j^4*k + 6*a^5*c^3*g*j^4*l + 6*a^5*c^3*f*j^4*m - 6*a^4
*c^4*g^4*j*l + 6*a^3*c^5*e^4*k*m + 6*a^5*b^3*f*j*m^4 - 6*a^4*c^4*g^4*h*m +
3*b^7*c*d^3*j*m^2 - 3*a^5*b^3*e*k*m^4 - 3*a^5*b^3*d*l*m^4 + 3*b^4*c^4*d^4*j
*l - 3*a^5*b^3*g*h*m^4 - 6*a^5*c^3*e*j*k^4 + 6*a^2*c^6*d^4*j*l + 3*b^4*c^4*
d^4*h*m + 6*a^6*c^2*e*g*m^4 + 6*a^6*c^2*d*h*m^4 + 6*a^6*b*c*j^3*m^3 - 6*a^5
*c^3*f*h*k^4 + 6*a^4*c^4*g*h^4*j + 6*a^4*c^4*f*h^4*k + 6*a^4*c^4*e*h^4*l +
6*a^4*c^4*d*h^4*m - 6*a^3*c^5*f^4*h*k - 6*a^3*c^5*f^4*g*l + 6*a^2*c^6*d^4*h
*m + 3*a^5*b*c^2*j^5*m + a^6*b*c*k^3*l^3 + 3*a^4*b^4*e*g*m^4 + 3*a^4*b^4*d*
h*m^4 + 6*b^3*c^5*d^4*g*k - 3*b^3*c^5*d^4*h*j - 3*b^3*c^5*d^4*f*l - 3*b^3*c
^5*d^4*e*m + 3*a*b^7*d^2*g*m^3 + 6*a^5*c^3*d*f*l^4 - 6*a^4*c^4*e*g*j^4 - 6*
a^4*c^4*d*h*j^4 + 6*a^3*c^5*e*g^4*j + 6*a^3*c^5*d*g^4*k - 6*a^2*c^6*e^4*g*j
- 6*a^2*c^6*e^4*f*k - 6*a^2*c^6*d*e^4*m + 3*a^4*b*c^3*h^5*l + 6*a^3*c^5*f*
g^4*h - 3*a^3*b^5*d*e*m^4 + 3*b^2*c^6*d^4*e*j + 3*a^5*b*c^2*g*k^5 + 3*a^3*b
*c^4*g^5*k + 8*a*b^6*c*d^3*m^3 + 3*b^2*c^6*d^4*f*h - 3*a^5*b^2*c*e*l^5 - 3*
a*b^2*c^5*e^5*l - 6*a^3*c^5*d*f*h^4 + 6*a^2*c^6*e*f^4*g + 6*a^2*c^6*d*f^4*h
+ 3*a^4*b*c^3*f*j^5 + 3*a^2*b*c^5*f^5*j + 6*a*c^7*d^3*e^2*h - 6*a*c^7*d^2*
e^3*g + 3*a^3*b*c^4*e*h^5 + 6*a*b*c^6*d^3*g^3 + 3*a^2*b*c^5*d*g^5 + a*b*c^6
*e^3*f^3 - 9*a^6*c^2*j^2*k^2*l^2 - 9*a^6*c^2*h^2*k^2*m^2 - 9*a^6*c^2*g^2*l^
2*m^2 - 18*a^5*c^3*f^2*j^2*m^2 - 9*a^5*c^3*h^2*j^2*k^2 - 9*a^5*c^3*g^2*j^2*
l^2 - 9*a^5*c^3*f^2*k^2*l^2 - 9*a^5*c^3*e^2*k^2*m^2 - 9*a^5*c^3*d^2*l^2*m^2
- 9*a^5*c^3*g^2*h^2*m^2 - 9*a^4*c^4*e^2*j^2*k^2 - 9*a^4*c^4*d^2*j^2*l^2 -
18*a^4*c^4*e^2*h^2*l^2 - 9*a^4*c^4*g^2*h^2*j^2 - 9*a^4*c^4*f^2*h^2*k^2 - 9*
a^4*c^4*f^2*g^2*l^2 - 9*a^4*c^4*e^2*g^2*m^2 - 9*a^4*c^4*d^2*h^2*m^2 - 18*a^
3*c^5*d^2*g^2*k^2 - 9*a^3*c^5*e^2*g^2*j^2 - 9*a^3*c^5*e^2*f^2*k^2 - 9*a^3*c

```
^5*d^2*h^2*j^2 - 9*a^3*c^5*d^2*f^2*l^2 - 9*a^3*c^5*d^2*e^2*m^2 - 3*a^4*b^2*
c^2*h^4*l^2 - 18*a^4*b^2*c^2*f^3*m^3 + 12*a^3*b^2*c^3*f^4*m^2 - 9*a^3*c^5*f
^2*g^2*h^2 + 4*a^4*b^2*c^2*g^3*l^3 - 3*a^2*b^4*c^2*f^4*m^2 + 14*a^3*b^3*c^2
*e^3*m^3 - 5*a^3*b^3*c^2*f^3*l^3 - 3*a^4*b^2*c^2*g^2*k^4 - 3*a^3*b^2*c^3*g^
4*k^2 + a^3*b^3*c^2*g^3*k^3 - 20*a^2*b^4*c^2*d^3*m^3 - 18*a^3*b^2*c^3*e^3*l
^3 + 16*a^3*b^2*c^3*d^3*m^3 + 12*a^4*b^2*c^2*e^2*l^4 + 12*a^2*b^2*c^4*e^4*l
^2 - 9*a^2*c^6*d^2*e^2*j^2 + 6*a^2*b^4*c^2*e^3*l^3 + 4*a^3*b^2*c^3*f^3*k^3
+ 14*a^2*b^3*c^3*d^3*l^3 - 9*a^2*c^6*e^2*f^2*g^2 - 9*a^2*c^6*d^2*f^2*h^2 -
5*a^2*b^3*c^3*e^3*k^3 - 3*a^3*b^2*c^3*f^2*j^4 - 3*a^2*b^2*c^4*f^4*j^2 + a^2
*b^3*c^3*f^3*j^3 - 18*a^2*b^2*c^4*d^3*k^3 + 12*a^3*b^2*c^3*d^2*k^4 + 4*a^2*
b^2*c^4*e^3*j^3 - 3*a^2*b^4*c^2*d^2*k^4 - 3*a^2*b^2*c^4*e^2*h^4 + 6*a^7*c*k
*l^4*m - 3*a^7*b*k*l*m^4 - 6*a^7*c*h*k*m^4 - 6*a^7*c*g*l*m^4 + 3*a^6*b*c*h*
l^5 - 6*a*c^7*d^4*e*j - 6*a*c^7*d^4*f*h - 3*b*c^7*d^4*e*f + 6*a*c^7*d*e^4*f
 + 3*a*b*c^6*e^5*h - a^5*b^2*c*j^3*l^3 - a^3*b^4*c*g^3*l^3 - a*b^4*c^3*e^3*
j^3 - a*b^2*c^5*e^3*g^3 + 3*a^7*b*j*m^5 + 6*a^7*c*f*m^5 + 6*a*c^7*d^5*k + 3
*b*c^7*d^5*g - 3*a^6*c^2*j^4*m^2 - 3*a^6*b^2*j^2*m^4 + 2*a^6*c^2*j^3*l^3 +
a^5*b^3*j^3*m^3 - 2*a^6*c^2*h^3*m^3 - 3*a^6*c^2*h^2*l^4 - 3*a^5*c^3*h^4*l^2
 - a*b^6*c*e^3*l^3 + 20*a^5*c^3*f^3*m^3 - 15*a^6*c^2*f^2*m^4 - 15*a^4*c^4*f
^4*m^2 + 2*a^5*c^3*h^3*k^3 - 2*a^5*c^3*g^3*l^3 + a^3*b^5*g^3*m^3 - 3*a^5*c^
3*g^2*k^4 - 3*a^4*c^4*g^4*k^2 - 3*a^4*b^4*f^2*m^4 + 20*a^4*c^4*e^3*l^3 - 15
*a^5*c^3*e^2*l^4 - 15*a^3*c^5*e^4*l^2 + 2*a^4*c^4*g^3*j^3 - 2*a^4*c^4*f^3*k
^3 - 2*a^4*c^4*d^3*m^3 - 3*b^4*c^4*d^4*k^2 - 3*a^4*c^4*f^2*j^4 - 3*a^3*c^5*
f^4*j^2 + 20*a^3*c^5*d^3*k^3 - 15*a^4*c^4*d^2*k^4 - 15*a^2*c^6*d^4*k^2 - 2*
a^3*c^5*e^3*j^3 + b^5*c^3*d^3*j^3 + 2*a^3*c^5*f^3*h^3 - 3*a^3*c^5*e^2*h^4 -
 3*a^2*c^6*e^4*h^2 - 3*b^2*c^6*d^4*g^2 + 2*a^2*c^6*e^3*g^3 - 2*a^2*c^6*d^3*
h^3 + b^3*c^5*d^3*g^3 - 3*a^2*c^6*d^2*g^4 - a^4*b^2*c^2*h^3*k^3 - a^3*b^2*c
^3*g^3*j^3 - a^2*b^4*c^2*f^3*k^3 - a^2*b^2*c^4*f^3*h^3 + 2*a^7*c*k^3*m^3 +
a^7*b*l^3*m^3 - 3*a^7*c*j^2*m^4 + 6*a^3*c^5*f^5*m - 3*a^6*b^2*f*m^5 + 6*a^6
*c^2*e*l^5 + 6*a^2*c^6*e^5*l + b^7*c*d^3*l^3 + a*b^7*e^3*m^3 - 3*b^2*c^6*d^
5*k + 6*a^5*c^3*d*k^5 - 3*a*c^7*d^4*g^2 + 2*a*c^7*d^3*f^3 + b*c^7*d^3*e^3 -
 a^6*b^2*k^3*m^3 - a^4*b^4*h^3*m^3 - a^2*b^6*f^3*m^3 - b^6*c^2*d^3*k^3 - b^
4*c^4*d^3*h^3 - b^2*c^6*d^3*f^3 - b^8*d^3*m^3 - a^6*c^2*k^6 - a^5*c^3*j^6 -
 a^4*c^4*h^6 - a^3*c^5*g^6 - a^2*c^6*f^6 - a^7*c*l^6 - a*c^7*e^6 - a^8*m^6
- c^8*d^6, z, k1)*(root(34992*a^4*b^2*c^8*z^6 - 8748*a^3*b^4*c^7*z^6 + 729*
a^2*b^6*c^6*z^6 - 46656*a^5*c^9*z^6 + 34992*a^4*b^3*c^6*m*z^5 - 8748*a^3*b^
5*c^5*m*z^5 + 729*a^2*b^7*c^4*m*z^5 - 34992*a^4*b^2*c^7*j*z^5 + 8748*a^3*b^
4*c^6*j*z^5 - 729*a^2*b^6*c^5*j*z^5 - 46656*a^5*b*c^7*m*z^5 + 46656*a^5*c^8
*j*z^5 + 34992*a^5*b*c^6*j*m*z^4 - 11664*a^5*b*c^6*k*l*z^4 + 3888*a^4*b*c^7
*f*j*z^4 + 3888*a^4*b*c^7*e*k*z^4 + 3888*a^4*b*c^7*d*l*z^4 + 3888*a^4*b*c^7
*g*h*z^4 + 3888*a^3*b*c^8*d*e*z^4 + 243*a*b^5*c^6*d*e*z^4 - 25272*a^4*b^3*c
^5*j*m*z^4 + 9720*a^4*b^3*c^5*k*l*z^4 + 6075*a^3*b^5*c^4*j*m*z^4 - 2673*a^3
*b^5*c^4*k*l*z^4 - 486*a^2*b^7*c^3*j*m*z^4 + 243*a^2*b^7*c^3*k*l*z^4 - 7776
*a^4*b^2*c^6*h*k*z^4 - 7776*a^4*b^2*c^6*g*l*z^4 - 7776*a^4*b^2*c^6*f*m*z^4
+ 2430*a^3*b^4*c^5*h*k*z^4 + 2430*a^3*b^4*c^5*g*l*z^4 + 2430*a^3*b^4*c^5*f*
m*z^4 - 243*a^2*b^6*c^4*h*k*z^4 - 243*a^2*b^6*c^4*g*l*z^4 - 243*a^2*b^6*c^4
```

*f*m*z^4 - 1944*a^3*b^3*c^6*f*j*z^4 - 1944*a^3*b^3*c^6*e*k*z^4 - 1944*a^3*b
^3*c^6*d*l*z^4 + 243*a^2*b^5*c^5*f*j*z^4 + 243*a^2*b^5*c^5*e*k*z^4 + 243*a^
2*b^5*c^5*d*l*z^4 - 1944*a^3*b^3*c^6*g*h*z^4 + 243*a^2*b^5*c^5*g*h*z^4 + 38
88*a^3*b^2*c^7*e*g*z^4 + 3888*a^3*b^2*c^7*d*h*z^4 - 486*a^2*b^4*c^6*e*g*z^4
 - 486*a^2*b^4*c^6*d*h*z^4 - 1944*a^2*b^3*c^7*d*e*z^4 + 7776*a^5*c^7*h*k*z^
4 + 7776*a^5*c^7*g*l*z^4 + 7776*a^5*c^7*f*m*z^4 - 7776*a^4*c^8*e*g*z^4 - 77
76*a^4*c^8*d*h*z^4 - 13608*a^5*b^2*c^5*m^2*z^4 + 11421*a^4*b^4*c^4*m^2*z^4
- 2916*a^3*b^6*c^3*m^2*z^4 + 243*a^2*b^8*c^2*m^2*z^4 + 13608*a^4*b^2*c^6*j^
2*z^4 - 3159*a^3*b^4*c^5*j^2*z^4 + 243*a^2*b^6*c^4*j^2*z^4 + 1944*a^3*b^2*c
^7*f^2*z^4 - 243*a^2*b^4*c^6*f^2*z^4 - 3888*a^6*c^6*m^2*z^4 - 19440*a^5*c^7
*j^2*z^4 - 3888*a^4*c^8*f^2*z^4 + 3078*a^4*b^4*c^3*k*l*m*z^3 - 2592*a^5*b^2
*c^4*k*l*m*z^3 - 891*a^3*b^6*c^2*k*l*m*z^3 - 4536*a^4*b^3*c^4*j*k*l*z^3 + 1
053*a^3*b^5*c^3*j*k*l*z^3 - 81*a^2*b^7*c^2*j*k*l*z^3 - 2592*a^4*b^3*c^4*h*k
*m*z^3 - 2592*a^4*b^3*c^4*g*l*m*z^3 + 810*a^3*b^5*c^3*h*k*m*z^3 + 810*a^3*b
^5*c^3*g*l*m*z^3 - 81*a^2*b^7*c^2*h*k*m*z^3 - 81*a^2*b^7*c^2*g*l*m*z^3 + 77
76*a^4*b^2*c^5*f*j*m*z^3 + 3888*a^4*b^2*c^5*h*j*k*z^3 + 3888*a^4*b^2*c^5*g*
j*l*z^3 - 3888*a^4*b^2*c^5*f*k*l*z^3 - 2916*a^3*b^4*c^4*f*j*m*z^3 + 1458*a^
3*b^4*c^4*f*k*l*z^3 - 972*a^3*b^4*c^4*h*j*k*z^3 - 972*a^3*b^4*c^4*g*j*l*z^3
 - 486*a^3*b^4*c^4*e*k*m*z^3 - 486*a^3*b^4*c^4*d*l*m*z^3 + 324*a^2*b^6*c^3*
f*j*m*z^3 - 162*a^2*b^6*c^3*f*k*l*z^3 + 81*a^2*b^6*c^3*h*j*k*z^3 + 81*a^2*b
^6*c^3*g*j*l*z^3 + 81*a^2*b^6*c^3*e*k*m*z^3 + 81*a^2*b^6*c^3*d*l*m*z^3 - 48
6*a^3*b^4*c^4*g*h*m*z^3 + 81*a^2*b^6*c^3*g*h*m*z^3 + 648*a^3*b^3*c^5*e*j*k*
z^3 + 648*a^3*b^3*c^5*d*j*l*z^3 - 81*a^2*b^5*c^4*e*j*k*z^3 - 81*a^2*b^5*c^4
*d*j*l*z^3 + 2592*a^3*b^3*c^5*e*g*m*z^3 + 2592*a^3*b^3*c^5*d*h*m*z^3 - 1296
*a^3*b^3*c^5*f*h*k*z^3 - 1296*a^3*b^3*c^5*f*g*l*z^3 - 1296*a^3*b^3*c^5*e*h*
l*z^3 + 648*a^3*b^3*c^5*g*h*j*z^3 - 324*a^2*b^5*c^4*e*g*m*z^3 - 324*a^2*b^5
*c^4*d*h*m*z^3 + 162*a^2*b^5*c^4*f*h*k*z^3 + 162*a^2*b^5*c^4*f*g*l*z^3 + 16
2*a^2*b^5*c^4*e*h*l*z^3 - 81*a^2*b^5*c^4*g*h*j*z^3 + 5184*a^3*b^2*c^6*d*e*m
*z^3 - 2592*a^3*b^2*c^6*e*g*j*z^3 - 2592*a^3*b^2*c^6*d*h*j*z^3 - 2106*a^2*b
^4*c^5*d*e*m*z^3 + 1296*a^3*b^2*c^6*e*f*k*z^3 + 1296*a^3*b^2*c^6*d*g*k*z^3
+ 1296*a^3*b^2*c^6*d*f*l*z^3 + 324*a^2*b^4*c^5*e*g*j*z^3 + 324*a^2*b^4*c^5*
d*h*j*z^3 - 162*a^2*b^4*c^5*e*f*k*z^3 - 162*a^2*b^4*c^5*d*g*k*z^3 - 162*a^2
*b^4*c^5*d*f*l*z^3 + 1296*a^3*b^2*c^6*f*g*h*z^3 - 162*a^2*b^4*c^5*f*g*h*z^3
 + 1944*a^2*b^3*c^6*d*e*j*z^3 - 1296*a^2*b^2*c^7*d*e*f*z^3 + 81*a^2*b^8*c*k
*l*m*z^3 + 6480*a^5*b*c^5*j*k*l*z^3 + 2592*a^5*b*c^5*h*k*m*z^3 + 2592*a^5*b
*c^5*g*l*m*z^3 - 1296*a^4*b*c^6*e*j*k*z^3 - 1296*a^4*b*c^6*d*j*l*z^3 - 5184
*a^4*b*c^6*e*g*m*z^3 - 5184*a^4*b*c^6*d*h*m*z^3 + 2592*a^4*b*c^6*f*h*k*z^3
+ 2592*a^4*b*c^6*f*g*l*z^3 + 2592*a^4*b*c^6*e*h*l*z^3 - 1296*a^4*b*c^6*g*h*
j*z^3 + 243*a*b^6*c^4*d*e*m*z^3 - 3888*a^3*b*c^7*d*e*j*z^3 - 243*a*b^5*c^5*
d*e*j*z^3 + 162*a*b^4*c^6*d*e*f*z^3 - 2592*a^6*c^5*k*l*m*z^3 - 5184*a^5*c^6
*h*j*k*z^3 - 5184*a^5*c^6*g*j*l*z^3 - 5184*a^5*c^6*f*j*m*z^3 + 2592*a^5*c^6
*f*k*l*z^3 + 2592*a^5*c^6*e*k*m*z^3 + 2592*a^5*c^6*d*l*m*z^3 + 2592*a^5*c^6
*g*h*m*z^3 + 5184*a^4*c^7*e*g*j*z^3 + 5184*a^4*c^7*d*h*j*z^3 - 2592*a^4*c^7
*e*f*k*z^3 - 2592*a^4*c^7*d*g*k*z^3 - 2592*a^4*c^7*d*f*l*z^3 - 2592*a^4*c^7
*d*e*m*z^3 - 2592*a^4*c^7*f*g*h*z^3 + 2592*a^3*c^8*d*e*f*z^3 + 6480*a^5*b^2

```
*c^4*j*m^2*z^3 + 6480*a^4*b^3*c^4*j^2*m*z^3 - 5022*a^4*b^4*c^3*j*m^2*z^3 -
1296*a^3*b^5*c^3*j^2*m*z^3 + 1134*a^3*b^6*c^2*j*m^2*z^3 + 81*a^2*b^7*c^2*j^
2*m*z^3 + 2592*a^4*b^3*c^4*h*l^2*z^3 - 1944*a^4*b^2*c^5*h^2*l*z^3 - 810*a^3
*b^5*c^3*h*l^2*z^3 + 729*a^3*b^4*c^4*h^2*l*z^3 + 81*a^2*b^7*c^2*h*l^2*z^3 -
 81*a^2*b^6*c^3*h^2*l*z^3 - 5184*a^4*b^3*c^4*f*m^2*z^3 + 1620*a^3*b^5*c^3*f
*m^2*z^3 + 1296*a^3*b^3*c^5*f^2*m*z^3 - 162*a^2*b^7*c^2*f*m^2*z^3 - 162*a^2
*b^5*c^4*f^2*m*z^3 - 1944*a^4*b^2*c^5*g*k^2*z^3 + 729*a^3*b^4*c^4*g*k^2*z^3
 - 648*a^3*b^3*c^5*g^2*k*z^3 - 81*a^2*b^6*c^3*g*k^2*z^3 + 81*a^2*b^5*c^4*g^
2*k*z^3 - 1944*a^4*b^2*c^5*e*l^2*z^3 + 729*a^3*b^4*c^4*e*l^2*z^3 + 648*a^3*
b^2*c^6*e^2*l*z^3 - 81*a^2*b^6*c^3*e*l^2*z^3 - 81*a^2*b^4*c^5*e^2*l*z^3 + 1
296*a^3*b^3*c^5*f*j^2*z^3 - 1296*a^3*b^2*c^6*f^2*j*z^3 - 162*a^2*b^5*c^4*f*
j^2*z^3 + 162*a^2*b^4*c^5*f^2*j*z^3 - 648*a^3*b^3*c^5*d*k^2*z^3 + 81*a^2*b^
5*c^4*d*k^2*z^3 + 648*a^3*b^2*c^6*e*h^2*z^3 - 81*a^2*b^4*c^5*e*h^2*z^3 - 64
8*a^2*b^2*c^7*d^2*g*z^3 - 10368*a^5*b*c^5*j^2*m*z^3 - 81*a^2*b^8*c*j*m^2*z^
3 - 2592*a^5*b*c^5*h*l^2*z^3 + 5184*a^5*b*c^5*f*m^2*z^3 - 2592*a^4*b*c^6*f^
2*m*z^3 + 1296*a^4*b*c^6*g^2*k*z^3 - 2592*a^4*b*c^6*f*j^2*z^3 + 1296*a^4*b*
c^6*d*k^2*z^3 + 81*a*b^4*c^6*d^2*g*z^3 + 2592*a^6*c^5*j*m^2*z^3 + 1296*a^5*
c^6*h^2*l*z^3 + 1296*a^5*c^6*g*k^2*z^3 + 1296*a^5*c^6*e*l^2*z^3 - 1296*a^4*
c^7*e^2*l*z^3 + 2592*a^4*c^7*f^2*j*z^3 - 2592*a^6*b*c^4*m^3*z^3 - 324*a^3*b
^7*c*m^3*z^3 - 27*a^2*b^8*c*l^3*z^3 - 1296*a^4*c^7*e*h^2*z^3 - 864*a^5*b*c^
5*k^3*z^3 + 1296*a^3*c^8*d^2*g*z^3 + 432*a^4*b*c^6*h^3*z^3 + 27*a*b^4*c^6*e
^3*z^3 - 432*a^2*b*c^8*d^3*z^3 + 216*a*b^3*c^7*d^3*z^3 + 1134*a^4*b^5*c^2*m
^3*z^3 - 432*a^5*b^3*c^3*m^3*z^3 + 1512*a^5*b^2*c^4*l^3*z^3 - 1107*a^4*b^4*
c^3*l^3*z^3 + 297*a^3*b^6*c^2*l^3*z^3 + 864*a^4*b^3*c^4*k^3*z^3 - 270*a^3*b
^5*c^3*k^3*z^3 + 27*a^2*b^7*c^2*k^3*z^3 - 2592*a^4*b^2*c^5*j^3*z^3 + 486*a^
3*b^4*c^4*j^3*z^3 - 27*a^2*b^6*c^3*j^3*z^3 - 216*a^3*b^3*c^5*h^3*z^3 + 27*a
^2*b^5*c^4*h^3*z^3 + 216*a^3*b^2*c^6*g^3*z^3 - 27*a^2*b^4*c^5*g^3*z^3 - 216
*a^2*b^2*c^7*e^3*z^3 - 432*a^6*c^5*l^3*z^3 + 27*a^2*b^9*m^3*z^3 + 4320*a^5*
c^6*j^3*z^3 - 432*a^4*c^7*g^3*z^3 + 432*a^3*c^8*e^3*z^3 - 27*b^5*c^6*d^3*z^
3 + 81*a^3*b^6*c*j*k*l*m*z^2 - 1296*a^5*b*c^4*h*j*k*m*z^2 - 1296*a^5*b*c^4*
g*j*l*m*z^2 + 1296*a^5*b*c^4*f*k*l*m*z^2 - 81*a^2*b^7*c*f*k*l*m*z^2 + 2592*
a^4*b*c^5*e*g*j*m*z^2 + 2592*a^4*b*c^5*d*h*j*m*z^2 - 1296*a^4*b*c^5*f*h*j*k
*z^2 - 1296*a^4*b*c^5*f*g*j*l*z^2 - 1296*a^4*b*c^5*e*f*k*m*z^2 - 1296*a^4*b
*c^5*d*f*l*m*z^2 - 648*a^4*b*c^5*e*h*j*l*z^2 - 648*a^4*b*c^5*e*g*k*l*z^2 -
648*a^4*b*c^5*d*h*k*l*z^2 - 648*a^4*b*c^5*d*g*k*m*z^2 - 1296*a^4*b*c^5*f*g*
h*m*z^2 - 162*a*b^6*c^3*d*e*j*m*z^2 + 81*a*b^6*c^3*d*e*k*l*z^2 + 1296*a^3*b
*c^6*d*e*f*m*z^2 - 648*a^3*b*c^6*d*f*g*k*z^2 - 648*a^3*b*c^6*d*e*h*k*z^2 -
648*a^3*b*c^6*d*e*g*l*z^2 - 81*a*b^5*c^4*d*e*h*k*z^2 - 81*a*b^5*c^4*d*e*g*l
*z^2 + 81*a*b^5*c^4*d*e*f*m*z^2 - 81*a*b^4*c^5*d*e*f*j*z^2 + 81*a*b^4*c^5*d
*e*g*h*z^2 + 648*a^5*b^2*c^3*j*k*l*m*z^2 - 567*a^4*b^4*c^2*j*k*l*m*z^2 - 19
44*a^4*b^3*c^3*f*k*l*m*z^2 + 729*a^3*b^5*c^2*f*k*l*m*z^2 + 648*a^4*b^3*c^3*
h*j*k*m*z^2 + 648*a^4*b^3*c^3*g*j*l*m*z^2 - 81*a^3*b^5*c^2*h*j*k*m*z^2 - 81
*a^3*b^5*c^2*g*j*l*m*z^2 + 1944*a^4*b^2*c^4*f*j*k*l*z^2 - 729*a^3*b^4*c^3*f
*j*k*l*z^2 + 648*a^4*b^2*c^4*e*j*k*m*z^2 + 648*a^4*b^2*c^4*d*j*l*m*z^2 - 81
*a^3*b^4*c^3*e*j*k*m*z^2 - 81*a^3*b^4*c^3*d*j*l*m*z^2 + 81*a^2*b^6*c^2*f*j*
```

k*l*z^2 + 1296*a^4*b^2*c^4*f*h*k*m*z^2 + 1296*a^4*b^2*c^4*f*g*l*m*z^2 + 648
*a^4*b^2*c^4*g*h*j*m*z^2 - 648*a^3*b^4*c^3*f*h*k*m*z^2 - 648*a^3*b^4*c^3*f*
g*l*m*z^2 - 324*a^4*b^2*c^4*g*h*k*l*z^2 - 324*a^4*b^2*c^4*e*h*l*m*z^2 + 81*
a^3*b^4*c^3*g*h*k*l*z^2 - 81*a^3*b^4*c^3*g*h*j*m*z^2 + 81*a^2*b^6*c^2*f*h*k
*m*z^2 + 81*a^2*b^6*c^2*f*g*l*m*z^2 - 1296*a^3*b^3*c^4*e*g*j*m*z^2 - 1296*a
^3*b^3*c^4*d*h*j*m*z^2 + 648*a^3*b^3*c^4*f*h*j*k*z^2 + 648*a^3*b^3*c^4*f*g*
j*l*z^2 + 648*a^3*b^3*c^4*e*f*k*m*z^2 + 648*a^3*b^3*c^4*d*f*l*m*z^2 + 486*a
^3*b^3*c^4*e*g*k*l*z^2 + 486*a^3*b^3*c^4*d*h*k*l*z^2 + 162*a^3*b^3*c^4*e*h*
j*l*z^2 + 162*a^3*b^3*c^4*d*g*k*m*z^2 + 162*a^2*b^5*c^3*e*g*j*m*z^2 + 162*a
^2*b^5*c^3*d*h*j*m*z^2 - 81*a^2*b^5*c^3*f*h*j*k*z^2 - 81*a^2*b^5*c^3*f*g*j*
l*z^2 - 81*a^2*b^5*c^3*e*g*k*l*z^2 - 81*a^2*b^5*c^3*e*f*k*m*z^2 - 81*a^2*b^
5*c^3*d*h*k*l*z^2 - 81*a^2*b^5*c^3*d*f*l*m*z^2 + 648*a^3*b^3*c^4*f*g*h*m*z^
2 - 81*a^2*b^5*c^3*f*g*h*m*z^2 - 3240*a^3*b^2*c^5*d*e*j*m*z^2 + 1620*a^3*b^
2*c^5*d*e*k*l*z^2 + 1377*a^2*b^4*c^4*d*e*j*m*z^2 - 648*a^3*b^2*c^5*e*f*j*k*
z^2 - 648*a^3*b^2*c^5*d*f*j*l*z^2 - 648*a^2*b^4*c^4*d*e*k*l*z^2 - 324*a^3*b
^2*c^5*d*g*j*k*z^2 + 81*a^2*b^4*c^4*e*f*j*k*z^2 + 81*a^2*b^4*c^4*d*f*j*l*z^
2 + 972*a^3*b^2*c^5*e*f*h*l*z^2 - 648*a^3*b^2*c^5*f*g*h*j*z^2 - 324*a^3*b^2
*c^5*e*g*h*k*z^2 - 324*a^3*b^2*c^5*d*g*h*l*z^2 - 162*a^2*b^4*c^4*e*f*h*l*z^
2 + 81*a^2*b^4*c^4*f*g*h*j*z^2 + 81*a^2*b^4*c^4*e*g*h*k*z^2 + 81*a^2*b^4*c^
4*d*g*h*l*z^2 - 648*a^2*b^3*c^5*d*e*f*m*z^2 + 486*a^2*b^3*c^5*d*e*h*k*z^2 +
486*a^2*b^3*c^5*d*e*g*l*z^2 + 162*a^2*b^3*c^5*d*f*g*k*z^2 + 648*a^2*b^2*c^
6*d*e*f*j*z^2 - 324*a^2*b^2*c^6*d*e*g*h*z^2 - 1296*a^6*b*c^3*k*l*m^2*z^2 -
81*a^4*b^5*c*k*l*m^2*z^2 - 1296*a^5*b*c^4*j^2*k*l*z^2 - 324*a^5*b*c^4*h^2*l
*m*z^2 + 324*a^5*b*c^4*h*k^2*l*z^2 - 324*a^5*b*c^4*g*k^2*m*z^2 + 972*a^5*b*
c^4*h*j*l^2*z^2 + 324*a^5*b*c^4*g*k*l^2*z^2 - 324*a^5*b*c^4*e*l^2*m*z^2 - 3
24*a^4*b*c^5*e^2*l*m*z^2 - 1944*a^5*b*c^4*f*j*m^2*z^2 + 1296*a^5*b*c^4*e*k*
m^2*z^2 + 1296*a^5*b*c^4*d*l*m^2*z^2 + 648*a^4*b*c^5*f^2*j*m*z^2 + 81*a^2*b
^7*c*f*j*m^2*z^2 + 1296*a^5*b*c^4*g*h*m^2*z^2 - 324*a^4*b*c^5*g^2*j*k*z^2 +
324*a^4*b*c^5*g^2*h*l*z^2 + 972*a^4*b*c^5*f*h^2*l*z^2 + 324*a^4*b*c^5*g*h^
2*k*z^2 - 324*a^4*b*c^5*e*h^2*m*z^2 - 324*a^4*b*c^5*d*j*k^2*z^2 - 324*a^3*b
*c^6*d^2*j*k*z^2 + 972*a^4*b*c^5*f*g*k^2*z^2 + 972*a^3*b*c^6*d^2*g*m*z^2 +
324*a^4*b*c^5*e*h*k^2*z^2 + 324*a^3*b*c^6*d^2*h*l*z^2 + 81*a*b^5*c^4*d^2*g*
m*z^2 + 972*a^4*b*c^5*e*f*l^2*z^2 + 324*a^4*b*c^5*d*g*l^2*z^2 - 324*a^3*b*c
^6*e^2*h*j*z^2 + 324*a^3*b*c^6*e^2*g*k*z^2 - 324*a^3*b*c^6*e^2*f*l*z^2 - 12
96*a^4*b*c^5*d*e*m^2*z^2 + 81*a*b^7*c^2*d*e*m^2*z^2 - 324*a^3*b*c^6*d*g^2*j
*z^2 - 81*a*b^4*c^5*d^2*g*j*z^2 + 81*a*b^4*c^5*d^2*e*l*z^2 + 324*a^3*b*c^6*
e*g^2*h*z^2 + 81*a*b^4*c^5*d*e^2*k*z^2 + 1296*a^3*b*c^6*d*e*j^2*z^2 - 324*a
^3*b*c^6*e*f*h^2*z^2 + 324*a^3*b*c^6*d*g*h^2*z^2 + 81*a*b^5*c^4*d*e*j^2*z^2
- 324*a^2*b*c^7*d^2*f*g*z^2 + 324*a^2*b*c^7*d^2*e*h*z^2 + 81*a*b^3*c^6*d^2
*f*g*z^2 - 81*a*b^3*c^6*d^2*e*h*z^2 + 324*a^2*b*c^7*d*e^2*g*z^2 - 81*a*b^3*
c^6*d*e^2*g*z^2 + 1296*a^6*c^4*j*k*l*m*z^2 - 1296*a^5*c^5*f*j*k*l*z^2 - 129
6*a^5*c^5*e*j*k*m*z^2 - 1296*a^5*c^5*d*j*l*m*z^2 - 1296*a^5*c^5*g*h*j*m*z^2
+ 1296*a^5*c^5*e*h*l*m*z^2 + 1296*a^4*c^6*e*f*j*k*z^2 + 1296*a^4*c^6*d*g*j
*k*z^2 + 1296*a^4*c^6*d*f*j*l*z^2 - 1296*a^4*c^6*d*e*k*l*z^2 + 1296*a^4*c^6
*d*e*j*m*z^2 + 1296*a^4*c^6*f*g*h*j*z^2 - 1296*a^4*c^6*e*f*h*l*z^2 - 1296*a

^3*c^7*d*e*f*j*z^2 + 648*a^5*b^3*c^2*k*l*m^2*z^2 + 648*a^4*b^3*c^3*j^2*k*l*
z^2 + 486*a^5*b^2*c^3*h*l^2*m*z^2 - 81*a^4*b^4*c^2*h*l^2*m*z^2 + 81*a^4*b^3
*c^3*h^2*l*m*z^2 - 81*a^3*b^5*c^2*j^2*k*l*z^2 - 162*a^4*b^2*c^4*g^2*k*m*z^2
 - 81*a^4*b^3*c^3*h*k^2*l*z^2 + 81*a^4*b^3*c^3*g*k^2*m*z^2 - 567*a^4*b^3*c^
3*h*j*l^2*z^2 + 486*a^4*b^2*c^4*h^2*j*l*z^2 - 81*a^4*b^3*c^3*g*k*l^2*z^2 +
81*a^4*b^3*c^3*e*l^2*m*z^2 + 81*a^3*b^5*c^2*h*j*l^2*z^2 - 81*a^3*b^4*c^3*h^
2*j*l*z^2 + 81*a^3*b^3*c^4*e^2*l*m*z^2 + 2430*a^4*b^3*c^3*f*j*m^2*z^2 - 226
8*a^4*b^2*c^4*f*j^2*m*z^2 - 810*a^3*b^5*c^2*f*j*m^2*z^2 + 810*a^3*b^4*c^3*f
*j^2*m*z^2 - 648*a^4*b^3*c^3*e*k*m^2*z^2 - 648*a^4*b^3*c^3*d*l*m^2*z^2 - 64
8*a^4*b^2*c^4*h*j^2*k*z^2 - 648*a^4*b^2*c^4*g*j^2*l*z^2 - 162*a^3*b^3*c^4*f
^2*j*m*z^2 + 81*a^3*b^5*c^2*e*k*m^2*z^2 + 81*a^3*b^5*c^2*d*l*m^2*z^2 + 81*a
^3*b^4*c^3*h*j^2*k*z^2 + 81*a^3*b^4*c^3*g*j^2*l*z^2 - 81*a^2*b^6*c^2*f*j^2*
m*z^2 - 648*a^4*b^3*c^3*g*h*m^2*z^2 + 486*a^4*b^2*c^4*g*j*k^2*z^2 - 486*a^4
*b^2*c^4*e*k^2*l*z^2 + 486*a^3*b^2*c^5*d^2*k*m*z^2 - 162*a^4*b^2*c^4*d*k^2*
m*z^2 + 81*a^3*b^5*c^2*g*h*m^2*z^2 - 81*a^3*b^4*c^3*g*j*k^2*z^2 + 81*a^3*b^
4*c^3*e*k^2*l*z^2 + 81*a^3*b^3*c^4*g^2*j*k*z^2 - 81*a^2*b^4*c^4*d^2*k*m*z^2
 + 486*a^4*b^2*c^4*e*j*l^2*z^2 - 486*a^4*b^2*c^4*d*k*l^2*z^2 - 162*a^3*b^2*
c^5*e^2*j*l*z^2 - 81*a^3*b^4*c^3*e*j*l^2*z^2 + 81*a^3*b^4*c^3*d*k*l^2*z^2 -
 81*a^3*b^3*c^4*g^2*h*l*z^2 - 1458*a^4*b^2*c^4*f*h*l^2*z^2 + 648*a^3*b^4*c^
3*f*h*l^2*z^2 - 567*a^3*b^3*c^4*f*h^2*l*z^2 + 486*a^3*b^2*c^5*e^2*h*m*z^2 -
 81*a^3*b^3*c^4*g*h^2*k*z^2 + 81*a^3*b^3*c^4*e*h^2*m*z^2 - 81*a^2*b^6*c^2*f
*h*l^2*z^2 + 81*a^2*b^5*c^3*f*h^2*l*z^2 - 81*a^2*b^4*c^4*e^2*h*m*z^2 - 1296
*a^4*b^2*c^4*e*g*m^2*z^2 - 1296*a^4*b^2*c^4*d*h*m^2*z^2 + 648*a^3*b^4*c^3*e
*g*m^2*z^2 + 648*a^3*b^4*c^3*d*h*m^2*z^2 + 81*a^3*b^3*c^4*d*j*k^2*z^2 - 81*
a^2*b^6*c^2*e*g*m^2*z^2 - 81*a^2*b^6*c^2*d*h*m^2*z^2 + 81*a^2*b^3*c^5*d^2*j
*k*z^2 - 567*a^3*b^3*c^4*f*g*k^2*z^2 - 567*a^2*b^3*c^5*d^2*g*m*z^2 + 486*a^
3*b^2*c^5*f*g^2*k*z^2 - 486*a^3*b^2*c^5*e*g^2*l*z^2 + 486*a^3*b^2*c^5*d*g^2
*m*z^2 - 81*a^3*b^3*c^4*e*h*k^2*z^2 + 81*a^2*b^5*c^3*f*g*k^2*z^2 - 81*a^2*b
^4*c^4*f*g^2*k*z^2 + 81*a^2*b^4*c^4*e*g^2*l*z^2 - 81*a^2*b^4*c^4*d*g^2*m*z^
2 - 81*a^2*b^3*c^5*d^2*h*l*z^2 - 567*a^3*b^3*c^4*e*f*l^2*z^2 - 486*a^3*b^2*
c^5*d*h^2*k*z^2 - 162*a^3*b^2*c^5*e*h^2*j*z^2 - 81*a^3*b^3*c^4*d*g*l^2*z^2
+ 81*a^2*b^5*c^3*e*f*l^2*z^2 + 81*a^2*b^4*c^4*d*h^2*k*z^2 + 81*a^2*b^3*c^5*
e^2*h*j*z^2 - 81*a^2*b^3*c^5*e^2*g*k*z^2 + 81*a^2*b^3*c^5*e^2*f*l*z^2 + 194
4*a^3*b^3*c^4*d*e*m^2*z^2 - 729*a^2*b^5*c^3*d*e*m^2*z^2 + 648*a^3*b^2*c^5*e
*g*j^2*z^2 + 648*a^3*b^2*c^5*d*h*j^2*z^2 - 81*a^2*b^4*c^4*e*g*j^2*z^2 - 81*
a^2*b^4*c^4*d*h*j^2*z^2 + 486*a^3*b^2*c^5*d*f*k^2*z^2 + 486*a^2*b^2*c^6*d^2
*g*j*z^2 - 486*a^2*b^2*c^6*d^2*e*l*z^2 - 162*a^2*b^2*c^6*d^2*f*k*z^2 - 81*a
^2*b^4*c^4*d*f*k^2*z^2 + 81*a^2*b^3*c^5*d*g^2*j*z^2 - 486*a^2*b^2*c^6*d*e^2
*k*z^2 - 81*a^2*b^3*c^5*e*g^2*h*z^2 - 648*a^2*b^3*c^5*d*e*j^2*z^2 - 162*a^2
*b^2*c^6*e^2*f*h*z^2 + 81*a^2*b^3*c^5*e*f*h^2*z^2 - 81*a^2*b^3*c^5*d*g*h^2*
z^2 - 162*a^2*b^2*c^6*d*f*g^2*z^2 - 189*a^5*b^3*c^2*l^3*m*z^2 + 162*a^5*b^2
*c^3*k^3*m*z^2 - 27*a^4*b^4*c^2*k^3*m*z^2 - 702*a^4*b^3*c^3*j^3*m*z^2 - 81*
a^3*b^6*c*j^2*m^2*z^2 + 81*a^3*b^5*c^2*j^3*m*z^2 - 54*a^5*b^3*c^2*j*m^3*z^2
 - 486*a^5*b^2*c^3*j*l^3*z^2 + 216*a^4*b^4*c^2*j*l^3*z^2 - 189*a^4*b^3*c^3*
j*k^3*z^2 - 54*a^4*b^2*c^4*h^3*m*z^2 + 27*a^3*b^5*c^2*j*k^3*z^2 + 27*a^3*b^

3*c^4*g^3*m*z^2 - 810*a^4*b^4*c^2*f*m^3*z^2 + 540*a^5*b^2*c^3*f*m^3*z^2 - 3
24*a^3*b^2*c^5*f^3*m*z^2 + 54*a^2*b^4*c^4*f^3*m*z^2 + 675*a^4*b^3*c^3*f*l^3
*z^2 - 243*a^3*b^5*c^2*f*l^3*z^2 - 189*a^2*b^3*c^5*e^3*m*z^2 + 27*a^3*b^3*c
^4*h^3*j*z^2 - 486*a^4*b^2*c^4*f*k^3*z^2 - 486*a^2*b^2*c^6*d^3*m*z^2 + 216*
a^3*b^4*c^3*f*k^3*z^2 - 54*a^3*b^2*c^5*g^3*j*z^2 - 27*a^2*b^6*c^2*f*k^3*z^2
 - 270*a^3*b^3*c^4*f*j^3*z^2 - 54*a^2*b^3*c^5*f^3*j*z^2 + 27*a^2*b^5*c^3*f*
j^3*z^2 + 162*a^2*b^2*c^6*e^3*j*z^2 + 162*a^3*b^2*c^5*f*h^3*z^2 - 27*a^2*b^
4*c^4*f*h^3*z^2 + 27*a^2*b^3*c^5*f*g^3*z^2 + 81*a*b^2*c^7*d^2*e^2*z^2 - 648
*a^6*c^4*h*l^2*m*z^2 + 648*a^5*c^5*g^2*k*m*z^2 - 648*a^5*c^5*h^2*j*l*z^2 +
1296*a^5*c^5*h*j^2*k*z^2 + 1296*a^5*c^5*g*j^2*l*z^2 + 1296*a^5*c^5*f*j^2*m*
z^2 - 648*a^5*c^5*g*j*k^2*z^2 + 648*a^5*c^5*e*k^2*l*z^2 + 648*a^5*c^5*d*k^2
*m*z^2 - 648*a^4*c^6*d^2*k*m*z^2 - 648*a^5*c^5*e*j*l^2*z^2 + 648*a^5*c^5*d*
k*l^2*z^2 + 648*a^4*c^6*e^2*j*l*z^2 + 324*a^6*b*c^3*l^3*m*z^2 + 27*a^4*b^5*
c*l^3*m*z^2 + 648*a^5*c^5*f*h*l^2*z^2 - 648*a^4*c^6*e^2*h*m*z^2 + 1512*a^5*
b*c^4*j^3*m*z^2 + 1080*a^6*b*c^3*j*m^3*z^2 - 162*a^4*b^5*c*j*m^3*z^2 - 648*
a^4*c^6*f*g^2*k*z^2 + 648*a^4*c^6*e*g^2*l*z^2 - 648*a^4*c^6*d*g^2*m*z^2 - 2
7*a^3*b^6*c*j*l^3*z^2 + 648*a^4*c^6*e*h^2*j*z^2 + 648*a^4*c^6*d*h^2*k*z^2 +
 324*a^5*b*c^4*j*k^3*z^2 - 1296*a^4*c^6*e*g*j^2*z^2 - 1296*a^4*c^6*d*h*j^2*
z^2 - 108*a^4*b*c^5*g^3*m*z^2 - 648*a^4*c^6*d*f*k^2*z^2 - 648*a^3*c^7*d^2*g
*j*z^2 + 648*a^3*c^7*d^2*f*k*z^2 + 648*a^3*c^7*d^2*e*l*z^2 + 270*a^3*b^6*c*
f*m^3*z^2 + 648*a^3*c^7*d*e^2*k*z^2 - 540*a^5*b*c^4*f*l^3*z^2 + 324*a^3*b*c
^6*e^3*m*z^2 - 108*a^4*b*c^5*h^3*j*z^2 + 27*a^2*b^7*c*f*l^3*z^2 + 27*a*b^5*
c^4*e^3*m*z^2 + 648*a^3*c^7*e^2*f*h*z^2 + 216*a*b^4*c^5*d^3*m*z^2 + 648*a^4
*b*c^5*f*j^3*z^2 + 216*a^3*b*c^6*f^3*j*z^2 + 648*a^3*c^7*d*f*g^2*z^2 - 27*a
*b^4*c^5*e^3*j*z^2 + 324*a^2*b*c^7*d^3*j*z^2 - 189*a*b^3*c^6*d^3*j*z^2 - 10
8*a^3*b*c^6*f*g^3*z^2 - 108*a^2*b*c^7*e^3*f*z^2 + 27*a*b^3*c^6*e^3*f*z^2 +
162*a*b^2*c^7*d^3*f*z^2 - 1134*a^5*b^2*c^3*j^2*m^2*z^2 + 648*a^4*b^4*c^2*j^
2*m^2*z^2 + 81*a^5*b^2*c^3*k^2*l^2*z^2 + 162*a^4*b^2*c^4*f^2*m^2*z^2 + 81*a
^4*b^2*c^4*h^2*k^2*z^2 + 81*a^4*b^2*c^4*g^2*l^2*z^2 + 162*a^3*b^2*c^5*f^2*j
^2*z^2 + 81*a^3*b^2*c^5*e^2*k^2*z^2 + 81*a^3*b^2*c^5*d^2*l^2*z^2 + 81*a^3*b
^2*c^5*g^2*h^2*z^2 + 81*a^2*b^2*c^6*e^2*g^2*z^2 + 81*a^2*b^2*c^6*d^2*h^2*z^
2 - 216*a^6*c^4*k^3*m*z^2 + 216*a^6*c^4*j*l^3*z^2 + 27*a^3*b^7*j*m^3*z^2 +
216*a^5*c^5*h^3*m*z^2 + 432*a^6*c^4*f*m^3*z^2 + 432*a^4*c^6*f^3*m*z^2 - 27*
b^6*c^4*d^3*m*z^2 - 27*a^2*b^8*f*m^3*z^2 + 216*a^5*c^5*f*k^3*z^2 + 216*a^4*
c^6*g^3*j*z^2 + 216*a^3*c^7*d^3*m*z^2 + 216*a^5*b^4*c*m^4*z^2 - 216*a^3*c^7
*e^3*j*z^2 + 27*b^5*c^5*d^3*j*z^2 - 216*a^4*c^6*f*h^3*z^2 - 27*b^4*c^6*d^3*
f*z^2 - 216*a^2*c^8*d^3*f*z^2 - 648*a^6*c^4*j^2*m^2*z^2 - 324*a^6*c^4*k^2*l
^2*z^2 - 648*a^5*c^5*f^2*m^2*z^2 - 324*a^5*c^5*h^2*k^2*z^2 - 324*a^5*c^5*g^
2*l^2*z^2 - 648*a^4*c^6*f^2*j^2*z^2 - 324*a^4*c^6*e^2*k^2*z^2 - 324*a^4*c^6
*d^2*l^2*z^2 - 405*a^6*b^2*c^2*m^4*z^2 - 324*a^4*c^6*g^2*h^2*z^2 - 324*a^3*
c^7*e^2*g^2*z^2 - 324*a^3*c^7*d^2*h^2*z^2 + 243*a^4*b^2*c^4*j^4*z^2 - 27*a^
3*b^4*c^3*j^4*z^2 - 324*a^2*c^8*d^2*e^2*z^2 + 27*a^2*b^2*c^6*f^4*z^2 - 108*
a^7*c^3*m^4*z^2 - 27*a^4*b^6*m^4*z^2 - 540*a^5*c^5*j^4*z^2 - 108*a^3*c^7*f^
4*z^2 - 216*a^5*b*c^3*f*j*k*l*m*z - 54*a^3*b^5*c*f*j*k*l*m*z + 27*a^3*b^5*c
*g*h*k*l*m*z - 27*a^2*b^6*c*e*g*k*l*m*z - 27*a^2*b^6*c*d*h*k*l*m*z + 432*a^

4*b*c^4*d*g*j*k*m*z - 432*a^4*b*c^4*d*e*k*l*m*z + 216*a^4*b*c^4*e*g*j*k*l*z + 216*a^4*b*c^4*e*f*j*k*m*z + 216*a^4*b*c^4*d*h*j*k*l*z + 216*a^4*b*c^4*d*f*j*l*m*z + 216*a^4*b*c^4*f*g*h*j*m*z - 27*a*b^6*c^2*d*e*j*k*l*z - 27*a*b^6*c^2*d*e*h*k*m*z - 27*a*b^6*c^2*d*e*g*l*m*z + 216*a^3*b*c^5*d*e*h*j*k*z + 216*a^3*b*c^5*d*e*g*j*l*z - 216*a^3*b*c^5*d*e*f*j*m*z + 27*a*b^5*c^3*d*e*h*j*k*z + 27*a*b^5*c^3*d*e*g*j*l*z + 27*a*b^5*c^3*d*e*g*h*m*z - 27*a*b^4*c^4*d*e*g*h*j*z + 27*a*b^7*c*d*e*k*l*m*z + 270*a^4*b^3*c^2*f*j*k*l*m*z - 108*a^4*b^3*c^2*g*h*k*l*m*z - 216*a^4*b^2*c^3*f*h*j*k*m*z - 216*a^4*b^2*c^3*f*g*j*l*m*z - 216*a^4*b^2*c^3*e*g*k*l*m*z - 216*a^4*b^2*c^3*d*h*k*l*m*z + 162*a^3*b^4*c^2*e*g*k*l*m*z + 162*a^3*b^4*c^2*d*h*k*l*m*z + 108*a^4*b^2*c^3*g*h*j*k*l*z + 108*a^4*b^2*c^3*e*h*j*l*m*z + 54*a^3*b^4*c^2*f*h*j*k*m*z + 54*a^3*b^4*c^2*f*g*j*l*m*z - 27*a^3*b^4*c^2*g*h*j*k*l*z + 540*a^3*b^3*c^3*d*e*k*l*m*z - 216*a^2*b^5*c^2*d*e*k*l*m*z - 162*a^3*b^3*c^3*e*g*j*k*l*z - 162*a^3*b^3*c^3*d*h*j*k*l*z - 108*a^3*b^3*c^3*d*g*j*k*m*z - 54*a^3*b^3*c^3*e*f*j*k*m*z - 54*a^3*b^3*c^3*d*f*j*l*m*z + 27*a^2*b^5*c^2*e*g*j*k*l*z + 27*a^2*b^5*c^2*d*h*j*k*l*z - 108*a^3*b^3*c^3*e*g*h*k*m*z - 108*a^3*b^3*c^3*d*g*h*l*m*z - 54*a^3*b^3*c^3*f*g*h*j*m*z + 27*a^2*b^5*c^2*e*g*h*k*m*z + 27*a^2*b^5*c^2*d*g*h*l*m*z - 540*a^3*b^2*c^4*d*e*j*k*l*z + 216*a^2*b^4*c^3*d*e*j*k*l*z - 216*a^3*b^2*c^4*d*e*h*k*m*z - 216*a^3*b^2*c^4*d*e*g*l*m*z + 162*a^2*b^4*c^3*d*e*h*k*m*z + 162*a^2*b^4*c^3*d*e*g*l*m*z + 108*a^3*b^2*c^4*e*g*h*j*k*z - 108*a^3*b^2*c^4*e*f*h*j*l*z + 108*a^3*b^2*c^4*d*g*h*j*l*z + 108*a^3*b^2*c^4*d*f*g*k*m*z - 27*a^2*b^4*c^3*e*g*h*j*k*z - 27*a^2*b^4*c^3*d*g*h*j*l*z - 162*a^2*b^3*c^4*d*e*h*j*k*z - 162*a^2*b^3*c^4*d*e*g*j*l*z + 54*a^2*b^3*c^4*d*e*f*j*m*z - 108*a^2*b^3*c^4*d*e*g*h*m*z + 108*a^2*b^2*c^5*d*e*g*h*j*z + 324*a^6*b*c^2*j*k*l*m^2*z - 81*a^5*b^3*c*j*k*l*m^2*z + 27*a^4*b^4*c*j^2*k*l*m*z - 27*a^4*b^4*c*h*k^2*l*m*z - 27*a^4*b^4*c*g*k*l^2*m*z + 216*a^5*b*c^3*h*j^2*k*m*z + 216*a^5*b*c^3*g*j^2*l*m*z + 54*a^4*b^4*c*f*k*l*m^2*z + 27*a^4*b^4*c*h*j*k*m^2*z + 27*a^4*b^4*c*g*j*l*m^2*z + 27*a^2*b^6*c*f^2*k*l*m*z + 216*a^5*b*c^3*e*k^2*l*m*z - 108*a^5*b*c^3*h*j*k^2*l*z + 27*a^3*b^5*c*e*k^2*l*m*z + 216*a^5*b*c^3*d*k*l^2*m*z + 216*a^4*b*c^4*e^2*j*l*m*z - 108*a^5*b*c^3*g*j*k*l^2*z + 27*a^3*b^5*c*d*k*l^2*m*z - 324*a^5*b*c^3*e*j*k*m^2*z - 324*a^5*b*c^3*d*j*l*m^2*z - 216*a^5*b*c^3*f*h*l^2*m*z - 108*a^4*b*c^4*f^2*j*k*l*z - 27*a^3*b^5*c*e*j*k*m^2*z - 27*a^3*b^5*c*d*j*l*m^2*z - 324*a^5*b*c^3*g*h*j*m^2*z + 216*a^5*b*c^3*f*h*k*m^2*z + 216*a^5*b*c^3*f*g*l*m^2*z + 216*a^5*b*c^3*e*h*l*m^2*z - 216*a^4*b*c^4*f^2*h*k*m*z - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5*c*g*h*j*m^2*z + 216*a^4*b*c^4*e*g^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 216*a^4*b*c^4*f*h^2*j*l*z + 216*a^4*b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k*m*z - 108*a^4*b*c^4*g*h^2*j*k*z - 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^4*d*h*j^2*m*z + 216*a^4*b*c^4*f*h*j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*a^2*b^6*c*e*g*j*m^2*z + 27*a^2*b^6*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*z - 216*a^4*b*c^4*f*g*j*k^2*z + 216*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d^2*e*l*m*z - 108*a^4*b*c^4*e*h*j*k^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^3*b*c^5*d^2*h*j*l*z + 108*a^3*b*c^5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z + 27*a*b^5*c^3*d^2*g*k*l*z + 27*a*b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j*l^2*z + 216*a^3*b*c^5*d*e^2*k*m*z - 108*a^4*b*c^4*d*g*j*l^2*z - 10

8*a^3*b*c^5*e^2*g*j*k*z + 27*a*b^5*c^3*d*e^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^
2*z + 216*a^3*b*c^5*e^2*f*h*m*z - 108*a^4*b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5
*e^2*g*h*l*z + 108*a^3*b*c^5*e*f^2*j*k*z + 108*a^3*b*c^5*d*f^2*j*l*z + 27*a
*b^6*c^2*d*e*j^2*m*z - 216*a^3*b*c^5*e*f^2*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*
z - 27*a*b^4*c^4*d^2*e*j*l*z + 216*a^3*b*c^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*
g^2*h*j*z + 54*a*b^4*c^4*d^2*f*g*m*z - 27*a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*
c^4*d^2*e*h*m*z - 27*a*b^4*c^4*d*e^2*j*k*z - 108*a^3*b*c^5*d*g*h^2*j*z + 54
*a*b^4*c^4*d*e^2*h*l*z + 27*a*b^6*c^2*d*e*h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*
z - 27*a*b^4*c^4*d*e^2*g*m*z - 27*a*b^4*c^4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2
*f*g*j*z - 108*a^3*b*c^5*d*e*g*k^2*z - 108*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*
b*c^6*d^2*e*g*k*z - 54*a*b^3*c^5*d^2*f*g*j*z - 27*a*b^5*c^3*d*e*g*k^2*z + 2
7*a*b^4*c^4*d*e*g^2*k*z + 27*a*b^3*c^5*d^2*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k
*z - 108*a^2*b*c^6*d*e^2*g*j*z + 27*a*b^3*c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d
*e*f^2*j*z + 27*a*b^3*c^5*d*e*f^2*j*z - 432*a^5*c^4*e*h*j*l*m*z + 432*a^4*c
^5*d*e*j*k*l*z + 432*a^4*c^5*e*f*h*j*l*z - 432*a^4*c^5*d*f*g*k*m*z - 27*a*b
^7*c*d*e*j*m^2*z - 54*a^5*b^2*c^2*j^2*k*l*m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z
 + 108*a^5*b^2*c^2*g*k*l^2*m*z - 54*a^5*b^2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c
^3*f^2*k*l*m*z - 270*a^5*b^2*c^2*f*k*l*m^2*z - 189*a^3*b^4*c^2*f^2*k*l*m*z
 - 108*a^5*b^2*c^2*h*j*k*m^2*z - 108*a^5*b^2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^
2*h*j^2*k*m*z - 54*a^4*b^3*c^2*g*j^2*l*m*z - 162*a^4*b^3*c^2*e*k^2*l*m*z +
54*a^4*b^2*c^3*g^2*j*k*m*z + 27*a^4*b^3*c^2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d
*k*l^2*m*z + 108*a^4*b^2*c^3*g^2*h*l*m*z - 54*a^3*b^3*c^3*e^2*j*l*m*z + 27*
a^4*b^3*c^2*g*j*k*l^2*z - 27*a^3*b^4*c^2*g^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^
2*k*l*z + 189*a^4*b^3*c^2*e*j*k*m^2*z + 189*a^4*b^3*c^2*d*j*l*m^2*z - 162*a
^4*b^2*c^3*e*j^2*k*m*z - 162*a^4*b^2*c^3*d*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*
j*k*l*z + 108*a^4*b^2*c^3*g*h^2*k*m*z + 54*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4
*b^2*c^3*f*h^2*l*m*z + 54*a^3*b^4*c^2*f*j^2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*
m*z + 27*a^3*b^4*c^2*e*j^2*k*m*z + 27*a^3*b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*
c^2*f^2*j*k*l*z - 270*a^3*b^2*c^4*d^2*j*k*m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z
 - 162*a^4*b^2*c^3*g*h*j^2*m*z + 162*a^4*b^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*
c^3*f^2*h*k*m*z + 162*a^3*b^3*c^3*f^2*g*l*m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z
 - 54*a^4*b^3*c^2*f*g*l*m^2*z - 54*a^4*b^3*c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*
d*j*k^2*m*z + 54*a^2*b^4*c^3*d^2*j*k*m*z + 27*a^3*b^4*c^2*g*h*j^2*m*z - 27*
a^3*b^4*c^2*e*j*k^2*l*z - 27*a^2*b^5*c^2*f^2*h*k*m*z - 27*a^2*b^5*c^2*f^2*g
*l*m*z + 162*a^4*b^2*c^3*d*j*k*l^2*z - 162*a^3*b^3*c^3*e*g^2*l*m*z + 108*a^
4*b^2*c^3*e*h*k^2*m*z + 108*a^3*b^2*c^4*d^2*h*l*m*z - 54*a^4*b^2*c^3*f*g*k^
2*m*z - 27*a^3*b^4*c^2*e*h*k^2*m*z - 27*a^3*b^4*c^2*d*j*k*l^2*z + 27*a^3*b^
3*c^3*g^2*h*j*l*z + 27*a^2*b^5*c^2*e*g^2*l*m*z - 27*a^2*b^4*c^3*d^2*h*l*m*z
 + 270*a^4*b^2*c^3*f*h*j*l^2*z - 270*a^3*b^2*c^4*e^2*h*j*m*z - 162*a^4*b^2*
c^3*e*h*k*l^2*z - 162*a^3*b^3*c^3*d*h^2*k*m*z + 162*a^3*b^2*c^4*e^2*h*k*l*z
 + 108*a^4*b^2*c^3*d*g*l^2*m*z + 108*a^3*b^2*c^4*e^2*g*k*m*z - 54*a^4*b^2*c
^3*e*f*l^2*m*z - 54*a^3*b^4*c^2*f*h*j*l^2*z + 54*a^3*b^3*c^3*f*h^2*j*l*z -
54*a^3*b^3*c^3*e*h^2*j*m*z + 54*a^3*b^2*c^4*e^2*f*l*m*z + 54*a^2*b^4*c^3*e^
2*h*j*m*z + 27*a^3*b^4*c^2*e*h*k*l^2*z - 27*a^3*b^4*c^2*d*g*l^2*m*z + 27*a^
3*b^3*c^3*g*h^2*j*k*z + 27*a^2*b^5*c^2*d*h^2*k*m*z - 27*a^2*b^4*c^3*e^2*h*k

*l*z - 27*a^2*b^4*c^3*e^2*g*k*m*z + 432*a^4*b^2*c^3*e*g*j*m^2*z + 432*a^4*b
^2*c^3*d*h*j*m^2*z - 270*a^4*b^2*c^3*d*g*k*m^2*z - 216*a^3*b^4*c^2*e*g*j*m^
2*z - 216*a^3*b^4*c^2*d*h*j*m^2*z + 216*a^3*b^3*c^3*e*g*j^2*m*z + 216*a^3*b
^3*c^3*d*h*j^2*m*z - 162*a^3*b^2*c^4*e*f^2*k*m*z - 162*a^3*b^2*c^4*d*f^2*l*
m*z - 108*a^3*b^2*c^4*f^2*h*j*k*z - 108*a^3*b^2*c^4*f^2*g*j*l*z + 54*a^4*b^
2*c^3*e*f*k*m^2*z + 54*a^4*b^2*c^3*d*f*l*m^2*z + 54*a^3*b^4*c^2*d*g*k*m^2*z
 - 54*a^3*b^3*c^3*f*h*j^2*k*z - 54*a^3*b^3*c^3*f*g*j^2*l*z - 27*a^2*b^5*c^2
*e*g*j^2*m*z - 27*a^2*b^5*c^2*d*h*j^2*m*z + 27*a^2*b^4*c^3*f^2*h*j*k*z + 27
*a^2*b^4*c^3*f^2*g*j*l*z + 27*a^2*b^4*c^3*e*f^2*k*m*z + 27*a^2*b^4*c^3*d*f^
2*l*m*z + 324*a^2*b^3*c^4*d^2*g*j*m*z - 270*a^3*b^2*c^4*d*g^2*j*m*z - 162*a
^3*b^2*c^4*f^2*g*h*m*z + 162*a^3*b^2*c^4*e*g^2*j*l*z - 162*a^2*b^3*c^4*d^2*
e*l*m*z - 135*a^2*b^3*c^4*d^2*g*k*l*z + 108*a^3*b^2*c^4*d*g^2*k*l*z + 54*a^
4*b^2*c^3*f*g*h*m^2*z + 54*a^3*b^3*c^3*f*g*j*k^2*z - 54*a^3*b^2*c^4*f*g^2*j
*k*z + 54*a^2*b^4*c^3*d*g^2*j*m*z - 54*a^2*b^3*c^4*d^2*f*k*m*z + 27*a^3*b^3
*c^3*e*h*j*k^2*z + 27*a^3*b^3*c^3*d*g*k^2*l*z + 27*a^2*b^4*c^3*f^2*g*h*m*z
- 27*a^2*b^4*c^3*e*g^2*j*l*z - 27*a^2*b^4*c^3*d*g^2*k*l*z + 27*a^2*b^3*c^4*
d^2*h*j*l*z + 162*a^3*b^2*c^4*d*h^2*j*k*z - 162*a^2*b^3*c^4*d*e^2*k*m*z + 1
08*a^3*b^2*c^4*e*g^2*h*m*z + 54*a^3*b^3*c^3*e*f*j*l^2*z + 27*a^3*b^3*c^3*d*
g*j*l^2*z - 27*a^2*b^4*c^3*e*g^2*h*m*z - 27*a^2*b^4*c^3*d*h^2*j*k*z + 27*a^
2*b^3*c^4*e^2*g*j*k*z - 621*a^3*b^3*c^3*d*e*j*m^2*z + 594*a^3*b^2*c^4*d*e*j
^2*m*z + 243*a^2*b^5*c^2*d*e*j*m^2*z - 243*a^2*b^4*c^3*d*e*j^2*m*z + 135*a^
3*b^3*c^3*e*g*h*l^2*z - 108*a^3*b^2*c^4*e*g*h^2*l*z + 108*a^3*b^2*c^4*d*g*h
^2*m*z + 54*a^3*b^2*c^4*e*f*j^2*k*z + 54*a^3*b^2*c^4*e*f*h^2*m*z + 54*a^3*b
^2*c^4*d*g*j^2*k*z + 54*a^3*b^2*c^4*d*f*j^2*l*z - 54*a^2*b^3*c^4*e^2*f*h*m*
z - 27*a^2*b^5*c^2*e*g*h*l^2*z + 27*a^2*b^4*c^3*e*g*h^2*l*z - 27*a^2*b^4*c^
3*d*g*h^2*m*z - 27*a^2*b^3*c^4*e^2*g*h*l*z - 27*a^2*b^3*c^4*e*f^2*j*k*z - 2
7*a^2*b^3*c^4*d*f^2*j*l*z + 162*a^2*b^2*c^5*d^2*e*j*l*z + 54*a^3*b^2*c^4*f*
g*h*j^2*z - 54*a^3*b^2*c^4*d*f*j*k^2*z + 54*a^2*b^3*c^4*e*f^2*h*l*z + 54*a^
2*b^2*c^5*d^2*f*j*k*z - 27*a^2*b^3*c^4*f^2*g*h*j*z - 270*a^2*b^2*c^5*d^2*f*
g*m*z - 162*a^3*b^2*c^4*d*g*h*k^2*z + 162*a^2*b^2*c^5*d^2*g*h*k*z + 162*a^2
*b^2*c^5*d*e^2*j*k*z + 108*a^2*b^2*c^5*d^2*e*h*m*z - 54*a^2*b^3*c^4*d*f*g^2
*m*z + 27*a^2*b^4*c^3*d*g*h*k^2*z + 27*a^2*b^3*c^4*e*g^2*h*j*z + 270*a^3*b^
2*c^4*d*e*h*l^2*z - 270*a^2*b^2*c^5*d*e^2*h*l*z - 162*a^2*b^4*c^3*d*e*h*l^2
*z + 108*a^2*b^3*c^4*d*e*h^2*l*z + 108*a^2*b^2*c^5*d*e^2*g*m*z + 54*a^2*b^2
*c^5*e^2*f*h*j*z + 27*a^2*b^3*c^4*d*g*h^2*j*z + 162*a^2*b^2*c^5*d*e*f^2*m*z
 - 54*a^3*b^2*c^4*d*e*f*m^2*z - 54*a^2*b^2*c^5*d*f^2*g*k*z + 135*a^2*b^3*c^
4*d*e*g*k^2*z - 108*a^2*b^2*c^5*d*e*g^2*k*z + 54*a^2*b^2*c^5*d*f*g^2*j*z -
54*a^2*b^2*c^5*d*e*f*j^2*z - 9*a*b^7*c*d*e*l^3*z - 36*a*b*c^7*d^3*e*g*z - 1
08*a^6*b*c^2*k^2*l^2*m*z + 27*a^5*b^3*c*k^2*l^2*m*z - 18*a^5*b^2*c^2*j*k^3*
m*z - 27*a^4*b^3*c^2*j^3*k*l*z - 108*a^5*b*c^3*h^2*k^2*m*z - 108*a^5*b*c^3*
g^2*l^2*m*z + 108*a^5*b*c^3*h^2*k*l^2*z + 108*a^5*b*c^3*g^2*k*m^2*z + 90*a^
5*b^2*c^2*f*l^3*m*z - 18*a^5*b^2*c^2*h*k*l^3*z + 18*a^4*b^2*c^3*h^3*k*l*z +
 18*a^4*b^2*c^3*h^3*j*m*z - 108*a^5*b*c^3*h*j^2*l^2*z + 18*a^4*b^3*c^2*f*k^
3*m*z - 18*a^3*b^3*c^3*g^3*j*m*z - 9*a^4*b^3*c^2*g*k^3*l*z + 9*a^3*b^3*c^3*
g^3*k*l*z + 252*a^4*b^2*c^3*f*j^3*m*z + 216*a^5*b*c^3*f*j^2*m^2*z + 180*a^3

*b^2*c^4*f^3*j*m*z - 108*a^4*b*c^4*e^2*k^2*m*z - 108*a^4*b*c^4*d^2*l^2*m*z
+ 90*a^5*b^2*c^2*e*k*m^3*z + 90*a^5*b^2*c^2*d*l*m^3*z - 90*a^3*b^2*c^4*f^3*
k*l*z + 54*a^3*b^5*c*f*j^2*m^2*z - 54*a^3*b^4*c^2*f*j^3*m*z + 36*a^5*b^2*c^
2*f*j*m^3*z + 36*a^4*b^2*c^3*h*j^3*k*z + 36*a^4*b^2*c^3*g*j^3*l*z - 36*a^2*
b^4*c^3*f^3*j*m*z - 27*a^2*b^6*c*f^2*j*m^2*z + 18*a^2*b^4*c^3*f^3*k*l*z - 2
16*a^4*b*c^4*d^2*k*m^2*z + 108*a^5*b*c^3*d*k^2*m^2*z - 108*a^4*b^3*c^2*f*j*
l^3*z - 108*a^4*b*c^4*g^2*h^2*m*z + 108*a^2*b^3*c^4*e^3*j*m*z + 90*a^5*b^2*
c^2*g*h*m^3*z + 54*a^4*b^3*c^2*e*k*l^3*z - 54*a^2*b^3*c^4*e^3*k*l*z + 234*a
^2*b^2*c^5*d^3*j*m*z - 144*a^2*b^2*c^5*d^3*k*l*z + 90*a^4*b^2*c^3*f*j*k^3*z
 - 72*a^4*b^2*c^3*d*k^3*l*z + 27*a^4*b^3*c^2*g*h*l^3*z - 27*a^3*b^3*c^3*g*h
^3*l*z - 18*a^3*b^4*c^2*f*j*k^3*z + 9*a^3*b^4*c^2*d*k^3*l*z + 216*a^4*b*c^4
*f^2*h*l^2*z - 216*a^4*b*c^4*e^2*h*m^2*z + 108*a^4*b*c^4*g^2*h*k^2*z - 18*a
^4*b^2*c^3*g*h*k^3*z + 18*a^3*b^2*c^4*g^3*h*k*z + 18*a^3*b^2*c^4*f*g^3*m*z
+ 9*a^3*b^4*c^2*g*h*k^3*z - 9*a^3*b^3*c^3*e*j^3*k*z - 9*a^3*b^3*c^3*d*j^3*l
*z - 144*a^4*b^3*c^2*e*g*m^3*z - 144*a^4*b^3*c^2*d*h*m^3*z - 108*a^3*b*c^5*
e^2*g^2*m*z + 108*a^3*b*c^5*d^2*j^2*k*z - 108*a^3*b*c^5*d^2*h^2*m*z - 18*a^
2*b^3*c^4*f^3*h*k*z - 18*a^2*b^3*c^4*f^3*g*l*z - 9*a^3*b^3*c^3*g*h*j^3*z -
216*a^4*b*c^4*d*g^2*m^2*z + 144*a^4*b^2*c^3*e*g*l^3*z - 126*a^3*b^2*c^4*d*h
^3*l*z - 108*a^4*b*c^4*d*h^2*l^2*z - 108*a^3*b*c^5*f^2*g^2*k*z - 108*a^3*b*
c^5*e^2*h^2*k*z - 90*a^2*b^2*c^5*e^3*f*m*z + 72*a^2*b^2*c^5*e^3*g*l*z - 63*
a^3*b^4*c^2*e*g*l^3*z - 36*a^3*b^4*c^2*d*h*l^3*z + 27*a^2*b^4*c^3*d*h^3*l*z
 + 27*a*b^6*c^2*d^2*g*m^2*z - 18*a^4*b^2*c^3*d*h*l^3*z - 18*a^3*b^2*c^4*f*h
^3*j*z - 18*a^3*b^2*c^4*e*h^3*k*z + 18*a^2*b^2*c^5*e^3*h*k*z + 108*a^3*b*c^
5*e^2*h*j^2*z + 54*a^3*b^3*c^3*d*h*k^3*z + 27*a^3*b^3*c^3*e*g*k^3*z - 27*a^
2*b^3*c^4*e*g^3*k*z + 27*a^2*b^3*c^4*d*g^3*l*z - 27*a*b^4*c^4*d^2*g^2*l*z -
 9*a^2*b^5*c^2*e*g*k^3*z - 9*a^2*b^5*c^2*d*h*k^3*z + 207*a^3*b^4*c^2*d*e*m^
3*z - 108*a^2*b*c^6*d^2*e^2*m*z - 90*a^4*b^2*c^3*d*e*m^3*z - 72*a^3*b^2*c^4
*e*g*j^3*z - 72*a^3*b^2*c^4*d*h*j^3*z + 27*a*b^3*c^5*d^2*e^2*m*z + 18*a^2*b
^2*c^5*e*f^3*k*z + 18*a^2*b^2*c^5*d*f^3*l*z + 9*a^2*b^4*c^3*e*g*j^3*z + 9*a
^2*b^4*c^3*d*h*j^3*z - 216*a^3*b*c^5*d*e^2*l^2*z - 198*a^3*b^3*c^3*d*e*l^3*
z + 108*a^3*b*c^5*d*g^2*j^2*z - 108*a^3*b*c^5*d*f^2*k^2*z + 72*a^2*b^5*c^2*
d*e*l^3*z - 27*a*b^5*c^3*d*e^2*l^2*z + 27*a*b^4*c^4*d^2*g*j^2*z + 18*a^2*b^
2*c^5*f^3*g*h*z + 144*a^3*b^2*c^4*d*e*k^3*z - 63*a^2*b^4*c^3*d*e*k^3*z + 27
*a*b^4*c^4*d^2*e*k^2*z - 9*a^2*b^3*c^4*e*g*h^3*z - 108*a^2*b*c^6*d^2*g^2*h*
z + 81*a^2*b^3*c^4*d*e*j^3*z + 27*a*b^3*c^5*d^2*g^2*h*z - 27*a*b^2*c^6*d^2*
e^2*j*z - 18*a^2*b^2*c^5*d*g^3*h*z + 108*a^2*b*c^6*d*e^2*h^2*z - 27*a*b^3*c
^5*d*e^2*h^2*z + 27*a*b^2*c^6*d^2*f^2*g*z - 18*a^2*b^2*c^5*d*e*h^3*z - 216*
a^6*c^3*j^2*k*l*m*z + 216*a^6*c^3*h*j*l^2*m*z + 216*a^6*c^3*f*k*l*m^2*z - 2
16*a^5*c^4*f^2*k*l*m*z - 216*a^5*c^4*g^2*j*k*m*z + 216*a^5*c^4*f*j^2*k*l*z
+ 216*a^5*c^4*f*h^2*l*m*z + 216*a^5*c^4*e*j^2*k*m*z + 216*a^5*c^4*d*j^2*l*m
*z + 216*a^5*c^4*g*h*j^2*m*z - 216*a^5*c^4*e*j*k^2*l*z - 216*a^5*c^4*d*j*k^
2*m*z + 216*a^4*c^5*d^2*j*k*m*z - 18*a^6*b^2*c*k*l*m^3*z + 216*a^5*c^4*f*g*
k^2*m*z - 216*a^5*c^4*d*j*k*l^2*z - 72*a^6*b*c^2*j*l^3*m*z + 18*a^5*b^3*c*j
*l^3*m*z - 216*a^5*c^4*f*h*j*l^2*z + 216*a^5*c^4*e*h*k*l^2*z + 216*a^5*c^4*
e*f*l^2*m*z - 216*a^4*c^5*e^2*h*k*l*z + 216*a^4*c^5*e^2*h*j*m*z - 216*a^4*c

^5*e^2*f*l*m*z - 216*a^5*c^4*e*f*k*m^2*z + 216*a^5*c^4*d*g*k*m^2*z - 216*a^
5*c^4*d*f*l*m^2*z + 216*a^4*c^5*e*f^2*k*m*z + 216*a^4*c^5*d*f^2*l*m*z + 108
*a^5*b*c^3*j^3*k*l*z - 216*a^5*c^4*f*g*h*m^2*z + 216*a^4*c^5*f^2*g*h*m*z +
216*a^4*c^5*f*g^2*j*k*z - 216*a^4*c^5*e*g^2*j*l*z + 216*a^4*c^5*d*g^2*j*m*z
 - 72*a^6*b*c^2*h*k*m^3*z - 72*a^6*b*c^2*g*l*m^3*z + 54*a^5*b^3*c*h*k*m^3*z
 + 54*a^5*b^3*c*g*l*m^3*z - 216*a^4*c^5*d*h^2*j*k*z - 18*a^4*b^4*c*f*l^3*m*
z + 9*a^4*b^4*c*h*k*l^3*z - 216*a^4*c^5*e*f*j^2*k*z - 216*a^4*c^5*e*f*h^2*m
*z - 216*a^4*c^5*d*g*j^2*k*z - 216*a^4*c^5*d*f*j^2*l*z - 216*a^4*c^5*d*e*j^
2*m*z - 72*a^5*b*c^3*f*k^3*m*z + 72*a^4*b*c^4*g^3*j*m*z + 36*a^5*b*c^3*g*k^
3*l*z - 36*a^4*b*c^4*g^3*k*l*z - 216*a^4*c^5*f*g*h*j^2*z + 216*a^4*c^5*d*f*
j*k^2*z - 216*a^3*c^6*d^2*f*j*k*z - 216*a^3*c^6*d^2*e*j*l*z + 72*a^4*b^4*c*
f*j*m^3*z - 63*a^4*b^4*c*e*k*m^3*z - 63*a^4*b^4*c*d*l*m^3*z + 216*a^4*c^5*d
*g*h*k^2*z - 216*a^3*c^6*d^2*g*h*k*z + 216*a^3*c^6*d^2*f*g*m*z - 216*a^3*c^
6*d*e^2*j*k*z + 144*a^5*b*c^3*f*j*l^3*z - 144*a^3*b*c^5*e^3*j*m*z - 72*a^5*
b*c^3*e*k*l^3*z + 72*a^3*b*c^5*e^3*k*l*z - 63*a^4*b^4*c*g*h*m^3*z + 18*a^3*
b^5*c*f*j*l^3*z - 18*a*b^5*c^3*e^3*j*m*z - 9*a^3*b^5*c*e*k*l^3*z + 9*a*b^5*
c^3*e^3*k*l*z - 216*a^4*c^5*d*e*h*l^2*z - 216*a^3*c^6*e^2*f*h*j*z + 216*a^3
*c^6*d*e^2*h*l*z - 126*a*b^4*c^4*d^3*j*m*z + 108*a^4*b*c^4*g*h^3*l*z + 63*a
*b^4*c^4*d^3*k*l*z + 36*a^5*b*c^3*g*h*l^3*z - 9*a^3*b^5*c*g*h*l^3*z + 216*a
^4*c^5*d*e*f*m^2*z + 216*a^3*c^6*d*f^2*g*k*z - 216*a^3*c^6*d*e*f^2*m*z + 36
*a^4*b*c^4*e*j^3*k*z + 36*a^4*b*c^4*d*j^3*l*z - 216*a^3*c^6*d*f*g^2*j*z + 7
2*a^3*b^5*c*e*g*m^3*z + 72*a^3*b^5*c*d*h*m^3*z + 72*a^3*b*c^5*f^3*h*k*z + 7
2*a^3*b*c^5*f^3*g*l*z + 36*a^4*b*c^4*g*h*j^3*z + 18*a*b^4*c^4*e^3*f*m*z + 9
*a^2*b^6*c*e*g*l^3*z + 9*a^2*b^6*c*d*h*l^3*z - 9*a*b^4*c^4*e^3*h*k*z - 9*a*
b^4*c^4*e^3*g*l*z + 216*a^3*c^6*d*e*f*j^2*z - 144*a^2*b*c^6*d^3*f*m*z + 108
*a^3*b*c^5*e*g^3*k*z - 108*a^3*b*c^5*d*g^3*l*z + 108*a*b^3*c^5*d^3*f*m*z -
72*a^4*b*c^4*d*h*k^3*z + 72*a^2*b*c^6*d^3*h*k*z - 54*a*b^3*c^5*d^3*h*k*z +
36*a^4*b*c^4*e*g*k^3*z - 36*a^2*b*c^6*d^3*g*l*z - 27*a*b^3*c^5*d^3*g*l*z -
81*a^2*b^6*c*d*e*m^3*z + 216*a^4*b*c^4*d*e*l^3*z + 72*a^2*b*c^6*e^3*f*j*z +
 72*a^2*b*c^6*d*e^3*l*z - 18*a*b^3*c^5*e^3*f*j*z - 18*a*b^3*c^5*d*e^3*l*z -
 90*a*b^2*c^6*d^3*f*j*z + 72*a*b^2*c^6*d^3*e*k*z + 36*a^3*b*c^5*e*g*h^3*z -
 36*a^2*b*c^6*e^3*g*h*z + 9*a*b^6*c^2*d*e*k^3*z + 9*a*b^3*c^5*e^3*g*h*z - 1
80*a^3*b*c^5*d*e*j^3*z + 18*a*b^2*c^6*d^3*g*h*z - 9*a*b^5*c^3*d*e*j^3*z + 1
8*a*b^2*c^6*d*e^3*h*z + 9*a*b^4*c^4*d*e*h^3*z + 36*a^2*b*c^6*d*e*g^3*z - 9*
a*b^3*c^5*d*e*g^3*z - 18*a*b^2*c^6*d*e*f^3*z + 27*a^5*b^2*c^2*h^2*l*m^2*z -
 27*a^5*b^2*c^2*j*k^2*l^2*z + 27*a^4*b^3*c^2*h^2*k^2*m*z + 27*a^4*b^3*c^2*g
^2*l^2*m*z + 27*a^5*b^2*c^2*g*k^2*m^2*z - 27*a^4*b^3*c^2*h^2*k*l^2*z - 27*a
^4*b^3*c^2*g^2*k*m^2*z - 135*a^4*b^2*c^3*e^2*l*m^2*z + 27*a^5*b^2*c^2*e*l^2
*m^2*z + 27*a^4*b^3*c^2*h*j^2*l^2*z - 27*a^4*b^2*c^3*h^2*j^2*l*z + 27*a^3*b
^4*c^2*e^2*l*m^2*z - 270*a^4*b^3*c^2*f*j^2*m^2*z - 270*a^4*b^2*c^3*f^2*j*m^
2*z + 162*a^3*b^4*c^2*f^2*j*m^2*z - 108*a^3*b^3*c^3*f^2*j^2*m*z - 27*a^4*b^
2*c^3*h^2*j*k^2*z - 27*a^4*b^2*c^3*g^2*j*l^2*z + 27*a^3*b^3*c^3*e^2*k^2*m*z
 + 27*a^3*b^3*c^3*d^2*l^2*m*z + 27*a^2*b^5*c^2*f^2*j^2*m*z + 162*a^3*b^3*c^
3*d^2*k*m^2*z - 27*a^4*b^3*c^2*d*k^2*m^2*z - 27*a^4*b^2*c^3*g*j^2*k^2*z + 2
7*a^3*b^3*c^3*g^2*h^2*m*z - 27*a^2*b^5*c^2*d^2*k*m^2*z + 162*a^3*b^2*c^4*d^

2*k^2*l*z - 108*a^4*b^2*c^3*g*h^2*l^2*z - 27*a^4*b^2*c^3*e*j^2*l^2*z + 27*a
^3*b^4*c^2*g*h^2*l^2*z + 27*a^3*b^2*c^4*e^2*j^2*l*z - 27*a^2*b^4*c^3*d^2*k^
2*l*z - 162*a^3*b^3*c^3*f^2*h*l^2*z + 162*a^3*b^3*c^3*e^2*h*m^2*z - 135*a^4
*b^2*c^3*e*h^2*m^2*z + 135*a^3*b^2*c^4*f^2*h^2*l*z + 27*a^3*b^4*c^2*e*h^2*m
^2*z - 27*a^3*b^3*c^3*g^2*h*k^2*z - 27*a^3*b^2*c^4*e^2*j*k^2*z - 27*a^3*b^2
*c^4*d^2*j*l^2*z + 27*a^2*b^5*c^2*f^2*h*l^2*z - 27*a^2*b^5*c^2*e^2*h*m^2*z
- 27*a^2*b^4*c^3*f^2*h^2*l*z - 27*a^3*b^2*c^4*g^2*h^2*j*z + 27*a^2*b^3*c^4*
e^2*g^2*m*z - 27*a^2*b^3*c^4*d^2*j^2*k*z + 27*a^2*b^3*c^4*d^2*h^2*m*z + 351
*a^3*b^2*c^4*d^2*g*m^2*z - 189*a^2*b^4*c^3*d^2*g*m^2*z + 162*a^3*b^3*c^3*d*
g^2*m^2*z - 162*a^3*b^2*c^4*e^2*g*l^2*z + 135*a^3*b^3*c^3*d*h^2*l^2*z + 135
*a^3*b^2*c^4*f^2*g*k^2*z - 27*a^2*b^5*c^2*d*h^2*l^2*z - 27*a^2*b^5*c^2*d*g^
2*m^2*z - 27*a^2*b^4*c^3*f^2*g*k^2*z + 27*a^2*b^4*c^3*e^2*g*l^2*z + 27*a^2*
b^3*c^4*f^2*g^2*k*z + 27*a^2*b^3*c^4*e^2*h^2*k*z + 135*a^3*b^2*c^4*e*f^2*l^
2*z - 108*a^3*b^2*c^4*e*g^2*k^2*z + 108*a^2*b^2*c^5*d^2*g^2*l*z + 27*a^3*b^
2*c^4*e*h^2*j^2*z + 27*a^2*b^4*c^3*e*g^2*k^2*z - 27*a^2*b^4*c^3*e*f^2*l^2*z
- 27*a^2*b^3*c^4*e^2*h*j^2*z - 27*a^2*b^2*c^5*e^2*f^2*l*z - 27*a^2*b^2*c^5
*e^2*g^2*j*z - 27*a^2*b^2*c^5*d^2*h^2*j*z + 162*a^2*b^3*c^4*d*e^2*l^2*z - 1
35*a^2*b^2*c^5*d^2*g*j^2*z - 27*a^2*b^3*c^4*d*g^2*j^2*z + 27*a^2*b^3*c^4*d*
f^2*k^2*z - 162*a^2*b^2*c^5*d^2*e*k^2*z - 27*a^2*b^2*c^5*e*f^2*h^2*z - 72*a
^7*c^2*k*l*m^3*z + 9*a^5*b^4*k*l*m^3*z + 72*a^6*c^3*j*k^3*m*z - 72*a^6*c^3*
h*k*l^3*z - 72*a^6*c^3*f*l^3*m*z - 72*a^5*c^4*h^3*k*l*z - 72*a^5*c^4*h^3*j*
m*z - 9*a^4*b^5*h*k*m^3*z - 9*a^4*b^5*g*l*m^3*z - 144*a^6*c^3*f*j*m^3*z - 1
44*a^5*c^4*h*j^3*k*z - 144*a^5*c^4*g*j^3*l*z - 144*a^5*c^4*f*j^3*m*z - 144*
a^4*c^5*f^3*j*m*z + 72*a^6*c^3*e*k*m^3*z + 72*a^6*c^3*d*l*m^3*z + 72*a^4*c^
5*f^3*k*l*z + 72*a^6*c^3*g*h*m^3*z + 18*b^6*c^3*d^3*j*m*z - 18*a^3*b^6*f*j*
m^3*z - 9*b^6*c^3*d^3*k*l*z + 9*a^3*b^6*e*k*m^3*z + 9*a^3*b^6*d*l*m^3*z + 1
44*a^5*c^4*d*k^3*l*z + 144*a^3*c^6*d^3*k*l*z - 72*a^5*c^4*f*j*k^3*z - 72*a^
3*c^6*d^3*j*m*z + 9*a^3*b^6*g*h*m^3*z - 72*a^5*c^4*g*h*k^3*z - 72*a^4*c^5*g
^3*h*k*z - 72*a^4*c^5*f*g^3*m*z - 108*a^5*b*c^3*j^4*m*z + 63*a^6*b^2*c*j*m^
4*z + 36*a^6*b*c^2*k*l^4*z - 9*a^5*b^3*c*k*l^4*z - 144*a^5*c^4*e*g*l^3*z -
144*a^3*c^6*e^3*g*l*z + 72*a^5*c^4*d*h*l^3*z + 72*a^4*c^5*f*h^3*j*z + 72*a^
4*c^5*e*h^3*k*z + 72*a^4*c^5*d*h^3*l*z + 72*a^3*c^6*e^3*h*k*z + 72*a^3*c^6*
e^3*f*m*z - 18*b^5*c^4*d^3*f*m*z + 9*b^5*c^4*d^3*h*k*z + 9*b^5*c^4*d^3*g*l*
z - 9*a^2*b^7*e*g*m^3*z - 9*a^2*b^7*d*h*m^3*z + 144*a^4*c^5*e*g*j^3*z + 144
*a^4*c^5*d*h*j^3*z - 72*a^5*c^4*d*e*m^3*z - 72*a^3*c^6*e*f^3*k*z - 72*a^3*c
^6*d*f^3*l*z + 144*a^6*b*c^2*f*m^4*z - 108*a^5*b^3*c*f*m^4*z - 72*a^3*c^6*f
^3*g*h*z + 36*a^5*b*c^3*h*k^4*z - 36*a^3*b*c^5*f^4*m*z + 18*b^4*c^5*d^3*f*j
*z - 9*b^4*c^5*d^3*e*k*z + 9*a^4*b^4*c*g*l^4*z - 144*a^4*c^5*d*e*k^3*z - 14
4*a^2*c^7*d^3*e*k*z + 72*a^2*c^7*d^3*f*j*z - 9*b^4*c^5*d^3*g*h*z + 72*a^3*c
^6*d*g^3*h*z + 72*a^2*c^7*d^3*g*h*z - 72*a^5*b*c^3*d*l^4*z - 72*a^4*b*c^4*f
*j^4*z + 45*a*b^2*c^6*d^4*l*z - 36*a^2*b*c^6*e^4*k*z - 9*a^3*b^5*c*d*l^4*z
+ 9*a*b^3*c^5*e^4*k*z - 72*a^3*c^6*d*e*h^3*z - 72*a^2*c^7*d*e^3*h*z + 9*b^3
*c^6*d^3*e*g*z + 72*a^2*c^7*d*e*f^3*z + 36*a^3*b*c^5*d*h^4*z - 9*a*b^2*c^6*
e^4*g*z + 36*a*b*c^7*d^3*f^2*z + 90*a^5*b^2*c^2*j^3*m^2*z + 45*a^5*b^2*c^2*
j^2*l^3*z + 9*a^4*b^3*c^2*j^2*k^3*z - 9*a^4*b^3*c^2*h^3*m^2*z - 45*a^4*b^2*

c^3*g^3*m^2*z + 9*a^3*b^4*c^2*g^3*m^2*z + 198*a^4*b^3*c^2*f^2*m^3*z - 108*a
^3*b^3*c^3*f^3*m^2*z + 18*a^2*b^5*c^2*f^3*m^2*z - 117*a^4*b^2*c^3*f^2*l^3*z
 + 117*a^3*b^2*c^4*e^3*m^2*z + 63*a^3*b^4*c^2*f^2*l^3*z - 63*a^2*b^4*c^3*e^
3*m^2*z - 171*a^2*b^3*c^4*d^3*m^2*z - 54*a^3*b^3*c^3*f^2*k^3*z + 9*a^3*b^2*
c^4*g^3*j^2*z + 9*a^2*b^5*c^2*f^2*k^3*z + 18*a^3*b^2*c^4*f^2*j^3*z + 18*a^2
*b^3*c^4*f^3*j^2*z - 9*a^2*b^4*c^3*f^2*j^3*z - 45*a^2*b^2*c^5*e^3*j^2*z + 9
*a^2*b^3*c^4*f^2*h^3*z - 9*a^2*b^2*c^5*f^2*g^3*z + 9*a*b^8*d*e*m^3*z - 36*a
*b*c^7*d^4*h*z - 108*a^6*c^3*h^2*l*m^2*z + 108*a^6*c^3*j*k^2*l^2*z - 108*a^
6*c^3*g*k^2*m^2*z - 108*a^6*c^3*e*l^2*m^2*z + 108*a^5*c^4*h^2*j^2*l*z + 108
*a^5*c^4*e^2*l*m^2*z + 216*a^5*c^4*f^2*j*m^2*z + 108*a^5*c^4*h^2*j*k^2*z +
108*a^5*c^4*g^2*j*l^2*z + 108*a^5*c^4*g*j^2*k^2*z - 216*a^4*c^5*d^2*k^2*l*z
 + 108*a^5*c^4*e*j^2*l^2*z - 108*a^4*c^5*e^2*j^2*l*z - 9*a^6*b^2*c*l^3*m^2*
z + 108*a^5*c^4*e*h^2*m^2*z - 108*a^4*c^5*f^2*h^2*l*z + 108*a^4*c^5*e^2*j*k
^2*z + 108*a^4*c^5*d^2*j*l^2*z - 144*a^6*b*c^2*j^2*m^3*z + 108*a^4*c^5*g^2*
h^2*j*z - 27*a^4*b^4*c*j^3*m^2*z + 27*a^4*b^3*c^2*j^4*m*z + 9*a^5*b^2*c^2*k
^4*l*z + 216*a^4*c^5*e^2*g*l^2*z - 108*a^4*c^5*f^2*g*k^2*z - 108*a^4*c^5*d^
2*g*m^2*z - 9*a^4*b^4*c*j^2*l^3*z - 108*a^4*c^5*e*h^2*j^2*z - 108*a^4*c^5*e
*f^2*l^2*z + 108*a^3*c^6*e^2*f^2*l*z - 36*a^5*b*c^3*j^2*k^3*z + 36*a^5*b*c^
3*h^3*m^2*z + 108*a^3*c^6*e^2*g^2*j*z + 108*a^3*c^6*d^2*h^2*j*z - 216*a^5*b
*c^3*f^2*m^3*z + 144*a^4*b*c^4*f^3*m^2*z + 108*a^3*c^6*d^2*g*j^2*z - 72*a^3
*b^5*c*f^2*m^3*z - 45*a^5*b^2*c^2*g*l^4*z - 9*a^4*b^3*c^2*h*k^4*z - 9*a^3*b
^2*c^4*g^4*l*z + 9*a^2*b^3*c^4*f^4*m*z + 216*a^3*c^6*d^2*e*k^2*z - 9*a^2*b^
6*c*f^2*l^3*z + 9*a*b^6*c^2*e^3*m^2*z + 108*a^3*c^6*e*f^2*h^2*z + 108*a^3*b
*c^5*d^3*m^2*z + 108*a^2*c^7*d^2*e^2*j*z + 72*a^4*b*c^4*f^2*k^3*z + 72*a*b^
5*c^3*d^3*m^2*z - 72*a^3*b*c^5*f^3*j^2*z + 54*a^4*b^3*c^2*d*l^4*z - 45*a^4*
b^2*c^3*e*k^4*z + 18*a^3*b^3*c^3*f*j^4*z + 9*a^3*b^4*c^2*e*k^4*z - 9*a^2*b^
2*c^5*f^4*j*z - 108*a^2*c^7*d^2*f^2*g*z + 9*a^3*b^2*c^4*g*h^4*z + 9*a*b^4*c
^4*e^3*j^2*z - 72*a^2*b*c^6*d^3*j^2*z + 54*a*b^3*c^5*d^3*j^2*z - 36*a^3*b*c
^5*f^2*h^3*z - 9*a^2*b^3*c^4*d*h^4*z + 9*a^2*b^2*c^5*e*g^4*z + 9*a*b^2*c^6*
e^3*f^2*z + 36*a^7*c^2*l^3*m^2*z + 72*a^6*c^3*j^3*m^2*z - 36*a^6*c^3*j^2*l^
3*z + 9*a^4*b^5*j^2*m^3*z + 36*a^5*c^4*g^3*m^2*z + 36*a^5*c^4*f^2*l^3*z - 3
6*a^4*c^5*e^3*m^2*z - 9*b^7*c^2*d^3*m^2*z + 9*a^2*b^7*f^2*m^3*z - 36*a^4*c^
5*g^3*j^2*z + 72*a^4*c^5*f^2*j^3*z + 36*a^3*c^6*e^3*j^2*z - 9*b^5*c^4*d^3*j
^2*z + 36*a^3*c^6*f^2*g^3*z - 9*a^4*b^2*c^3*j^5*z - 36*a^2*c^7*e^3*f^2*z -
9*b^3*c^6*d^3*f^2*z + 36*a^7*c^2*j*m^4*z - 36*a^6*c^3*k^4*l*z - 18*a^5*b^4*
j*m^4*z + 36*a^6*c^3*g*l^4*z + 36*a^4*c^5*g^4*l*z + 18*a^4*b^5*f*m^4*z - 9*
b^4*c^5*d^4*l*z + 36*a^5*c^4*e*k^4*z + 36*a^3*c^6*f^4*j*z - 36*a^2*c^7*d^4*
l*z - 36*a^4*c^5*g*h^4*z + 9*b^3*c^6*d^4*h*z - 36*a^3*c^6*e*g^4*z + 36*a^2*
c^7*e^4*g*z - 9*b^2*c^7*d^4*e*z - 36*a^7*b*c*m^5*z + 36*a*c^8*d^4*e*z + 9*a
^6*b^3*m^5*z + 36*a^5*c^4*j^5*z + 9*a^4*b^3*c*g*h*j*k*l*m - 9*a^3*b^4*c*e*g
*j*k*l*m - 9*a^3*b^4*c*d*h*j*k*l*m - 9*a^3*b^4*c*f*g*h*k*l*m + 36*a^4*b*c^3
*d*e*j*k*l*m + 9*a^2*b^5*c*d*e*j*k*l*m + 36*a^4*b*c^3*e*f*h*j*l*m + 36*a^4*
b*c^3*e*f*g*k*l*m + 36*a^4*b*c^3*d*f*h*k*l*m + 9*a^2*b^5*c*e*f*g*k*l*m + 9*
a^2*b^5*c*d*f*h*k*l*m + 36*a^3*b*c^4*d*e*f*j*k*l + 9*a*b^5*c^2*d*e*f*j*k*l
+ 36*a^3*b*c^4*d*e*g*h*k*l + 36*a^3*b*c^4*d*e*f*h*k*m + 36*a^3*b*c^4*d*e*f*

```
g*l*m + 9*a*b^5*c^2*d*e*f*h*k*m + 9*a*b^5*c^2*d*e*f*g*l*m - 9*a*b^4*c^3*d*e
*f*h*j*k - 9*a*b^4*c^3*d*e*f*g*j*l - 9*a*b^4*c^3*d*e*f*g*h*m + 9*a*b^3*c^4*
d*e*f*g*h*j - 9*a*b^6*c*d*e*f*k*l*m + 18*a^4*b^2*c^2*e*g*j*k*l*m + 18*a^4*b
^2*c^2*d*h*j*k*l*m + 18*a^4*b^2*c^2*f*g*h*k*l*m - 36*a^3*b^3*c^2*d*e*j*k*l*
m - 36*a^3*b^3*c^2*e*f*g*k*l*m - 36*a^3*b^3*c^2*d*f*h*k*l*m + 9*a^3*b^3*c^2
*f*g*h*j*k*l + 9*a^3*b^3*c^2*e*g*h*j*k*m + 9*a^3*b^3*c^2*d*g*h*j*l*m - 108*
a^3*b^2*c^3*d*e*f*k*l*m + 54*a^2*b^4*c^2*d*e*f*k*l*m - 36*a^3*b^2*c^3*d*f*g
*j*k*m + 18*a^3*b^2*c^3*e*f*g*j*k*l + 18*a^3*b^2*c^3*d*f*h*j*k*l + 18*a^3*b
^2*c^3*d*e*h*j*k*m + 18*a^3*b^2*c^3*d*e*g*j*l*m - 9*a^2*b^4*c^2*e*f*g*j*k*l
 - 9*a^2*b^4*c^2*d*f*h*j*k*l - 9*a^2*b^4*c^2*d*e*h*j*k*m - 9*a^2*b^4*c^2*d*
e*g*j*l*m + 18*a^3*b^2*c^3*e*f*g*h*k*m + 18*a^3*b^2*c^3*d*f*g*h*l*m - 9*a^2
*b^4*c^2*e*f*g*h*k*m - 9*a^2*b^4*c^2*d*f*g*h*l*m - 36*a^2*b^3*c^3*d*e*f*j*k
*l - 36*a^2*b^3*c^3*d*e*f*h*k*m - 36*a^2*b^3*c^3*d*e*f*g*l*m + 9*a^2*b^3*c^
3*e*f*g*h*j*k + 9*a^2*b^3*c^3*d*f*g*h*j*l + 9*a^2*b^3*c^3*d*e*g*h*j*m + 18*
a^2*b^2*c^4*d*e*f*h*j*k + 18*a^2*b^2*c^4*d*e*f*g*j*l + 18*a^2*b^2*c^4*d*e*f
*g*h*m - 9*a^5*b^2*c*h*j*k^2*l*m - 9*a^5*b^2*c*g*j*k*l^2*m + 27*a^5*b^2*c*f
*j*k*l*m^2 - 9*a^4*b^3*c*f*j^2*k*l*m + 9*a^3*b^4*c*f^2*j*k*l*m - 18*a^5*b*c
^2*e*j*k^2*l*m - 9*a^5*b^2*c*g*h*k*l*m^2 + 9*a^4*b^3*c*e*j*k^2*l*m - 18*a^5
*b*c^2*f*h*k^2*l*m - 18*a^5*b*c^2*d*j*k*l^2*m + 9*a^4*b^3*c*f*h*k^2*l*m + 9
*a^4*b^3*c*d*j*k*l^2*m + 36*a^5*b*c^2*e*h*k*l^2*m - 36*a^4*b*c^3*e^2*h*k*l*
m + 18*a^5*b*c^2*f*h*j*l^2*m - 18*a^5*b*c^2*f*g*k*l^2*m - 18*a^4*b^3*c*e*h*
k*l^2*m + 9*a^4*b^3*c*f*g*k*l^2*m + 9*a^3*b^4*c*e*h^2*k*l*m - 9*a^2*b^5*c*e
^2*h*k*l*m - 54*a^5*b*c^2*e*h*j*l*m^2 - 18*a^5*b*c^2*e*g*k*l*m^2 - 18*a^5*b
*c^2*d*h*k*l*m^2 + 18*a^4*b^3*c*e*h*j*l*m^2 - 9*a^4*b^3*c*f*h*j*k*m^2 - 9*a
^4*b^3*c*f*g*j*l*m^2 + 9*a^4*b^3*c*e*g*k*l*m^2 + 9*a^4*b^3*c*d*h*k*l*m^2 +
18*a^4*b*c^3*f*g^2*j*k*m - 18*a^4*b*c^3*e*g^2*j*l*m + 18*a^3*b^4*c*d*g*k^2*
l*m - 9*a^3*b^4*c*e*f*k^2*l*m - 9*a^2*b^5*c*d*g^2*k*l*m - 18*a^4*b*c^3*f*g^
2*h*l*m - 18*a^4*b*c^3*d*h^2*j*k*m - 9*a^3*b^4*c*d*f*k*l^2*m - 54*a^4*b*c^3
*d*g*j^2*k*m - 18*a^4*b*c^3*f*g*h^2*k*m - 18*a^4*b*c^3*e*g*j^2*k*l - 18*a^4
*b*c^3*d*h*j^2*k*l - 18*a^3*b^4*c*d*g*j*k*m^2 + 9*a^3*b^4*c*e*f*j*k*m^2 + 9
*a^3*b^4*c*d*f*j*l*m^2 - 9*a^3*b^4*c*d*e*k*l*m^2 - 54*a^3*b*c^4*d^2*f*j*k*m
 + 36*a^4*b*c^3*d*g*j*k^2*l - 36*a^3*b*c^4*d^2*g*j*k*l - 18*a^4*b*c^3*e*f*j
*k^2*l + 18*a^4*b*c^3*d*f*j*k^2*m - 18*a^3*b*c^4*d^2*e*j*l*m + 9*a^3*b^4*c*
f*g*h*j*m^2 - 9*a*b^5*c^2*d^2*g*j*k*l + 36*a^4*b*c^3*d*g*h*k^2*m - 36*a^3*b
*c^4*d^2*g*h*k*m + 18*a^4*b*c^3*e*g*h*k^2*l - 18*a^4*b*c^3*e*f*h*k^2*m - 18
*a^4*b*c^3*d*f*j*k*l^2 - 18*a^3*b*c^4*d^2*f*h*l*m - 18*a^3*b*c^4*d*e^2*j*k*
m - 9*a*b^5*c^2*d^2*g*h*k*m - 54*a^4*b*c^3*d*g*h*k*l^2 - 54*a^3*b*c^4*e^2*f
*h*j*m - 18*a^4*b*c^3*d*f*g*l^2*m - 18*a^3*b*c^4*e^2*f*g*k*m - 54*a^4*b*c^3
*d*f*g*k*m^2 - 36*a^4*b*c^3*e*f*g*j*m^2 - 36*a^4*b*c^3*d*f*h*j*m^2 + 36*a^3
*b*c^4*e*f^2*g*j*m + 36*a^3*b*c^4*d*f^2*h*j*m - 18*a^4*b*c^3*d*e*h*k*m^2 -
18*a^4*b*c^3*d*e*g*l*m^2 + 18*a^3*b*c^4*e*f^2*h*j*l - 18*a^3*b*c^4*e*f^2*g*
k*l - 18*a^3*b*c^4*d*f^2*h*k*l + 18*a^3*b*c^4*d*f^2*g*k*m - 9*a^2*b^5*c*e*f
*g*j*m^2 - 9*a^2*b^5*c*d*f*h*j*m^2 - 54*a^3*b*c^4*d*f*g^2*j*m - 18*a^3*b*c^
4*e*f*g^2*j*l - 18*a*b^4*c^3*d^2*f*g*j*m + 9*a*b^4*c^3*d^2*g*h*j*k + 9*a*b^
4*c^3*d^2*f*g*k*l + 9*a*b^4*c^3*d^2*e*g*k*m - 9*a*b^4*c^3*d^2*e*f*l*m - 18*
```

a^3*b*c^4*e*f*g^2*h*m - 18*a^3*b*c^4*d*f*h^2*j*k - 9*a*b^4*c^3*d*e^2*f*k*m
+ 18*a^3*b*c^4*d*f*g*j^2*k - 18*a^3*b*c^4*d*f*g*h^2*m - 18*a^3*b*c^4*d*e*h*
j^2*k - 18*a^3*b*c^4*d*e*g*j^2*l + 18*a*b^4*c^3*d*e*f^2*j*m - 9*a*b^5*c^2*d
*e*f*j^2*m - 9*a*b^4*c^3*d*e*f^2*k*l - 18*a^2*b*c^5*d^2*e*f*j*l - 9*a*b^3*c
^4*d^2*e*g*j*k + 9*a*b^3*c^4*d^2*e*f*j*l - 54*a^2*b*c^5*d^2*e*g*h*l - 18*a^
2*b*c^5*d^2*e*f*h*m - 18*a^2*b*c^5*d*e^2*f*j*k + 18*a*b^3*c^4*d^2*e*g*h*l -
 9*a*b^3*c^4*d^2*f*g*h*k + 9*a*b^3*c^4*d^2*e*f*h*m + 9*a*b^3*c^4*d*e^2*f*j*
k - 36*a^3*b*c^4*d*e*f*h*l^2 + 36*a^2*b*c^5*d*e^2*f*h*l + 18*a^2*b*c^5*d*e^
2*g*h*k - 18*a^2*b*c^5*d*e^2*f*g*m - 18*a*b^3*c^4*d*e^2*f*h*l - 9*a*b^5*c^2
*d*e*f*h*l^2 + 9*a*b^4*c^3*d*e*f*h^2*l + 9*a*b^3*c^4*d*e^2*f*g*m - 18*a^2*b
*c^5*d*e*f^2*h*k - 18*a^2*b*c^5*d*e*f^2*g*l + 9*a*b^3*c^4*d*e*f^2*h*k + 9*a
*b^3*c^4*d*e*f^2*g*l + 27*a*b^2*c^5*d^2*e*f*g*k + 9*a*b^4*c^3*d*e*f*g*k^2 -
 9*a*b^3*c^4*d*e*f*g^2*k - 9*a*b^2*c^5*d^2*e*f*h*j - 9*a*b^2*c^5*d*e^2*f*g*
j - 9*a*b^2*c^5*d*e*f^2*g*h + 72*a^4*c^4*d*f*g*j*k*m + 72*a^4*c^4*d*e*f*k*l
*m + 9*a*b^6*c*d^2*g*k*l*m + 9*a*b^6*c*d*e*f*j*m^2 - 27*a^4*b^2*c^2*f^2*j*k
*l*m - 9*a^4*b^2*c^2*g^2*h*j*l*m + 36*a^3*b^3*c^2*e^2*h*k*l*m - 18*a^4*b^2*
c^2*e*h^2*k*l*m - 9*a^4*b^2*c^2*g*h^2*j*k*m + 18*a^4*b^2*c^2*f*h*j^2*k*m +
18*a^4*b^2*c^2*f*g*j^2*l*m - 18*a^4*b^2*c^2*e*h*j^2*l*m - 9*a^4*b^2*c^2*g*h
*j^2*k*l - 9*a^3*b^3*c^2*f^2*h*j*k*m - 9*a^3*b^3*c^2*f^2*g*j*l*m - 63*a^4*b
^2*c^2*d*g*k^2*l*m + 63*a^3*b^2*c^3*d^2*g*k*l*m - 45*a^2*b^4*c^2*d^2*g*k*l*
m + 36*a^4*b^2*c^2*e*f*k^2*l*m + 27*a^3*b^3*c^2*d*g^2*k*l*m - 9*a^4*b^2*c^2
*f*h*j*k^2*l - 9*a^4*b^2*c^2*e*h*j*k^2*m + 9*a^3*b^3*c^2*e*g^2*j*l*m - 9*a^
3*b^2*c^3*d^2*h*j*l*m + 36*a^4*b^2*c^2*d*f*k*l^2*m + 27*a^4*b^2*c^2*e*h*j*k
*l^2 - 27*a^3*b^2*c^3*e^2*h*j*k*l - 18*a^3*b^2*c^3*e^2*f*j*l*m - 9*a^4*b^2*
c^2*f*g*j*k*l^2 - 9*a^4*b^2*c^2*d*g*j*l^2*m + 9*a^3*b^3*c^2*f*g^2*h*l*m - 9
*a^3*b^3*c^2*e*h^2*j*k*l + 9*a^3*b^3*c^2*d*h^2*j*k*m - 9*a^3*b^2*c^3*e^2*g*
j*k*m + 9*a^2*b^4*c^2*e^2*h*j*k*l + 72*a^4*b^2*c^2*d*g*j*k*m^2 + 36*a^4*b^2
*c^2*d*e*k*l*m^2 + 27*a^4*b^2*c^2*e*g*h*l^2*m - 27*a^4*b^2*c^2*e*f*j*k*m^2
- 27*a^4*b^2*c^2*d*f*j*l*m^2 - 27*a^3*b^2*c^3*e^2*g*h*l*m + 27*a^3*b^2*c^3*
e*f^2*j*k*m + 27*a^3*b^2*c^3*d*f^2*j*l*m + 18*a^3*b^3*c^2*d*g*j^2*k*m + 9*a
^3*b^3*c^2*f*g*h^2*k*m + 9*a^3*b^3*c^2*e*g*j^2*k*l - 9*a^3*b^3*c^2*e*g*h^2*
l*m - 9*a^3*b^3*c^2*e*f*j^2*k*m + 9*a^3*b^3*c^2*d*h*j^2*k*l - 9*a^3*b^3*c^2
*d*f*j^2*l*m + 9*a^2*b^4*c^2*e^2*g*h*l*m + 36*a^2*b^3*c^3*d^2*g*j*k*l - 27*
a^4*b^2*c^2*f*g*h*j*m^2 + 27*a^3*b^2*c^3*f^2*g*h*j*m - 18*a^4*b^2*c^2*e*f*h
*l*m^2 - 18*a^3*b^3*c^2*d*g*j*k^2*l - 18*a^3*b^2*c^3*d*g^2*j*k*l + 18*a^2*b
^3*c^3*d^2*f*j*k*m - 9*a^4*b^2*c^2*e*g*h*k*m^2 - 9*a^4*b^2*c^2*d*g*h*l*m^2
- 9*a^3*b^3*c^2*f*g*h*j^2*m + 9*a^3*b^3*c^2*e*f*j*k^2*l - 9*a^3*b^2*c^3*f^2
*g*h*k*l + 9*a^2*b^4*c^2*d*g^2*j*k*l + 9*a^2*b^3*c^3*d^2*e*j*l*m + 36*a^3*b
^2*c^3*e*f*g^2*l*m + 36*a^2*b^3*c^3*d^2*g*h*k*m - 18*a^3*b^3*c^2*d*g*h*k^2*
m - 18*a^3*b^2*c^3*d*g^2*h*k*m + 9*a^3*b^3*c^2*e*f*h*k^2*m + 9*a^3*b^3*c^2*
d*f*j*k*l^2 - 9*a^3*b^2*c^3*f*g^2*h*j*l - 9*a^3*b^2*c^3*e*g^2*h*j*m - 9*a^2
*b^4*c^2*e*f*g^2*l*m + 9*a^2*b^4*c^2*d*g^2*h*k*m + 9*a^2*b^3*c^3*d^2*f*h*l*
m + 9*a^2*b^3*c^3*d*e^2*j*k*m + 36*a^3*b^2*c^3*d*f*h^2*k*m + 36*a^3*b^2*c^3
*d*e*j^2*k*l + 18*a^3*b^3*c^2*d*g*h*k*l^2 + 18*a^3*b^2*c^3*e*g*h^2*j*l + 18
*a^3*b^2*c^3*e*f*h^2*k*l - 18*a^3*b^2*c^3*e*f*h^2*j*m - 18*a^3*b^2*c^3*d*g*

h^2*k*l + 18*a^3*b^2*c^3*d*e*h^2*l*m + 18*a^2*b^3*c^3*e^2*f*h*j*m - 9*a^3*b
^3*c^2*e*g*h*j*l^2 - 9*a^3*b^3*c^2*e*f*h*k*l^2 + 9*a^3*b^3*c^2*d*f*g*l^2*m
- 9*a^3*b^3*c^2*d*e*h*l^2*m - 9*a^3*b^2*c^3*f*g*h^2*j*k - 9*a^3*b^2*c^3*d*g
*h^2*j*m - 9*a^2*b^4*c^2*d*f*h^2*k*m - 9*a^2*b^4*c^2*d*e*j^2*k*l - 9*a^2*b^
3*c^3*e^2*g*h*j*l - 9*a^2*b^3*c^3*e^2*f*h*k*l + 9*a^2*b^3*c^3*e^2*f*g*k*m -
9*a^2*b^3*c^3*d*e^2*h*l*m + 36*a^3*b^3*c^2*e*f*g*j*m^2 + 36*a^3*b^3*c^2*d*
f*h*j*m^2 + 18*a^3*b^3*c^2*d*f*g*k*m^2 - 18*a^3*b^2*c^3*e*f*g*j^2*m - 18*a^
3*b^2*c^3*d*f*h*j^2*m - 18*a^2*b^3*c^3*e*f^2*g*j*m - 18*a^2*b^3*c^3*d*f^2*h
*j*m + 9*a^3*b^3*c^2*d*e*h*k*m^2 + 9*a^3*b^3*c^2*d*e*g*l*m^2 - 9*a^3*b^2*c^
3*e*g*h*j^2*k - 9*a^3*b^2*c^3*d*g*h*j^2*l + 9*a^2*b^4*c^2*e*f*g*j^2*m + 9*a
^2*b^4*c^2*d*f*h*j^2*m + 9*a^2*b^3*c^3*e*f^2*g*k*l + 9*a^2*b^3*c^3*d*f^2*h*
k*l + 72*a^2*b^2*c^4*d^2*f*g*j*m + 36*a^2*b^2*c^4*d^2*e*f*l*m + 27*a^3*b^2*
c^3*d*g*h*j*k^2 + 27*a^3*b^2*c^3*d*f*g*k^2*l + 27*a^3*b^2*c^3*d*e*g*k^2*m -
27*a^2*b^2*c^4*d^2*g*h*j*k - 27*a^2*b^2*c^4*d^2*f*g*k*l - 27*a^2*b^2*c^4*d
^2*e*g*k*m + 18*a^2*b^3*c^3*d*f*g^2*j*m - 18*a^2*b^2*c^4*d^2*e*h*k*l - 9*a^
3*b^2*c^3*e*f*h*j*k^2 + 9*a^2*b^3*c^3*e*f*g^2*j*l - 9*a^2*b^3*c^3*d*g^2*h*j
*k - 9*a^2*b^3*c^3*d*f*g^2*k*l - 9*a^2*b^3*c^3*d*e*g^2*k*m - 9*a^2*b^2*c^4*
d^2*f*h*j*l - 9*a^2*b^2*c^4*d^2*e*h*j*m + 36*a^2*b^2*c^4*d*e^2*f*k*m - 27*a
^3*b^2*c^3*d*e*h*j*l^2 + 27*a^2*b^2*c^4*d*e^2*h*j*l - 18*a^3*b^2*c^3*d*e*g*
k*l^2 - 9*a^3*b^2*c^3*d*f*g*j*l^2 + 9*a^2*b^4*c^2*d*e*h*j*l^2 + 9*a^2*b^3*c
^3*e*f*g^2*h*m + 9*a^2*b^3*c^3*d*f*h^2*j*k - 9*a^2*b^3*c^3*d*e*h^2*j*l - 9*
a^2*b^2*c^4*e^2*f*g*j*k - 9*a^2*b^2*c^4*d*e^2*g*j*m + 63*a^3*b^2*c^3*d*e*f*
j*m^2 - 63*a^2*b^2*c^4*d*e*f^2*j*m - 45*a^2*b^4*c^2*d*e*f*j*m^2 + 36*a^2*b^
2*c^4*d*e*f^2*k*l - 27*a^3*b^2*c^3*e*f*g*h*l^2 + 27*a^2*b^3*c^3*d*e*f*j^2*m
+ 27*a^2*b^2*c^4*e^2*f*g*h*l + 9*a^2*b^4*c^2*e*f*g*h*l^2 - 9*a^2*b^3*c^3*e
*f*g*h^2*l + 9*a^2*b^3*c^3*d*f*g*h^2*m + 9*a^2*b^3*c^3*d*e*h*j^2*k + 9*a^2*
b^3*c^3*d*e*g*j^2*l + 18*a^2*b^2*c^4*d*e*g^2*j*k - 9*a^3*b^2*c^3*d*e*g*h*m^
2 - 9*a^2*b^3*c^3*d*e*g*j*k^2 - 9*a^2*b^2*c^4*e*f^2*g*h*k - 9*a^2*b^2*c^4*d
*f^2*g*h*l + 18*a^2*b^2*c^4*d*f*g^2*h*k - 18*a^2*b^2*c^4*d*e*g^2*h*l - 9*a^
2*b^3*c^3*d*f*g*h*k^2 - 9*a^2*b^2*c^4*e*f*g^2*h*j + 36*a^2*b^3*c^3*d*e*f*h*
l^2 - 18*a^2*b^2*c^4*d*e*f*h^2*l - 9*a^2*b^2*c^4*d*f*g*h^2*j - 9*a^2*b^2*c^
4*d*e*g*h*j^2 - 27*a^2*b^2*c^4*d*e*f*g*k^2 + 18*a^2*b^2*c^4*d^2*f*h*k^2 - 9
*a^2*b^3*c^3*e*f*g^2*k^2 - 9*a^2*b^2*c^4*e^2*f*h*j^2 - 9*a^2*b^2*c^4*d*f^2*
h^2*k + 45*a^2*b^3*c^3*d*e*f^2*m^2 + 36*a^2*b^2*c^4*d^2*e*g*l^2 + 9*a^2*b^3
*c^3*d*e*g^2*l^2 + 9*a^2*b^2*c^4*e*f^2*g*j^2 + 9*a^2*b^2*c^4*d*f^2*h*j^2 -
9*a^2*b^2*c^4*d*e^2*h*k^2 - 36*a^2*b^2*c^4*d*e^2*f*l^2 - 9*a^2*b^2*c^4*d*f*
g^2*j^2 - 12*a^6*b*c*h*k*l^3*m + 3*a*b^6*c*e^3*k*l*m + 3*a*b^6*c*d*e*f*l^3
- 12*a*b*c^6*d*e^3*f*h + 9*a^5*b^2*c*h^2*k*l^2*m + 18*a^5*b*c^2*g^2*k^2*l*m
- 9*a^5*b^2*c*h^2*j*l*m^2 + 9*a^5*b*c^2*h^2*j^2*l*m - 9*a^4*b^3*c*g^2*k^2*
l*m - 3*a^4*b^2*c^2*g^3*k*l*m + 18*a^5*b*c^2*f^2*k*l*m^2 + 15*a^3*b^3*c^2*f
^3*k*l*m + 9*a^5*b^2*c*h*j^2*k*m^2 + 9*a^5*b^2*c*g*j^2*l*m^2 - 9*a^5*b^2*c*
f*k^2*l^2*m + 9*a^5*b*c^2*h^2*j*k^2*m + 9*a^5*b*c^2*g^2*j*l^2*m - 9*a^4*b^3
*c*f^2*k*l*m^2 + 36*a^3*b^2*c^3*e^3*k*l*m - 27*a^5*b*c^2*g^2*j*k*m^2 - 18*a
^5*b*c^2*h^2*j*k*l^2 - 18*a^2*b^4*c^2*e^3*k*l*m - 9*a^5*b^2*c*g*j*k^2*m^2 -
9*a^5*b^2*c*e*k^2*l*m^2 + 9*a^5*b*c^2*h*j^2*k^2*l + 9*a^5*b*c^2*g*j^2*k^2*

m + 9*a^4*b^3*c*g^2*j*k*m^2 + 9*a^3*b^4*c*e^2*k*l^2*m + 3*a^4*b^2*c^2*h^3*j
*k*l - 54*a^4*b*c^3*d^2*k^2*l*m - 51*a^2*b^3*c^3*d^3*k*l*m - 27*a^4*b*c^3*e
^2*j^2*l*m - 18*a^5*b*c^2*g*h^2*l^2*m - 9*a^5*b^2*c*e*j*l^2*m^2 - 9*a^5*b^2
*c*d*k*l^2*m^2 + 9*a^5*b*c^2*g^2*h*l*m^2 + 9*a^5*b*c^2*g*j^2*k*l^2 + 9*a^5*
b*c^2*e*j^2*l^2*m - 9*a^3*b^4*c*e^2*j*l*m^2 - 9*a^2*b^5*c*d^2*k^2*l*m + 3*a
^4*b^2*c^2*g*h^3*l*m - 3*a^3*b^3*c^2*g^3*j*k*l + 18*a^5*b*c^2*e*j^2*k*m^2 +
 18*a^5*b*c^2*d*j^2*l*m^2 + 18*a^4*b*c^3*f^2*j^2*k*l + 9*a^5*b*c^2*g*h^2*k*
m^2 + 9*a^5*b*c^2*f*h^2*l*m^2 + 9*a^5*b*c^2*f*j*k^2*l^2 - 9*a^4*b^3*c*e*j^2
*k*m^2 - 9*a^4*b^3*c*d*j^2*l*m^2 + 9*a^4*b^2*c^2*f*j^3*k*l + 9*a^4*b^2*c^2*
e*j^3*k*m + 9*a^4*b^2*c^2*d*j^3*l*m + 9*a^4*b*c^3*f^2*h^2*l*m + 9*a^4*b*c^3
*e^2*j*k^2*m + 9*a^4*b*c^3*d^2*j*l^2*m - 3*a^3*b^3*c^2*g^3*h*k*m - 3*a^3*b^
2*c^3*f^3*j*k*l + 3*a^2*b^4*c^2*f^3*j*k*l + 45*a^4*b*c^3*d^2*j*k*m^2 - 27*a
^5*b*c^2*d*j*k^2*m^2 + 18*a^5*b*c^2*g*h*j^2*m^2 + 18*a^4*b*c^3*e^2*j*k*l^2
+ 15*a^2*b^3*c^3*e^3*j*k*l - 12*a^3*b^2*c^3*f^3*h*k*m - 12*a^3*b^2*c^3*f^3*
g*l*m + 9*a^5*b*c^2*g*h*k^2*l^2 - 9*a^4*b^3*c*g*h*j^2*m^2 + 9*a^4*b^3*c*d*j
*k^2*m^2 + 9*a^4*b^2*c^2*g*h*j^3*m + 9*a^4*b*c^3*g^2*h^2*k*l + 9*a^4*b*c^3*
g^2*h^2*j*m + 9*a^2*b^5*c*d^2*j*k*m^2 + 3*a^2*b^4*c^2*f^3*h*k*m + 3*a^2*b^4
*c^2*f^3*g*l*m + 36*a^2*b^2*c^4*d^3*j*k*l + 18*a^4*b*c^3*e^2*g*l^2*m + 15*a
^2*b^3*c^3*e^3*g*l*m + 12*a^4*b^2*c^2*d*j*k^3*l + 9*a^5*b*c^2*f*g*k^2*m^2 +
 9*a^5*b*c^2*e*h*k^2*m^2 + 9*a^4*b*c^3*g^2*h*j^2*l + 9*a^4*b*c^3*f^2*h*k^2*
l + 9*a^4*b*c^3*f^2*g*k^2*m + 9*a^4*b*c^3*d^2*h*l*m^2 - 9*a^3*b^3*c^2*e*h^3
*k*m + 6*a^2*b^3*c^3*e^3*h*k*m + 45*a^4*b*c^3*e^2*h*j*m^2 + 36*a^2*b^2*c^4*
d^3*h*k*m - 33*a^3*b^2*c^3*d*g^3*l*m - 27*a^4*b*c^3*f^2*h*j*l^2 - 27*a^4*b*
c^3*e^2*f*l*m^2 - 27*a^4*b*c^3*e*h^2*j^2*m - 18*a^4*b*c^3*g^2*h*j*k^2 - 18*
a^4*b*c^3*f*g^2*k^2*l - 18*a^4*b*c^3*e*g^2*k^2*m - 18*a^3*b*c^4*d^2*g^2*l*m
 + 12*a^4*b^2*c^2*d*h*k^3*m + 9*a^5*b*c^2*e*f*l^2*m^2 + 9*a^5*b*c^2*d*g*l^2
*m^2 + 9*a^4*b*c^3*f^2*g*k*l^2 + 9*a^4*b*c^3*e^2*g*k*m^2 + 9*a^4*b*c^3*g*h^
2*j^2*k + 9*a^4*b*c^3*f*h^2*j^2*l + 9*a^4*b*c^3*e*f^2*l^2*m - 9*a^3*b^4*c*e
*h^2*j*m^2 + 9*a^3*b*c^4*e^2*f^2*l*m + 9*a^2*b^5*c*e^2*h*j*m^2 + 9*a^2*b^4*
c^2*d*g^3*l*m - 9*a^2*b^2*c^4*d^3*g*l*m - 9*a*b^5*c^2*d^2*g^2*l*m - 6*a^4*b
^2*c^2*e*h*k^3*l - 6*a^3*b^2*c^3*f*g^3*j*m + 3*a^4*b^2*c^2*g*h*j*k^3 + 3*a^
4*b^2*c^2*f*g*k^3*l + 3*a^4*b^2*c^2*e*g*k^3*m + 3*a^3*b^2*c^3*g^3*h*j*k + 3
*a^3*b^2*c^3*f*g^3*k*l + 3*a^3*b^2*c^3*e*g^3*k*m - 27*a^3*b*c^4*d^2*h^2*k*l
 + 18*a^4*b*c^3*e*f^2*k*m^2 + 18*a^4*b*c^3*d*f^2*l*m^2 + 9*a^4*b*c^3*f*h^2*
j*k^2 + 9*a^4*b*c^3*f*g^2*j*l^2 + 9*a^4*b*c^3*e*g^2*k*l^2 + 9*a^4*b*c^3*d*h
^2*k^2*l + 9*a^3*b^4*c*e*g*j^2*m^2 + 9*a^3*b^4*c*d*h*j^2*m^2 - 9*a^3*b^3*c^
2*e*g*j^3*m - 9*a^3*b^3*c^2*d*h*j^3*m + 9*a^3*b*c^4*e^2*g^2*k*l + 9*a^3*b*c
^4*e^2*g^2*j*m + 9*a^3*b*c^4*d^2*h^2*j*m - 3*a^2*b^3*c^3*f^3*h*j*k - 3*a^2*
b^3*c^3*f^3*g*j*l - 3*a^2*b^3*c^3*e*f^3*k*m - 3*a^2*b^3*c^3*d*f^3*l*m + 45*
a^4*b*c^3*d*g^2*j*m^2 + 45*a^3*b*c^4*d^2*g*j^2*m + 24*a^4*b^2*c^2*d*g*k*l^3
 + 24*a^2*b^2*c^4*e^3*f*j*m + 18*a^4*b*c^3*f^2*g*h*m^2 + 18*a^4*b*c^3*d*h^2
*j*l^2 + 18*a^3*b*c^4*e^2*h^2*j*k - 12*a^4*b^2*c^2*e*g*j*l^3 - 12*a^4*b^2*c
^2*e*f*k*l^3 - 12*a^4*b^2*c^2*d*e*l^3*m - 12*a^2*b^2*c^4*e^3*g*j*l - 12*a^2
*b^2*c^4*e^3*f*k*l - 12*a^2*b^2*c^4*d*e^3*l*m + 9*a^4*b*c^3*f*g*j^2*k^2 + 9
*a^4*b*c^3*e*h*j^2*k^2 + 9*a^3*b^2*c^3*e*h^3*j*k + 9*a^3*b^2*c^3*d*h^3*j*l

+ 9*a^3*b*c^4*f^2*g^2*j*k + 9*a^3*b*c^4*d^2*h*j^2*l + 9*a^2*b^5*c*d*g^2*j*m
^2 + 9*a*b^5*c^2*d^2*g*j^2*m - 3*a^4*b^2*c^2*d*h*j*l^3 - 3*a^2*b^3*c^3*f^3*
g*h*m - 3*a^2*b^2*c^4*e^3*h*j*k + 18*a^4*b*c^3*f*g*h^2*l^2 + 18*a^3*b*c^4*e
^2*g*h^2*m + 18*a^3*b*c^4*d^2*h*j*k^2 + 18*a^3*b*c^4*d^2*f*k^2*l + 18*a^3*b
*c^4*d^2*e*k^2*m + 9*a^4*b*c^3*e*g^2*h*m^2 + 9*a^4*b*c^3*e*f*j^2*l^2 + 9*a^
4*b*c^3*d*g*j^2*l^2 + 9*a^3*b^2*c^3*f*g*h^3*l + 9*a^3*b^2*c^3*e*g*h^3*m + 9
*a^3*b*c^4*f^2*g^2*h*l + 9*a^3*b*c^4*e^2*g*j^2*k + 9*a^3*b*c^4*e^2*f*j^2*l
- 9*a^2*b^3*c^3*d*g^3*j*l + 9*a*b^4*c^3*d^2*g^2*j*l - 3*a^4*b^2*c^2*f*g*h*l
^3 - 3*a^3*b^3*c^2*e*g*j*k^3 - 3*a^3*b^3*c^2*d*h*j*k^3 - 3*a^3*b^3*c^2*d*f*
k^3*l - 3*a^3*b^3*c^2*d*e*k^3*m - 3*a^2*b^2*c^4*e^3*g*h*m - 33*a^3*b^2*c^3*
d*e*j^3*m - 27*a^4*b*c^3*e*f*h^2*m^2 - 27*a^3*b*c^4*d^2*e*k*l^2 - 18*a^4*b*
c^3*d*e*j^2*m^2 - 18*a^3*b*c^4*e*f^2*j^2*k - 18*a^3*b*c^4*d*f^2*j^2*l - 9*a
^4*b^2*c^2*d*e*j*m^3 + 9*a^4*b*c^3*d*g*h^2*m^2 + 9*a^4*b*c^3*d*e*k^2*l^2 +
9*a^3*b*c^4*f^2*g*h^2*k + 9*a^3*b*c^4*e^2*f*j*k^2 + 9*a^3*b*c^4*d^2*f*j*l^2
 + 9*a^3*b*c^4*e*f^2*h^2*m + 9*a^3*b*c^4*d*e^2*k^2*l - 9*a^2*b^5*c*d*e*j^2*
m^2 + 9*a^2*b^4*c^2*d*e*j^3*m - 9*a^2*b^3*c^3*d*g^3*h*m + 9*a^2*b*c^5*d^2*e
^2*k*l + 9*a^2*b*c^5*d^2*e^2*j*m + 9*a*b^4*c^3*d^2*g^2*h*m - 6*a^3*b^2*c^3*
d*g*j^3*k - 3*a^3*b^3*c^2*f*g*h*k^3 + 3*a^3*b^2*c^3*e*f*j^3*k + 3*a^3*b^2*c
^3*d*f*j^3*l + 3*a^2*b^2*c^4*e*f^3*j*k + 3*a^2*b^2*c^4*d*f^3*j*l + 45*a^3*b
*c^4*d^2*g*h*l^2 + 36*a^4*b^2*c^2*e*f*g*m^3 + 36*a^4*b^2*c^2*d*f*h*m^3 - 27
*a^3*b*c^4*e^2*g*h*k^2 - 27*a^3*b*c^4*d*g^2*h^2*l - 18*a^3*b*c^4*f^2*g*h*j^
2 + 18*a^3*b*c^4*d*e^2*j*l^2 + 15*a^3*b^3*c^2*d*e*j*l^3 + 12*a^2*b^2*c^4*e*
f^3*g*m + 12*a^2*b^2*c^4*d*f^3*h*m + 9*a^3*b*c^4*f*g^2*h^2*j + 9*a^3*b*c^4*
e*g^2*h^2*k + 9*a^3*b*c^4*d*f^2*j*k^2 + 9*a^2*b*c^5*d^2*f^2*j*k + 9*a*b^5*c
^2*d^2*g*h*l^2 - 9*a*b^4*c^3*d^2*g*h^2*l - 6*a^2*b^2*c^4*e*f^3*h*l + 3*a^3*
b^2*c^3*f*g*h*j^3 + 3*a^2*b^2*c^4*f^3*g*h*j + 45*a^3*b*c^4*d^2*f*g*m^2 - 27
*a^2*b*c^5*d^2*f^2*g*m + 18*a^3*b*c^4*e^2*f*g*l^2 + 15*a^3*b^3*c^2*e*f*g*l^
3 - 12*a^3*b^2*c^3*d*e*j*k^3 + 9*a^3*b*c^4*d^2*e*h*m^2 + 9*a^3*b*c^4*e*g^2*
h*j^2 + 9*a^3*b*c^4*e*f^2*h*k^2 - 9*a^2*b^3*c^3*d*f*h^3*l + 9*a^2*b*c^5*d^2
*f^2*h*l + 9*a*b^5*c^2*d^2*f*g*m^2 + 9*a*b^3*c^4*d^2*f^2*g*m + 6*a^3*b^3*c^
2*d*f*h*l^3 + 3*a^2*b^4*c^2*d*e*j*k^3 + 18*a^3*b*c^4*e*f*g^2*k^2 + 18*a^2*b
*c^5*d^2*g^2*h*j + 18*a^2*b*c^5*d^2*f*g^2*l + 18*a^2*b*c^5*d^2*e*g^2*m - 12
*a^3*b^2*c^3*d*f*h*k^3 + 9*a^3*b*c^4*e*f*h^2*j^2 + 9*a^3*b*c^4*d*f^2*g*l^2
+ 9*a^3*b*c^4*d*e^2*g*m^2 + 9*a^3*b*c^4*d*g*h^2*j^2 + 9*a^2*b^2*c^4*e*f*g^3
*k + 9*a^2*b^2*c^4*d*g^3*h*j + 9*a^2*b^2*c^4*d*f*g^3*l + 9*a^2*b^2*c^4*d*e*
g^3*m + 9*a^2*b*c^5*e^2*f^2*h*j + 9*a^2*b*c^5*e^2*f^2*g*k - 9*a*b^3*c^4*d^2
*g^2*h*j - 9*a*b^3*c^4*d^2*f*g^2*l - 9*a*b^3*c^4*d^2*e*g^2*m - 3*a^3*b^2*c^
3*e*f*g*k^3 + 3*a^2*b^4*c^2*e*f*g*k^3 + 3*a^2*b^4*c^2*d*f*h*k^3 - 54*a^3*b*
c^4*d*e*f^2*m^2 - 51*a^3*b^3*c^2*d*e*f*m^3 - 27*a^3*b*c^4*d*e*g^2*l^2 + 9*a
^3*b*c^4*d*e*h^2*k^2 + 9*a^2*b*c^5*e^2*f*g^2*j + 9*a^2*b*c^5*d^2*f*h^2*j +
9*a^2*b*c^5*d^2*e*h^2*k + 9*a^2*b*c^5*d*e^2*g^2*l - 9*a*b^5*c^2*d*e*f^2*m^2
 - 9*a*b^4*c^3*d^2*e*g*l^2 - 9*a*b^2*c^5*d^2*e^2*g*l - 9*a*b^2*c^5*d^2*e^2*
f*m - 3*a^2*b^3*c^3*e*f*g*j^3 - 3*a^2*b^3*c^3*d*f*h*j^3 + 36*a^3*b^2*c^3*d*
e*f*l^3 - 27*a^2*b*c^5*d^2*f*g*j^2 - 18*a^2*b^4*c^2*d*e*f*l^3 - 18*a^2*b*c^
5*d*e^2*h^2*j + 9*a^2*b*c^5*d^2*e*h*j^2 + 9*a^2*b*c^5*d*f^2*g^2*j + 9*a*b^4

```
*c^3*d*e^2*f*l^2 + 9*a*b^3*c^4*d^2*f*g*j^2 - 9*a*b^2*c^5*d^2*f^2*g*j - 9*a*
b^2*c^5*d^2*e*f^2*l + 3*a^2*b^2*c^4*d*e*h^3*j - 18*a^2*b*c^5*e^2*f*g*h^2 +
18*a^2*b*c^5*d^2*e*f*k^2 + 15*a^2*b^3*c^3*d*e*f*k^3 + 9*a^2*b*c^5*e*f^2*g^2
*h + 9*a^2*b*c^5*d*e^2*g*j^2 - 9*a*b^3*c^4*d^2*e*f*k^2 + 9*a*b^2*c^5*d^2*e*
g^2*j - 9*a*b^2*c^5*d*e^2*f^2*k + 3*a^2*b^2*c^4*e*f*g*h^3 + 18*a^2*b*c^5*d*
e*f^2*j^2 + 9*a^2*b*c^5*d*f^2*g*h^2 - 9*a*b^3*c^4*d*e*f^2*j^2 + 9*a*b^2*c^5
*d^2*f*g^2*h - 3*a^2*b^2*c^4*d*e*f*j^3 + 9*a^2*b*c^5*d*e*g^2*h^2 - 9*a*b^2*
c^5*d^2*e*g*h^2 + 9*a*b^2*c^5*d*e^2*f*h^2 - 36*a^6*c^2*f*j*k*l*m^2 + 36*a^5
*c^3*f^2*j*k*l*m - 36*a^5*c^3*f*h^2*j*l*m + 36*a^5*c^3*e*h*j^2*l*m - 18*a^6
*b*c*j^2*k*l*m^2 + 9*a^6*b*c*j*k^2*l^2*m + 3*a^5*b^2*c*j^3*k*l*m - 36*a^5*c
^3*f*g*j*k^2*m - 36*a^5*c^3*e*f*k^2*l*m + 36*a^5*c^3*d*g*k^2*l*m - 36*a^4*c
^4*d^2*g*k*l*m - 36*a^5*c^3*e*h*j*k*l^2 - 36*a^5*c^3*e*f*j*l^2*m - 36*a^5*c
^3*d*f*k*l^2*m + 36*a^4*c^4*e^2*h*j*k*l + 36*a^4*c^4*e^2*f*j*l*m + 9*a^6*b*
c*h*k^2*l*m^2 - 3*a^4*b^3*c*h^3*k*l*m - 36*a^5*c^3*e*g*h*l^2*m + 36*a^5*c^3
*e*f*j*k*m^2 - 36*a^5*c^3*d*g*j*k*m^2 + 36*a^5*c^3*d*f*j*l*m^2 - 36*a^5*c^3
*d*e*k*l*m^2 + 36*a^4*c^4*e^2*g*h*l*m - 36*a^4*c^4*e*f^2*j*k*m - 36*a^4*c^4
*d*f^2*j*l*m + 9*a^6*b*c*h*j*l^2*m^2 + 9*a^6*b*c*g*k*l^2*m^2 + 9*a^5*b^2*c*
g*k^3*l*m + 3*a^3*b^4*c*g^3*k*l*m + 36*a^5*c^3*f*g*h*j*m^2 + 36*a^5*c^3*e*f
*h*l*m^2 - 36*a^4*c^4*f^2*g*h*j*m - 36*a^4*c^4*e*f^2*h*l*m - 24*a^4*b*c^3*f
^3*k*l*m - 12*a^5*b*c^2*h*j^3*k*m - 12*a^5*b*c^2*g*j^3*l*m - 3*a^2*b^5*c*f^
3*k*l*m - 36*a^4*c^4*e*g^2*h*k*l - 36*a^4*c^4*e*f*g^2*l*m + 12*a^5*b^2*c*e*
k*l^3*m - 6*a^5*b^2*c*f*j*l^3*m + 3*a^5*b^2*c*h*j*k*l^3 + 48*a^3*b*c^4*d^3*
k*l*m + 36*a^4*c^4*e*f*h^2*j*m + 36*a^4*c^4*d*g*h^2*k*l - 36*a^4*c^4*d*f*h^
2*k*m - 36*a^4*c^4*d*e*j^2*k*l + 24*a^5*b*c^2*d*k^3*l*m + 21*a*b^5*c^2*d^3*
k*l*m - 12*a^5*b*c^2*g*j*k^3*l - 9*a^4*b^3*c*d*k^3*l*m + 6*a^5*b*c^2*f*j*k^
3*m + 3*a^5*b^2*c*g*h*l^3*m - 36*a^4*c^4*e*f*h*j^2*l - 12*a^5*b*c^2*g*h*k^3
*m - 3*a^5*b^2*c*e*j*k*m^3 - 3*a^5*b^2*c*d*j*l*m^3 - 36*a^4*c^4*d*g*h*j*k^2
 - 36*a^4*c^4*d*f*g*k^2*l - 36*a^4*c^4*d*e*h*k^2*l - 36*a^4*c^4*d*e*g*k^2*m
 + 36*a^3*c^5*d^2*g*h*j*k + 36*a^3*c^5*d^2*f*g*k*l - 36*a^3*c^5*d^2*f*g*j*m
 + 36*a^3*c^5*d^2*e*h*k*l + 36*a^3*c^5*d^2*e*g*k*m - 36*a^3*c^5*d^2*e*f*l*m
 + 24*a^5*b^2*c*e*h*l*m^3 - 24*a^3*b*c^4*e^3*j*k*l - 12*a^5*b^2*c*f*h*k*m^3
 - 12*a^5*b^2*c*f*g*l*m^3 - 3*a^5*b^2*c*g*h*j*m^3 - 3*a^4*b^3*c*e*j*k*l^3 -
 3*a*b^5*c^2*e^3*j*k*l + 36*a^4*c^4*d*e*h*j*l^2 + 36*a^4*c^4*d*e*g*k*l^2 -
36*a^3*c^5*d*e^2*h*j*l - 36*a^3*c^5*d*e^2*g*k*l - 36*a^3*c^5*d*e^2*f*k*m +
24*a^4*b*c^3*e*h^3*k*m - 24*a^3*b*c^4*e^3*g*l*m - 18*a*b^4*c^3*d^3*j*k*l -
12*a^4*b*c^3*g*h^3*j*l - 12*a^4*b*c^3*f*h^3*k*l - 12*a^4*b*c^3*d*h^3*l*m +
12*a^3*b*c^4*e^3*h*k*m + 6*a^4*b*c^3*f*h^3*j*m - 3*a^4*b^3*c*g*h*j*l^3 - 3*
a^4*b^3*c*f*h*k*l^3 - 3*a^4*b^3*c*e*g*l^3*m - 3*a^4*b^3*c*d*h*l^3*m - 3*a*b
^5*c^2*e^3*h*k*m - 3*a*b^5*c^2*e^3*g*l*m + 36*a^4*c^4*e*f*g*h*l^2 - 36*a^4*
c^4*d*e*f*j*m^2 - 36*a^3*c^5*e^2*f*g*h*l - 36*a^3*c^5*d*f^2*g*j*k - 36*a^3*
c^5*d*e*f^2*k*l + 36*a^3*c^5*d*e*f^2*j*m - 18*a*b^4*c^3*d^3*h*k*m - 9*a*b^4
*c^3*d^3*g*l*m + 30*a^5*b*c^2*d*g*k*m^3 - 30*a^4*b^3*c*d*g*k*m^3 - 24*a^5*b
*c^2*e*f*k*m^3 - 24*a^5*b*c^2*d*f*l*m^3 + 24*a^4*b*c^3*e*g*j^3*m + 24*a^4*b
*c^3*d*h*j^3*m + 15*a^4*b^3*c*e*f*k*m^3 + 15*a^4*b^3*c*d*f*l*m^3 + 12*a^5*b
*c^2*e*g*j*m^3 + 12*a^5*b*c^2*d*h*j*m^3 - 12*a^4*b*c^3*f*h*j^3*k - 12*a^4*b
```

```
*c^3*f*g*j^3*l + 6*a^4*b^3*c*e*g*j*m^3 + 6*a^4*b^3*c*d*h*j*m^3 + 6*a^4*b*c^
3*e*h*j^3*l + 36*a^3*c^5*d*e*g^2*h*l - 24*a^5*b*c^2*f*g*h*m^3 + 15*a^4*b^3*
c*f*g*h*m^3 - 9*a*b^6*c*d^2*g*j*m^2 - 6*a^3*b^4*c*d*g*k*l^3 - 6*a*b^4*c^3*e
^3*f*j*m + 3*a^3*b^4*c*e*g*j*l^3 + 3*a^3*b^4*c*e*f*k*l^3 + 3*a^3*b^4*c*d*h*
j*l^3 + 3*a^3*b^4*c*d*e*l^3*m + 3*a*b^4*c^3*e^3*h*j*k + 3*a*b^4*c^3*e^3*g*j
*l + 3*a*b^4*c^3*e^3*f*k*l + 3*a*b^4*c^3*d*e^3*l*m - 36*a^3*c^5*d*e*g*h^2*k
 + 30*a^2*b*c^5*d^3*f*j*m - 30*a*b^3*c^4*d^3*f*j*m + 24*a^3*b*c^4*d*g^3*j*l
 - 24*a^2*b*c^5*d^3*h*j*k - 24*a^2*b*c^5*d^3*f*k*l - 24*a^2*b*c^5*d^3*e*k*m
 + 15*a*b^3*c^4*d^3*h*j*k + 15*a*b^3*c^4*d^3*f*k*l + 15*a*b^3*c^4*d^3*e*k*m
 - 12*a^3*b*c^4*e*g^3*j*k + 12*a^2*b*c^5*d^3*g*j*l + 6*a*b^3*c^4*d^3*g*j*l
+ 3*a^3*b^4*c*f*g*h*l^3 + 3*a*b^4*c^3*e^3*g*h*m + 24*a^3*b*c^4*d*g^3*h*m -
12*a^3*b*c^4*f*g^3*h*k + 12*a^2*b*c^5*d^3*g*h*m - 9*a^3*b^4*c*d*e*j*m^3 + 6
*a^3*b*c^4*e*g^3*h*l + 6*a*b^3*c^4*d^3*g*h*m + 36*a^3*c^5*d*e*f*g*k^2 - 36*
a^2*c^6*d^2*e*f*g*k - 24*a^4*b*c^3*d*e*j*l^3 - 18*a^3*b^4*c*e*f*g*m^3 - 18*
a^3*b^4*c*d*f*h*m^3 - 3*a^2*b^5*c*d*e*j*l^3 - 3*a*b^3*c^4*d*e^3*j*l - 24*a^
4*b*c^3*e*f*g*l^3 + 24*a^3*b*c^4*d*f*h^3*l + 12*a^4*b*c^3*d*f*h*l^3 - 12*a^
3*b*c^4*e*g*h^3*j - 12*a^3*b*c^4*e*f*h^3*k - 12*a^3*b*c^4*d*e*h^3*m - 12*a*
b^2*c^5*d^3*e*j*k + 6*a^3*b*c^4*d*g*h^3*k - 3*a^2*b^5*c*e*f*g*l^3 - 3*a^2*b
^5*c*d*f*h*l^3 - 3*a*b^3*c^4*e^3*g*h*j - 3*a*b^3*c^4*e^3*f*h*k - 3*a*b^3*c^
4*e^3*f*g*l - 3*a*b^3*c^4*d*e^3*h*m + 24*a*b^2*c^5*d^3*e*h*l - 12*a*b^2*c^5
*d^3*f*h*k - 3*a*b^2*c^5*d^3*g*h*j - 3*a*b^2*c^5*d^3*f*g*l - 3*a*b^2*c^5*d^
3*e*g*m + 48*a^4*b*c^3*d*e*f*m^3 + 24*a^2*b*c^5*d*e*f^3*m + 21*a^2*b^5*c*d*
e*f*m^3 - 12*a^2*b*c^5*e*f^3*g*j - 12*a^2*b*c^5*d*f^3*h*j - 9*a*b^3*c^4*d*e
*f^3*m + 6*a^2*b*c^5*d*f^3*g*k + 12*a*b^2*c^5*d*e^3*f*l - 6*a*b^2*c^5*d*e^3
*g*k + 3*a*b^2*c^5*d*e^3*h*j - 24*a^3*b*c^4*d*e*f*k^3 - 12*a^2*b*c^5*d*e*g^
3*j - 3*a*b^5*c^2*d*e*f*k^3 + 3*a*b^2*c^5*e^3*f*g*h - 12*a^2*b*c^5*d*f*g^3*
h + 9*a*b^2*c^5*d*e*f^3*j + 9*a*b*c^6*d^2*e^2*f*j + 3*a*b^4*c^3*d*e*f*j^3 +
 9*a*b*c^6*d^2*e^2*g*h + 9*a*b*c^6*d^2*e*f^2*h - 3*a*b^3*c^4*d*e*f*h^3 - 18
*a*b*c^6*d^2*e*f*g^2 + 9*a*b*c^6*d*e^2*f^2*g + 3*a*b^2*c^5*d*e*f*g^3 - 36*a
^4*b^2*c^2*e^2*k*l^2*m - 9*a^4*b^2*c^2*g^2*j^2*k*m + 45*a^3*b^3*c^2*d^2*k^2
*l*m + 36*a^4*b^2*c^2*e^2*j*l*m^2 + 9*a^4*b^2*c^2*g^2*j*k^2*l + 9*a^3*b^3*c
^2*e^2*j^2*l*m + 9*a^4*b^2*c^2*g^2*h*k^2*m - 9*a^4*b^2*c^2*f^2*h*l^2*m - 9*
a^3*b^3*c^2*f^2*j^2*k*l - 45*a^3*b^3*c^2*d^2*j*k*m^2 + 36*a^3*b^2*c^3*d^2*j
^2*k*m + 18*a^4*b^2*c^2*f^2*h*k*m^2 + 18*a^4*b^2*c^2*f^2*g*l*m^2 - 9*a^4*b^
2*c^2*g^2*h*k*l^2 - 9*a^4*b^2*c^2*f*h^2*k^2*m - 9*a^4*b^2*c^2*f*g^2*l^2*m -
 9*a^4*b^2*c^2*e*j^2*k^2*l - 9*a^4*b^2*c^2*d*j^2*k^2*m - 9*a^3*b^3*c^2*e^2*
j*k*l^2 - 9*a^2*b^4*c^2*d^2*j^2*k*m - 36*a^3*b^2*c^3*d^2*j*k^2*l - 27*a^3*b
^2*c^3*e^2*h^2*k*m + 9*a^4*b^2*c^2*g*h^2*j*l^2 + 9*a^4*b^2*c^2*f*h^2*k*l^2
- 9*a^4*b^2*c^2*f*g^2*k*m^2 - 9*a^4*b^2*c^2*e*g^2*l*m^2 - 9*a^4*b^2*c^2*d*j
^2*k*l^2 + 9*a^4*b^2*c^2*d*h^2*l^2*m - 9*a^3*b^3*c^2*e^2*g*l^2*m + 9*a^2*b^
4*c^2*e^2*h^2*k*m + 9*a^2*b^4*c^2*d^2*j*k^2*l - 45*a^3*b^3*c^2*e^2*h*j*m^2
+ 36*a^4*b^2*c^2*e*h^2*j*m^2 + 36*a^3*b^2*c^3*e^2*h*j^2*m - 36*a^3*b^2*c^3*
d^2*h*k^2*m + 36*a^2*b^3*c^3*d^2*g^2*l*m - 9*a^4*b^2*c^2*f*h*j^2*l^2 - 9*a^
4*b^2*c^2*d*h^2*k*m^2 + 9*a^3*b^3*c^2*f^2*h*j*l^2 + 9*a^3*b^3*c^2*e^2*f*l*m
^2 + 9*a^3*b^3*c^2*e*h^2*j^2*m - 9*a^3*b^2*c^3*f^2*h^2*j*l - 9*a^2*b^4*c^2*
```

e^2*h*j^2*m + 9*a^2*b^4*c^2*d^2*h*k^2*m + 36*a^3*b^2*c^3*d^2*h*k*l^2 - 27*a
^4*b^2*c^2*e*g*j^2*m^2 - 27*a^4*b^2*c^2*d*h*j^2*m^2 - 9*a^4*b^2*c^2*d*h*k^2
*l^2 - 9*a^3*b^3*c^2*e*f^2*k*m^2 - 9*a^3*b^3*c^2*d*f^2*l*m^2 + 9*a^3*b^2*c^
3*f^2*h*j^2*k + 9*a^3*b^2*c^3*f^2*g*j^2*l - 9*a^3*b^2*c^3*e^2*g*k^2*l - 9*a
^3*b^2*c^3*e^2*f*k^2*m - 9*a^3*b^2*c^3*d^2*f*l^2*m - 9*a^2*b^4*c^2*d^2*h*k*
l^2 + 9*a^2*b^3*c^3*d^2*h^2*k*l - 81*a^3*b^2*c^3*d^2*g*j*m^2 + 54*a^2*b^4*c
^2*d^2*g*j*m^2 - 45*a^3*b^3*c^2*d*g^2*j*m^2 - 45*a^2*b^3*c^3*d^2*g*j^2*m +
36*a^3*b^2*c^3*d^2*f*k*m^2 + 36*a^3*b^2*c^3*d*g^2*j^2*m + 18*a^3*b^2*c^3*e^
2*g*j*l^2 + 18*a^3*b^2*c^3*e^2*f*k*l^2 + 18*a^3*b^2*c^3*d*e^2*l^2*m - 9*a^4
*b^2*c^2*d*f*k^2*m^2 - 9*a^3*b^3*c^2*f^2*g*h*m^2 - 9*a^3*b^3*c^2*d*h^2*j*l^
2 - 9*a^3*b^2*c^3*f^2*g*j*k^2 - 9*a^3*b^2*c^3*d^2*e*l*m^2 - 9*a^3*b^2*c^3*f
*g^2*h^2*m - 9*a^3*b^2*c^3*e*g^2*j^2*l - 9*a^3*b^2*c^3*e*f^2*k^2*l - 9*a^2*
b^4*c^2*d^2*f*k*m^2 - 9*a^2*b^4*c^2*d*g^2*j^2*m - 9*a^2*b^3*c^3*e^2*h^2*j*k
 - 9*a^2*b^2*c^4*d^2*f^2*k*m - 27*a^2*b^2*c^4*d^2*g^2*j*l - 9*a^3*b^3*c^2*f
*g*h^2*l^2 + 9*a^3*b^2*c^3*e*g^2*j*k^2 - 9*a^3*b^2*c^3*e*f^2*j*l^2 - 9*a^3*
b^2*c^3*d*h^2*j^2*k - 9*a^3*b^2*c^3*d*f^2*k*l^2 - 9*a^3*b^2*c^3*d*e^2*k*m^2
 - 9*a^2*b^3*c^3*e^2*g*h^2*m - 9*a^2*b^3*c^3*d^2*h*j*k^2 - 9*a^2*b^3*c^3*d^
2*f*k^2*l - 9*a^2*b^3*c^3*d^2*e*k^2*m + 36*a^3*b^3*c^2*d*e*j^2*m^2 + 36*a^3
*b^2*c^3*e^2*f*h*m^2 - 27*a^2*b^2*c^4*d^2*g^2*h*m + 9*a^3*b^3*c^2*e*f*h^2*m
^2 + 9*a^3*b^2*c^3*f*g^2*h*k^2 - 9*a^2*b^4*c^2*e^2*f*h*m^2 + 9*a^2*b^3*c^3*
d^2*e*k*l^2 - 9*a^2*b^2*c^4*e^2*f^2*h*m - 45*a^2*b^3*c^3*d^2*g*h*l^2 - 36*a
^3*b^2*c^3*e*f^2*g*m^2 + 36*a^3*b^2*c^3*d*g^2*h*l^2 - 36*a^3*b^2*c^3*d*f^2*
h*m^2 + 36*a^2*b^2*c^4*d^2*g*h^2*l - 9*a^3*b^2*c^3*e*g*h^2*k^2 + 9*a^2*b^4*
c^2*e*f^2*g*m^2 - 9*a^2*b^4*c^2*d*g^2*h*l^2 + 9*a^2*b^4*c^2*d*f^2*h*m^2 + 9
*a^2*b^3*c^3*e^2*g*h*k^2 + 9*a^2*b^3*c^3*d*g^2*h^2*l - 9*a^2*b^3*c^3*d*e^2*
j*l^2 - 9*a^2*b^2*c^4*e^2*g^2*h*k - 9*a^2*b^2*c^4*e^2*f*g^2*m - 9*a^2*b^2*c
^4*d^2*f*j^2*k - 9*a^2*b^2*c^4*d^2*f*h^2*m - 9*a^2*b^2*c^4*d^2*e*j^2*l - 45
*a^2*b^3*c^3*d^2*f*g*m^2 + 36*a^3*b^2*c^3*d*f*g^2*m^2 - 27*a^3*b^2*c^3*d*f*
h^2*l^2 + 18*a^2*b^2*c^4*d^2*e*j*k^2 + 9*a^2*b^4*c^2*d*f*h^2*l^2 - 9*a^2*b^
4*c^2*d*f*g^2*m^2 - 9*a^2*b^3*c^3*e^2*f*g*l^2 + 9*a^2*b^2*c^4*e^2*g*h^2*j +
 9*a^2*b^2*c^4*e^2*f*h^2*k - 9*a^2*b^2*c^4*e*f^2*g^2*l - 9*a^2*b^2*c^4*d*f^
2*g^2*m - 9*a^2*b^2*c^4*d*e^2*j^2*k + 9*a^2*b^2*c^4*d*e^2*h^2*m + 18*a^4*b^
2*c^2*f^2*j^2*m^2 + 18*a^3*b^2*c^3*e^2*h^2*l^2 - 9*a^2*b^4*c^2*e^2*h^2*l^2
+ 18*a^2*b^2*c^4*d^2*g^2*k^2 + 12*a^6*c^2*j^3*k*l*m + 3*a^6*b^2*j*k*l*m^3 -
 12*a^6*c^2*g*k^3*l*m - 12*a^5*c^3*g^3*k*l*m - 24*a^6*c^2*e*k*l^3*m - 24*a^
4*c^4*e^3*k*l*m + 12*a^6*c^2*h*j*k*l^3 + 12*a^6*c^2*f*j*l^3*m + 12*a^5*c^3*
h^3*j*k*l - 3*a^5*b^3*h*j*k*m^3 - 3*a^5*b^3*g*j*l*m^3 - 3*a^5*b^3*f*k*l*m^3
 + 12*a^6*c^2*g*h*l^3*m + 12*a^5*c^3*g*h^3*l*m - 12*a^6*c^2*e*j*k*m^3 - 12*
a^6*c^2*d*j*l*m^3 - 12*a^5*c^3*f*j^3*k*l - 12*a^5*c^3*e*j^3*k*m - 12*a^5*c^
3*d*j^3*l*m - 12*a^4*c^4*f^3*j*k*l + 24*a^6*c^2*f*h*k*m^3 + 24*a^6*c^2*f*g*
l*m^3 + 24*a^4*c^4*f^3*h*k*m + 24*a^4*c^4*f^3*g*l*m - 12*a^6*c^2*g*h*j*m^3
- 12*a^6*c^2*e*h*l*m^3 - 12*a^5*c^3*g*h*j^3*m + 3*b^6*c^2*d^3*j*k*l + 3*a^4
*b^4*e*j*k*m^3 + 3*a^4*b^4*d*j*l*m^3 - 24*a^5*c^3*d*j*k^3*l - 24*a^3*c^5*d^
3*j*k*l - 6*a^4*b^4*e*h*l*m^3 + 3*b^6*c^2*d^3*h*k*m + 3*b^6*c^2*d^3*g*l*m +
 3*a^6*b*c*j^2*l^3*m + 3*a^4*b^4*g*h*j*m^3 + 3*a^4*b^4*f*h*k*m^3 + 3*a^4*b^

4*f*g*l*m^3 - 24*a^5*c^3*d*h*k^3*m - 24*a^3*c^5*d^3*h*k*m + 12*a^5*c^3*g*h*
j*k^3 + 12*a^5*c^3*f*g*k^3*l + 12*a^5*c^3*e*h*k^3*l + 12*a^5*c^3*e*g*k^3*m
+ 12*a^4*c^4*g^3*h*j*k + 12*a^4*c^4*f*g^3*k*l + 12*a^4*c^4*f*g^3*j*m + 12*a
^4*c^4*e*g^3*k*m + 12*a^4*c^4*d*g^3*l*m + 12*a^3*c^5*d^3*g*l*m + 3*a^6*b*c*
j*k^3*m^2 - 9*a^6*b*c*h^2*l*m^3 - 3*a^5*b*c^2*j^4*k*l + 24*a^5*c^3*e*g*j*l^
3 + 24*a^5*c^3*e*f*k*l^3 + 24*a^5*c^3*d*e*l^3*m + 24*a^3*c^5*e^3*g*j*l + 24
*a^3*c^5*e^3*f*k*l + 24*a^3*c^5*d*e^3*l*m - 12*a^5*c^3*d*h*j*l^3 - 12*a^5*c
^3*d*g*k*l^3 - 12*a^4*c^4*e*h^3*j*k - 12*a^4*c^4*d*h^3*j*l - 12*a^3*c^5*e^3
*h*j*k - 12*a^3*c^5*e^3*f*j*m + 9*a^4*b*c^3*g^4*l*m + 6*b^5*c^3*d^3*f*j*m +
6*a^3*b^5*d*g*k*m^3 - 3*b^5*c^3*d^3*h*j*k - 3*b^5*c^3*d^3*g*j*l - 3*b^5*c^
3*d^3*f*k*l - 3*b^5*c^3*d^3*e*k*m - 3*a^3*b^5*e*g*j*m^3 - 3*a^3*b^5*e*f*k*m
^3 - 3*a^3*b^5*d*h*j*m^3 - 3*a^3*b^5*d*f*l*m^3 - 12*a^5*c^3*f*g*h*l^3 - 12*
a^4*c^4*f*g*h^3*l - 12*a^4*c^4*e*g*h^3*m - 12*a^3*c^5*e^3*g*h*m - 9*a^6*b*c
*g*k^2*m^3 - 3*b^5*c^3*d^3*g*h*m + 3*a^6*b*c*f*l^3*m^2 - 3*a^3*b^5*f*g*h*m^
3 + 12*a^5*c^3*d*e*j*m^3 + 12*a^4*c^4*e*f*j^3*k + 12*a^4*c^4*d*g*j^3*k + 12
*a^4*c^4*d*f*j^3*l + 12*a^4*c^4*d*e*j^3*m + 12*a^3*c^5*e*f^3*j*k + 12*a^3*c
^5*d*f^3*j*l - 9*a^6*b*c*e*l^2*m^3 - 24*a^5*c^3*e*f*g*m^3 - 24*a^5*c^3*d*f*
h*m^3 - 24*a^3*c^5*e*f^3*g*m - 24*a^3*c^5*d*f^3*h*m - 15*a^2*b*c^5*d^4*l*m
+ 15*a*b^3*c^4*d^4*l*m + 12*a^4*c^4*f*g*h*j^3 + 12*a^3*c^5*f^3*g*h*j + 12*a
^3*c^5*e*f^3*h*l + 9*a^3*b*c^4*f^4*k*l - 9*a^3*b*c^4*f^4*j*m + 3*b^4*c^4*d^
3*e*j*k + 3*a^5*b^2*c*g*j*l^4 + 3*a^5*b^2*c*f*k*l^4 + 3*a^5*b^2*c*d*l^4*m -
3*a^5*b*c^2*h*j*k^4 - 3*a^5*b*c^2*f*k^4*l - 3*a^5*b*c^2*e*k^4*m - 3*a^4*b*
c^3*h^4*j*k + 3*a^2*b^6*d*e*j*m^3 + 3*a*b^4*c^3*e^4*k*m + 24*a^4*c^4*d*e*j*
k^3 + 24*a^2*c^6*d^3*e*j*k - 6*b^4*c^4*d^3*e*h*l + 3*b^4*c^4*d^3*g*h*j + 3*
b^4*c^4*d^3*f*h*k + 3*b^4*c^4*d^3*f*g*l + 3*b^4*c^4*d^3*e*g*m - 3*a^4*b*c^3
*g*h^4*m + 3*a^2*b^6*e*f*g*m^3 + 3*a^2*b^6*d*f*h*m^3 - 3*a*b^6*c*e^3*j*m^2
+ 24*a^4*c^4*d*f*h*k^3 + 24*a^2*c^6*d^3*f*h*k - 12*a^4*c^4*e*f*g*k^3 - 12*a
^3*c^5*e*f*g^3*k - 12*a^3*c^5*d*g^3*h*j - 12*a^3*c^5*d*f*g^3*l - 12*a^3*c^5
*d*e*g^3*m - 12*a^2*c^6*d^3*g*h*j - 12*a^2*c^6*d^3*f*g*l - 12*a^2*c^6*d^3*e
*h*l - 12*a^2*c^6*d^3*e*g*m - 12*a*b^2*c^5*d^4*j*l + 9*a^5*b*c^2*d*j*l^4 +
9*a^2*b*c^5*e^4*j*k - 3*a^4*b^3*c*d*j*l^4 - 3*a^4*b*c^3*e*j^4*k - 3*a^4*b*c
^3*d*j^4*l - 3*a*b^3*c^4*e^4*j*k - 24*a^4*c^4*d*e*f*l^3 - 24*a^2*c^6*d*e^3*
f*l - 12*a^5*b^2*c*e*g*m^4 - 12*a^5*b^2*c*d*h*m^4 + 12*a^3*c^5*d*e*h^3*j +
12*a^2*c^6*d*e^3*h*j + 12*a^2*c^6*d*e^3*g*k - 12*a*b^2*c^5*d^4*h*m + 9*a^5*
b*c^2*f*g*l^4 - 9*a^5*b*c^2*e*h*l^4 - 9*a^2*b*c^5*e^4*h*l + 9*a^2*b*c^5*e^4
*g*m + 6*a^4*b^3*c*e*h*l^4 + 6*a*b^3*c^4*e^4*h*l - 3*b^3*c^5*d^3*e*g*j - 3*
b^3*c^5*d^3*e*f*k - 3*a^4*b^3*c*f*g*l^4 - 3*a^4*b*c^3*g*h*j^4 - 3*a^3*b*c^4
*g^4*h*j - 3*a^3*b*c^4*f*g^4*l - 3*a^3*b*c^4*e*g^4*m - 3*a*b^3*c^4*e^4*g*m
+ 12*a^3*c^5*e*f*g*h^3 + 12*a^2*c^6*e^3*f*g*h - 3*b^3*c^5*d^3*f*g*h - 12*a^
3*c^5*d*e*f*j^3 - 12*a^2*c^6*d*e*f^3*j - 3*a*b^6*c*d^2*g*l^3 - 15*a^5*b*c^2
*d*e*m^4 + 15*a^4*b^3*c*d*e*m^4 + 9*a^4*b*c^3*e*f*k^4 - 9*a^4*b*c^3*d*g*k^4
+ 3*a^3*b^4*c*d*f*l^4 - 3*a^3*b*c^4*d*h^4*j - 3*a^2*b*c^5*e*f^4*k - 3*a^2*
b*c^5*d*f^4*l + 3*a*b^2*c^5*e^4*g*j + 3*a*b^2*c^5*e^4*f*k + 3*a*b^2*c^5*d*e
^4*m - 9*a*b*c^6*d^3*e^2*l + 3*b^2*c^6*d^3*e*f*g - 3*a^3*b*c^4*f*g*h^4 - 3*
a^2*b*c^5*f^4*g*h + 12*a^2*c^6*d*e*f*g^3 - 9*a*b*c^6*d^3*f^2*j + 3*a*b*c^6*

d^2*e^3*k + 9*a^3*b*c^4*d*e*j^4 - 3*a^2*b*c^5*e*f*g^4 - 9*a*b*c^6*d^3*e*h^2 + 3*a*b*c^6*d^2*f^3*g + 3*a*b*c^6*d*e^3*g^2 - 3*a^4*b^2*c^2*h^3*j^2*m + 12*a^4*b^2*c^2*g^3*j*m^2 - 3*a^4*b^2*c^2*f^2*k^3*m + 3*a^3*b^3*c^2*g^3*j^2*m - 9*a^3*b^4*c*f^2*j^2*m^2 + 9*a^3*b^3*c^2*f^2*j^3*m - 6*a^3*b^3*c^2*f^3*j*m^2 - 6*a^3*b^2*c^3*f^3*j^2*m - 3*a^2*b^4*c^2*f^3*j^2*m - 27*a^4*b^2*c^2*d^2*k*m^3 - 27*a^3*b^2*c^3*e^3*j*m^2 + 18*a^2*b^4*c^2*e^3*j*m^2 - 15*a^2*b^3*c^3*e^3*j^2*m + 12*a^4*b^2*c^2*f^2*j*l^3 + 3*a^3*b^3*c^2*e^2*k^3*l + 42*a^2*b^3*c^3*d^3*j*m^2 - 27*a^2*b^2*c^4*d^3*j^2*m - 15*a^3*b^3*c^2*d^2*k*l^3 - 3*a^4*b^2*c^2*f*j^2*k^3 - 3*a^4*b^2*c^2*f*h^3*m^2 + 3*a^3*b^3*c^2*g^3*h*l^2 + 3*a^3*b^3*c^2*f^2*j*k^3 - 3*a^3*b^2*c^3*g^3*h^2*l - 3*a^3*b^2*c^3*e^2*j^3*l - 27*a^4*b^2*c^2*e^2*h*m^3 + 12*a^3*b^2*c^3*f^3*h*l^2 + 3*a^3*b^3*c^2*f*g^3*m^2 - 3*a^2*b^4*c^2*f^3*h*l^2 + 3*a^2*b^3*c^3*f^3*h^2*l + 9*a^3*b^3*c^2*e*h^3*l^2 + 9*a^2*b^3*c^3*e^2*h^3*l - 6*a^4*b^2*c^2*e*h^2*l^3 - 6*a^3*b^3*c^2*e^2*h*l^3 - 6*a^2*b^3*c^3*e^3*h*l^2 - 6*a^2*b^2*c^4*e^3*h^2*l + 3*a^2*b^3*c^3*d^2*j^3*k + 42*a^3*b^3*c^2*d^2*g*m^3 - 27*a^4*b^2*c^2*d*g^2*m^3 - 27*a^2*b^2*c^4*d^3*h*l^2 - 15*a^2*b^3*c^3*e^3*f*m^2 + 12*a^3*b^2*c^3*e^2*h*k^3 + 3*a^3*b^3*c^2*e*h^2*k^3 - 3*a^3*b^2*c^3*e*g^3*l^2 - 3*a^2*b^4*c^2*e^2*h*k^3 + 3*a^2*b^3*c^3*f^3*g*k^2 - 3*a^2*b^2*c^4*f^3*g^2*k - 27*a^3*b^2*c^3*d^2*g*l^3 - 27*a^2*b^2*c^4*d^3*f*m^2 + 18*a^2*b^4*c^2*d^2*g*l^3 - 15*a^3*b^3*c^2*d*g^2*l^3 + 12*a^2*b^2*c^4*e^3*g*k^2 - 3*a^3*b^2*c^3*e*h^2*j^3 + 3*a^2*b^3*c^3*e^2*h*j^3 + 3*a^2*b^3*c^3*e*f^3*l^2 - 3*a^2*b^2*c^4*d^2*h^3*k + 9*a^2*b^3*c^3*d*g^3*k^2 - 9*a*b^4*c^3*d^2*g^2*k^2 - 6*a^3*b^2*c^3*d*g^2*k^3 - 6*a^2*b^3*c^3*d^2*g*k^3 - 3*a^2*b^4*c^2*d*g^2*k^3 + 12*a^2*b^2*c^4*d^2*g*j^3 + 3*a^2*b^3*c^3*d*g^2*j^3 - 3*a^2*b^2*c^4*d*f^3*k^2 - 3*a^2*b^2*c^4*d*g^2*h^3 + 12*a^7*c*j*k*l*m^3 - 3*b^7*c*d^3*k*l*m - 3*a^6*b*c*k^4*l*m - 3*a^6*b*c*j*k*l^4 - 3*a^6*b*c*g*l^4*m - 9*a^6*b*c*f*j*m^4 + 9*a^6*b*c*e*k*m^4 + 9*a^6*b*c*d*l*m^4 + 9*a^6*b*c*g*h*m^4 - 3*a*b^7*d*e*f*m^3 + 9*a*b*c^6*d^4*h*j - 9*a*b*c^6*d^4*g*k + 9*a*b*c^6*d^4*f*l + 9*a*b*c^6*d^4*e*m + 12*a*c^7*d^3*e*f*g - 3*a*b*c^6*d*e^4*j - 3*a*b*c^6*e^4*f*g - 3*a*b*c^6*d*e*f^4 + 18*a^6*c^2*h^2*j*l*m^2 - 18*a^6*c^2*h*j^2*l^2*m + 18*a^6*c^2*f*k^2*l^2*m + 36*a^5*c^3*e^2*k*l^2*m + 18*a^6*c^2*g*j*k^2*m^2 + 18*a^6*c^2*e*k^2*l*m^2 + 18*a^5*c^3*g^2*j^2*k*m + 18*a^6*c^2*e*j*l^2*m^2 + 18*a^6*c^2*d*k*l^2*m^2 - 18*a^5*c^3*e^2*j*l*m^2 - 18*a^6*c^2*f*h*l^2*m^2 + 18*a^5*c^3*f^2*h*l^2*m - 36*a^5*c^3*f^2*h*k*m^2 - 36*a^5*c^3*f^2*g*l*m^2 + 18*a^5*c^3*g^2*h*k*l^2 - 18*a^5*c^3*g*h^2*k^2*l + 18*a^5*c^3*f*h^2*k^2*m + 18*a^5*c^3*f*g^2*l^2*m + 18*a^5*c^3*e*j^2*k^2*l + 18*a^5*c^3*d*j^2*k^2*m - 18*a^4*c^4*d^2*j^2*k*m + 36*a^4*c^4*d^2*j*k^2*l + 18*a^5*c^3*f*g^2*k*m^2 + 18*a^5*c^3*e*g^2*l*m^2 + 18*a^5*c^3*d*j^2*k*l^2 - 18*a^4*c^4*f^2*g^2*k*m + 36*a^4*c^4*d^2*h*k^2*m + 18*a^5*c^3*f*h*j^2*l^2 - 18*a^5*c^3*e*h^2*j*m^2 + 18*a^5*c^3*d*h^2*k*m^2 + 18*a^4*c^4*f^2*h^2*j*l - 18*a^4*c^4*e^2*h*j^2*m - 18*a^5*c^3*e*g*k^2*l^2 + 18*a^5*c^3*d*h*k^2*l^2 + 18*a^4*c^4*e^2*g*k^2*l + 18*a^4*c^4*e^2*f*k^2*m - 18*a^4*c^4*d^2*h*k*l^2 + 18*a^4*c^4*d^2*f*l^2*m - 36*a^4*c^4*e^2*g*j*l^2 - 36*a^4*c^4*e^2*f*k*l^2 - 36*a^4*c^4*d*e^2*l^2*m + 18*a^5*c^3*d*f*k^2*m^2 + 18*a^4*c^4*f^2*g*j*k^2 + 18*a^4*c^4*d^2*g*j*m^2 - 18*a^4*c^4*d^2*f*k*m^2 + 18*a^4*c^4*d^2*e*l*m^2 - 18*a^4*c^4*f*g^2*j^2*k + 18*a^4*c^4*f*g^2*h^2*m + 18*a^

4*c^4*e*g^2*j^2*l + 18*a^4*c^4*e*f^2*k^2*l - 18*a^4*c^4*d*g^2*j^2*m - 18*a^
4*c^4*d*f^2*k^2*m + 18*a^3*c^5*d^2*f^2*k*m + 3*a^4*b^2*c^2*h^4*k*m - 3*a^3*
b^3*c^2*g^4*l*m + 18*a^4*c^4*e*f^2*j*l^2 + 18*a^4*c^4*d*h^2*j^2*k + 18*a^4*
c^4*d*f^2*k*l^2 + 18*a^4*c^4*d*e^2*k*m^2 - 18*a^3*c^5*e^2*f^2*j*l + 12*a^5*
b^2*c*g^2*k*m^3 - 9*a^5*b*c^2*h^3*j*m^2 - 9*a^5*b*c^2*f^2*l^3*m + 3*a^5*b*c
^2*h^2*k^3*l + 3*a^4*b^3*c*h^3*j*m^2 + 3*a^4*b^3*c*f^2*l^3*m - 18*a^4*c^4*e
^2*f*h*m^2 + 18*a^3*c^5*e^2*f^2*h*m + 15*a^5*b*c^2*e^2*l*m^3 - 15*a^4*b^3*c
*e^2*l*m^3 - 9*a^5*b*c^2*g^2*k*l^3 - 9*a^4*b*c^3*g^3*j^2*m - 3*a^5*b^2*c*g*
k^2*l^3 + 3*a^5*b*c^2*h*j^3*l^2 + 3*a^4*b^3*c*g^2*k*l^3 - 3*a^3*b^4*c*g^3*j
*m^2 + 36*a^4*c^4*e*f^2*g*m^2 + 36*a^4*c^4*d*f^2*h*m^2 + 18*a^4*c^4*e*g*h^2
*k^2 - 18*a^4*c^4*d*g^2*h*l^2 - 18*a^4*c^4*d*f*j^2*k^2 + 18*a^3*c^5*e^2*g^2
*h*k + 18*a^3*c^5*e^2*f*g^2*m - 18*a^3*c^5*d^2*g*h^2*l + 18*a^3*c^5*d^2*f*j
^2*k + 18*a^3*c^5*d^2*f*h^2*m + 18*a^3*c^5*d^2*e*j^2*l - 12*a^2*b^2*c^4*e^4
*k*m + 9*a^4*b^3*c*f*j^3*m^2 - 9*a^4*b^2*c^2*f*j^4*m - 6*a^5*b^2*c*f*j^2*m^
3 + 6*a^5*b*c^2*f^2*j*m^3 - 6*a^5*b*c^2*f*j^3*m^2 - 6*a^4*b^3*c*f^2*j*m^3 +
 6*a^4*b*c^3*f^3*j*m^2 - 6*a^4*b*c^3*f^2*j^3*m + 6*a^2*b^3*c^3*f^4*j*m + 3*
a^3*b^2*c^3*g^4*j*l + 3*a^2*b^5*c*f^3*j*m^2 - 3*a^2*b^3*c^3*f^4*k*l - 36*a^
3*c^5*d^2*e*j*k^2 - 18*a^4*c^4*d*f*g^2*m^2 + 18*a^3*c^5*e*f^2*g^2*l + 18*a^
3*c^5*d*f^2*g^2*m + 18*a^3*c^5*d*e^2*j^2*k + 18*a^3*b^4*c*d^2*k*m^3 + 15*a^
3*b*c^4*e^3*j^2*m + 12*a^5*b^2*c*d*k^2*m^3 - 9*a^5*b*c^2*f*j^2*l^3 - 9*a^4*
b*c^3*e^2*k^3*l + 3*a^5*b*c^2*e*k^3*l^2 + 3*a^4*b^3*c*f*j^2*l^3 + 3*a^4*b*c
^3*g^2*j^3*k - 3*a^3*b^4*c*f^2*j*l^3 + 3*a^3*b^2*c^3*g^4*h*m + 3*a*b^5*c^2*
e^3*j^2*m - 36*a^3*c^5*d^2*f*h*k^2 - 21*a^3*b*c^4*d^3*j*m^2 - 21*a*b^5*c^2*
d^3*j*m^2 + 18*a^3*c^5*e^2*f*h*j^2 - 18*a^3*c^5*e*f^2*h^2*j + 18*a^3*c^5*d*
f^2*h^2*k + 18*a*b^4*c^3*d^3*j^2*m + 15*a^4*b*c^3*d^2*k*l^3 - 9*a^5*b*c^2*d
*k^2*l^3 - 9*a^4*b*c^3*g^3*h*l^2 - 9*a^4*b*c^3*f^2*j*k^3 + 3*a^4*b^3*c*d*k^
2*l^3 + 3*a^2*b^5*c*d^2*k*l^3 - 18*a^3*c^5*d^2*e*g*l^2 + 18*a^3*c^5*d*e^2*h
*k^2 + 18*a^3*b^4*c*e^2*h*m^3 - 18*a^2*c^6*d^2*e^2*h*k + 18*a^2*c^6*d^2*e^2
*g*l + 18*a^2*c^6*d^2*e^2*f*m + 15*a^5*b*c^2*e*h^2*m^3 - 15*a^4*b^3*c*e*h^2
*m^3 - 9*a^4*b*c^3*f*g^3*m^2 - 9*a^3*b*c^4*f^3*h^2*l + 3*a^4*b^2*c^2*e*j*k^
4 + 3*a^4*b*c^3*g*h^3*k^2 + 3*a^3*b*c^4*f^2*g^3*m + 36*a^3*c^5*d*e^2*f*l^2
+ 18*a^3*c^5*d*f*g^2*j^2 + 18*a^2*c^6*d^2*f^2*g*j + 18*a^2*c^6*d^2*e*f^2*l
- 9*a^3*b^2*c^3*e*h^4*l - 9*a^3*b*c^4*d^2*j^3*k + 6*a^4*b*c^3*e^2*h*l^3 - 6
*a^4*b*c^3*e*h^3*l^2 + 6*a^3*b*c^4*e^3*h*l^2 - 6*a^3*b*c^4*e^2*h^3*l + 3*a^
4*b^2*c^2*f*h*k^4 + 3*a^4*b*c^3*d*j^3*k^2 - 3*a^3*b^4*c*e*h^2*l^3 + 3*a^2*b
^5*c*e^2*h*l^3 + 3*a^2*b^2*c^4*f^4*h*k + 3*a^2*b^2*c^4*f^4*g*l + 3*a*b^5*c^
2*e^3*h*l^2 - 3*a*b^4*c^3*e^3*h^2*l - 21*a^4*b*c^3*d^2*g*m^3 - 21*a^2*b^5*c
*d^2*g*m^3 + 18*a^3*b^4*c*d*g^2*m^3 + 18*a^2*c^6*d*e^2*f^2*k + 18*a*b^4*c^3
*d^3*h*l^2 + 15*a^3*b*c^4*e^3*f*m^2 + 15*a^2*b*c^5*d^3*h^2*l - 15*a*b^3*c^4
*d^3*h^2*l - 9*a^4*b*c^3*e*h^2*k^3 - 9*a^3*b*c^4*f^3*g*k^2 - 9*a^2*b*c^5*e^
3*f^2*m + 3*a^3*b*c^4*f^2*h^3*j + 3*a*b^5*c^2*e^3*f*m^2 + 3*a*b^3*c^4*e^3*f
^2*m + 18*a*b^4*c^3*d^3*f*m^2 + 15*a^4*b*c^3*d*g^2*l^3 + 12*a*b^2*c^5*d^3*f
^2*m - 9*a^3*b*c^4*e^2*h*j^3 - 9*a^3*b*c^4*e*f^3*l^2 - 9*a^2*b*c^5*e^3*g^2*
k + 3*a^3*b*c^4*f*g^3*j^2 + 3*a^2*b^5*c*d*g^2*l^3 + 3*a^2*b*c^5*e^2*f^3*l -
 3*a*b^4*c^3*e^3*g*k^2 + 3*a*b^3*c^4*e^3*g^2*k + 18*a^2*c^6*d^2*e*g*h^2 - 1

8*a^2*c^6*d*e^2*g^2*h - 12*a^4*b^2*c^2*d*f*l^4 - 9*a^2*b^2*c^4*d*g^4*k + 9*
a*b^3*c^4*d^2*g^3*k + 6*a^3*b^3*c^2*d*g*k^4 + 6*a^3*b*c^4*d^2*g*k^3 - 6*a^3
*b*c^4*d*g^3*k^2 + 6*a^2*b*c^5*d^3*g*k^2 - 6*a^2*b*c^5*d^2*g^3*k - 6*a*b^3*
c^4*d^3*g*k^2 - 6*a*b^2*c^5*d^3*g^2*k - 3*a^3*b^3*c^2*e*f*k^4 + 3*a^3*b^2*c
^3*e*g*j^4 + 3*a^3*b^2*c^3*d*h*j^4 + 3*a*b^5*c^2*d^2*g*k^3 + 15*a^2*b*c^5*d
^3*e*l^2 - 15*a*b^3*c^4*d^3*e*l^2 - 9*a^3*b*c^4*d*g^2*j^3 - 9*a^2*b*c^5*e^3
*f*j^2 - 3*a*b^4*c^3*d^2*g*j^3 + 3*a*b^3*c^4*e^3*f*j^2 - 3*a*b^2*c^5*e^3*f^
2*j + 12*a*b^2*c^5*d^3*f*j^2 - 9*a^2*b*c^5*d*e^3*k^2 + 3*a^2*b*c^5*e^2*g^3*
h + 3*a*b^3*c^4*d*e^3*k^2 - 9*a^2*b*c^5*d^2*g*h^3 - 3*a^2*b^3*c^3*d*e*j^4 +
 3*a^2*b*c^5*e*f^3*h^2 + 3*a*b^3*c^4*d^2*g*h^3 + 3*a^2*b^2*c^4*d*f*h^4 - 9*
a^7*c*k^2*l^2*m^2 - 6*a^6*c^2*j^2*k^3*m - 3*a^6*b^2*h*l^2*m^3 + 3*a^5*b^3*h
^2*l*m^3 - 6*a^6*c^2*g^2*k*m^3 - 6*a^6*c^2*h*k^3*l^2 + 6*a^5*c^3*h^3*j^2*m
+ 6*a^6*c^2*g*k^2*l^3 - 6*a^6*c^2*f*k^3*m^2 - 6*a^5*c^3*h^2*j^3*l - 6*a^5*c
^3*g^3*j*m^2 + 6*a^5*c^3*f^2*k^3*m + 3*a^5*b^3*g*k^2*m^3 - 3*a^4*b^4*g^2*k*
m^3 + 12*a^6*c^2*f*j^2*m^3 + 12*a^4*c^4*f^3*j^2*m + 3*a^5*b^3*e*l^2*m^3 + 3
*a^3*b^5*e^2*l*m^3 - 6*a^6*c^2*d*k^2*m^3 - 6*a^5*c^3*f^2*j*l^3 + 6*a^5*c^3*
d^2*k*m^3 - 6*a^5*c^3*g*j^3*k^2 + 6*a^4*c^4*e^3*j*m^2 - 3*b^6*c^2*d^3*j^2*m
 - 3*a^4*b^4*f*j^2*m^3 + 3*a^3*b^5*f^2*j*m^3 + 6*a^5*c^3*f*j^2*k^3 + 6*a^5*
c^3*f*h^3*m^2 - 6*a^5*c^3*e*j^3*l^2 + 6*a^4*c^4*g^3*h^2*l - 6*a^4*c^4*f^2*h
^3*m + 6*a^4*c^4*e^2*j^3*l + 6*a^3*c^5*d^3*j^2*m - 3*a^4*b^4*d*k^2*m^3 - 3*
a^2*b^6*d^2*k*m^3 + 6*a^5*c^3*e^2*h*m^3 - 6*a^4*c^4*g^2*h^3*k - 6*a^4*c^4*f
^3*h*l^2 + 12*a^5*c^3*e*h^2*l^3 + 12*a^3*c^5*e^3*h^2*l - 3*b^6*c^2*d^3*h*l^
2 + 3*b^5*c^3*d^3*h^2*l - 3*a^5*b^2*c*j^4*m^2 + 3*a^3*b^5*e*h^2*m^3 - 3*a^2
*b^6*e^2*h*m^3 + 6*a^5*c^3*d*g^2*m^3 - 6*a^4*c^4*e^2*h*k^3 - 6*a^4*c^4*f*h^
3*j^2 + 6*a^4*c^4*e*g^3*l^2 + 6*a^3*c^5*f^3*g^2*k - 6*a^3*c^5*e^2*g^3*l + 6
*a^3*c^5*d^3*h*l^2 - 3*b^6*c^2*d^3*f*m^2 - 3*b^4*c^4*d^3*f^2*m + 6*a^4*c^4*
d^2*g*l^3 + 6*a^4*c^4*e*h^2*j^3 - 6*a^4*c^4*d*h^3*k^2 - 6*a^3*c^5*f^2*g^3*j
 - 6*a^3*c^5*e^3*g*k^2 + 6*a^3*c^5*d^3*f*m^2 + 6*a^3*c^5*d^2*h^3*k - 6*a^2*
c^6*d^3*f^2*m + 4*a^5*b^2*c*h^3*m^3 + 3*b^5*c^3*d^3*g*k^2 - 3*b^4*c^4*d^3*g
^2*k - 3*a^2*b^6*d*g^2*m^3 + a^5*b*c^2*j^3*k^3 + 12*a^4*c^4*d*g^2*k^3 + 12*
a^2*c^6*d^3*g^2*k + 6*a^5*b*c^2*h^3*l^3 + 5*a^5*b*c^2*g^3*m^3 - 5*a^4*b^3*c
*g^3*m^3 + 3*b^5*c^3*d^3*e*l^2 + 3*b^3*c^5*d^3*e^2*l - 3*a^5*b^2*c*h^2*l^4
+ a^4*b^3*c*h^3*l^3 + 12*a^5*b^2*c*f^2*m^4 - 6*a^3*c^5*d^2*g*j^3 + 6*a^3*c^
5*d*f^3*k^2 + 6*a^3*b^4*c*f^3*m^3 + 6*a^2*c^6*e^3*f^2*j - 6*a^2*c^6*d^2*f^3
*k - 3*b^4*c^4*d^3*f*j^2 + 3*b^3*c^5*d^3*f^2*j - 3*a^2*b^2*c^4*f^5*m - 7*a^
4*b*c^3*e^3*m^3 - 7*a^2*b^5*c*e^3*m^3 + 6*a^4*b*c^3*g^3*k^3 - 6*a^3*c^5*e*g
^3*h^2 - 6*a^2*c^6*d^3*f*j^2 + 5*a^4*b*c^3*f^3*l^3 + a^4*b*c^3*h^3*j^3 + a^
2*b^5*c*f^3*l^3 + 6*a^3*c^5*d*g^2*h^3 - 6*a^2*c^6*e^2*f^3*h - 3*a^3*b^4*c*e
^2*l^4 - 3*a*b^4*c^3*e^4*l^2 - 7*a^3*b*c^4*d^3*l^3 - 7*a*b^5*c^2*d^3*l^3 +
6*a^3*b*c^4*f^3*j^3 + 5*a^3*b*c^4*e^3*k^3 + 3*b^3*c^5*d^3*e*h^2 - 3*b^2*c^6
*d^3*e^2*h + a*b^5*c^2*e^3*k^3 + 12*a*b^2*c^5*d^4*k^2 - 6*a^2*c^6*d*f^3*g^2
 + 6*a*b^4*c^3*d^3*k^3 - 3*a^4*b^2*c^2*d*k^5 + a^3*b*c^4*g^3*h^3 + 5*a^2*b*
c^5*d^3*j^3 - 5*a*b^3*c^4*d^3*j^3 - 9*a*c^7*d^2*e^2*f^2 + 6*a^2*b*c^5*e^3*h
^3 - 3*a*b^2*c^5*e^4*h^2 + a^2*b*c^5*f^3*g^3 + a*b^3*c^4*e^3*h^3 + 4*a*b^2*
c^5*d^3*h^3 - 3*a*b^2*c^5*d^2*g^4 - 6*a^7*c*j*l^3*m^2 + 6*a^7*c*h*l^2*m^3 +

```
 6*a^6*c^2*j*k^4*l + 6*a^6*c^2*h*k^4*m - 6*a^5*c^3*h^4*k*m + 3*a^6*b^2*h*k*
m^4 + 3*a^6*b^2*g*l*m^4 - 3*b^5*c^3*d^4*l*m - 6*a^6*c^2*g*j*l^4 - 6*a^6*c^2
*f*k*l^4 - 6*a^6*c^2*d*l^4*m + 6*a^5*c^3*h*j^4*k + 6*a^5*c^3*g*j^4*l + 6*a^
5*c^3*f*j^4*m - 6*a^4*c^4*g^4*j*l + 6*a^3*c^5*e^4*k*m + 6*a^5*b^3*f*j*m^4 -
 6*a^4*c^4*g^4*h*m + 3*b^7*c*d^3*j*m^2 - 3*a^5*b^3*e*k*m^4 - 3*a^5*b^3*d*l*
m^4 + 3*b^4*c^4*d^4*j*l - 3*a^5*b^3*g*h*m^4 - 6*a^5*c^3*e*j*k^4 + 6*a^2*c^6
*d^4*j*l + 3*b^4*c^4*d^4*h*m + 6*a^6*c^2*e*g*m^4 + 6*a^6*c^2*d*h*m^4 + 6*a^
6*b*c*j^3*m^3 - 6*a^5*c^3*f*h*k^4 + 6*a^4*c^4*g*h^4*j + 6*a^4*c^4*f*h^4*k +
 6*a^4*c^4*e*h^4*l + 6*a^4*c^4*d*h^4*m - 6*a^3*c^5*f^4*h*k - 6*a^3*c^5*f^4*
g*l + 6*a^2*c^6*d^4*h*m + 3*a^5*b*c^2*j^5*m + a^6*b*c*k^3*l^3 + 3*a^4*b^4*e
*g*m^4 + 3*a^4*b^4*d*h*m^4 + 6*b^3*c^5*d^4*g*k - 3*b^3*c^5*d^4*h*j - 3*b^3*
c^5*d^4*f*l - 3*b^3*c^5*d^4*e*m + 3*a*b^7*d^2*g*m^3 + 6*a^5*c^3*d*f*l^4 - 6
*a^4*c^4*e*g*j^4 - 6*a^4*c^4*d*h*j^4 + 6*a^3*c^5*e*g^4*j + 6*a^3*c^5*d*g^4*
k - 6*a^2*c^6*e^4*g*j - 6*a^2*c^6*e^4*f*k - 6*a^2*c^6*d*e^4*m + 3*a^4*b*c^3
*h^5*l + 6*a^3*c^5*f*g^4*h - 3*a^3*b^5*d*e*m^4 + 3*b^2*c^6*d^4*e*j + 3*a^5*
b*c^2*g*k^5 + 3*a^3*b*c^4*g^5*k + 8*a*b^6*c*d^3*m^3 + 3*b^2*c^6*d^4*f*h - 3
*a^5*b^2*c*e*l^5 - 3*a*b^2*c^5*e^5*l - 6*a^3*c^5*d*f*h^4 + 6*a^2*c^6*e*f^4*
g + 6*a^2*c^6*d*f^4*h + 3*a^4*b*c^3*f*j^5 + 3*a^2*b*c^5*f^5*j + 6*a*c^7*d^3
*e^2*h - 6*a*c^7*d^2*e^3*g + 3*a^3*b*c^4*e*h^5 + 6*a*b*c^6*d^3*g^3 + 3*a^2*
b*c^5*d*g^5 + a*b*c^6*e^3*f^3 - 9*a^6*c^2*j^2*k^2*l^2 - 9*a^6*c^2*h^2*k^2*m
^2 - 9*a^6*c^2*g^2*l^2*m^2 - 18*a^5*c^3*f^2*j^2*m^2 - 9*a^5*c^3*h^2*j^2*k^2
 - 9*a^5*c^3*g^2*j^2*l^2 - 9*a^5*c^3*f^2*k^2*l^2 - 9*a^5*c^3*e^2*k^2*m^2 -
9*a^5*c^3*d^2*l^2*m^2 - 9*a^5*c^3*g^2*h^2*m^2 - 9*a^4*c^4*e^2*j^2*k^2 - 9*a
^4*c^4*d^2*j^2*l^2 - 18*a^4*c^4*e^2*h^2*l^2 - 9*a^4*c^4*g^2*h^2*j^2 - 9*a^4
*c^4*f^2*h^2*k^2 - 9*a^4*c^4*f^2*g^2*l^2 - 9*a^4*c^4*e^2*g^2*m^2 - 9*a^4*c^
4*d^2*h^2*m^2 - 18*a^3*c^5*d^2*g^2*k^2 - 9*a^3*c^5*e^2*g^2*j^2 - 9*a^3*c^5*
e^2*f^2*k^2 - 9*a^3*c^5*d^2*h^2*j^2 - 9*a^3*c^5*d^2*f^2*l^2 - 9*a^3*c^5*d^2
*e^2*m^2 - 3*a^4*b^2*c^2*h^4*l^2 - 18*a^4*b^2*c^2*f^3*m^3 + 12*a^3*b^2*c^3*
f^4*m^2 - 9*a^3*c^5*f^2*g^2*h^2 + 4*a^4*b^2*c^2*g^3*l^3 - 3*a^2*b^4*c^2*f^4
*m^2 + 14*a^3*b^3*c^2*e^3*m^3 - 5*a^3*b^3*c^2*f^3*l^3 - 3*a^4*b^2*c^2*g^2*k
^4 - 3*a^3*b^2*c^3*g^4*k^2 + a^3*b^3*c^2*g^3*k^3 - 20*a^2*b^4*c^2*d^3*m^3 -
 18*a^3*b^2*c^3*e^3*l^3 + 16*a^3*b^2*c^3*d^3*m^3 + 12*a^4*b^2*c^2*e^2*l^4 +
 12*a^2*b^2*c^4*e^4*l^2 - 9*a^2*c^6*d^2*e^2*j^2 + 6*a^2*b^4*c^2*e^3*l^3 + 4
*a^3*b^2*c^3*f^3*k^3 + 14*a^2*b^3*c^3*d^3*l^3 - 9*a^2*c^6*e^2*f^2*g^2 - 9*a
^2*c^6*d^2*f^2*h^2 - 5*a^2*b^3*c^3*e^3*k^3 - 3*a^3*b^2*c^3*f^2*j^4 - 3*a^2*
b^2*c^4*f^4*j^2 + a^2*b^3*c^3*f^3*j^3 - 18*a^2*b^2*c^4*d^3*k^3 + 12*a^3*b^2
*c^3*d^2*k^4 + 4*a^2*b^2*c^4*e^3*j^3 - 3*a^2*b^4*c^2*d^2*k^4 - 3*a^2*b^2*c^
4*e^2*h^4 + 6*a^7*c*k*l^4*m - 3*a^7*b*k*l*m^4 - 6*a^7*c*h*k*m^4 - 6*a^7*c*g
*l*m^4 + 3*a^6*b*c*h*l^5 - 6*a*c^7*d^4*e*j - 6*a*c^7*d^4*f*h - 3*b*c^7*d^4*
e*f + 6*a*c^7*d*e^4*f + 3*a*b*c^6*e^5*h - a^5*b^2*c*j^3*l^3 - a^3*b^4*c*g^3
*l^3 - a*b^4*c^3*e^3*j^3 - a*b^2*c^5*e^3*g^3 + 3*a^7*b*j*m^5 + 6*a^7*c*f*m^
5 + 6*a*c^7*d^5*k + 3*b*c^7*d^5*g - 3*a^6*c^2*j^4*m^2 - 3*a^6*b^2*j^2*m^4 +
 2*a^6*c^2*j^3*l^3 + a^5*b^3*j^3*m^3 - 2*a^6*c^2*h^3*m^3 - 3*a^6*c^2*h^2*l^
4 - 3*a^5*c^3*h^4*l^2 - a*b^6*c*e^3*l^3 + 20*a^5*c^3*f^3*m^3 - 15*a^6*c^2*f
^2*m^4 - 15*a^4*c^4*f^4*m^2 + 2*a^5*c^3*h^3*k^3 - 2*a^5*c^3*g^3*l^3 + a^3*b
```

^5*g^3*m^3 - 3*a^5*c^3*g^2*k^4 - 3*a^4*c^4*g^4*k^2 - 3*a^4*b^4*f^2*m^4 + 20
*a^4*c^4*e^3*l^3 - 15*a^5*c^3*e^2*l^4 - 15*a^3*c^5*e^4*l^2 + 2*a^4*c^4*g^3*
j^3 - 2*a^4*c^4*f^3*k^3 - 2*a^4*c^4*d^3*m^3 - 3*b^4*c^4*d^4*k^2 - 3*a^4*c^4
*f^2*j^4 - 3*a^3*c^5*f^4*j^2 + 20*a^3*c^5*d^3*k^3 - 15*a^4*c^4*d^2*k^4 - 15
*a^2*c^6*d^4*k^2 - 2*a^3*c^5*e^3*j^3 + b^5*c^3*d^3*j^3 + 2*a^3*c^5*f^3*h^3
- 3*a^3*c^5*e^2*h^4 - 3*a^2*c^6*e^4*h^2 - 3*b^2*c^6*d^4*g^2 + 2*a^2*c^6*e^3
*g^3 - 2*a^2*c^6*d^3*h^3 + b^3*c^5*d^3*g^3 - 3*a^2*c^6*d^2*g^4 - a^4*b^2*c^
2*h^3*k^3 - a^3*b^2*c^3*g^3*j^3 - a^2*b^4*c^2*f^3*k^3 - a^2*b^2*c^4*f^3*h^3
+ 2*a^7*c*k^3*m^3 + a^7*b*l^3*m^3 - 3*a^7*c*j^2*m^4 + 6*a^3*c^5*f^5*m - 3*
a^6*b^2*f*m^5 + 6*a^6*c^2*e*l^5 + 6*a^2*c^6*e^5*l + b^7*c*d^3*l^3 + a*b^7*e
^3*m^3 - 3*b^2*c^6*d^5*k + 6*a^5*c^3*d*k^5 - 3*a*c^7*d^4*g^2 + 2*a*c^7*d^3*
f^3 + b*c^7*d^3*e^3 - a^6*b^2*k^3*m^3 - a^4*b^4*h^3*m^3 - a^2*b^6*f^3*m^3 -
b^6*c^2*d^3*k^3 - b^4*c^4*d^3*h^3 - b^2*c^6*d^3*f^3 - b^8*d^3*m^3 - a^6*c^
2*k^6 - a^5*c^3*j^6 - a^4*c^4*h^6 - a^3*c^5*g^6 - a^2*c^6*f^6 - a^7*c*l^6 -
a*c^7*e^6 - a^8*m^6 - c^8*d^6, z, k1)*((1296*a^3*c^8*f - 1296*a^4*c^7*m -
648*a^2*b^2*c^7*f + 1944*a^2*b^3*c^6*j - 2025*a^2*b^4*c^5*m + 4536*a^3*b^2*
c^6*m + 81*a*b^4*c^6*f - 243*a*b^5*c^5*j - 3888*a^3*b*c^7*j + 243*a*b^6*c^4
*m)/c^3 + (root(34992*a^4*b^2*c^8*z^6 - 8748*a^3*b^4*c^7*z^6 + 729*a^2*b^6*
c^6*z^6 - 46656*a^5*c^9*z^6 + 34992*a^4*b^3*c^6*m*z^5 - 8748*a^3*b^5*c^5*m*
z^5 + 729*a^2*b^7*c^4*m*z^5 - 34992*a^4*b^2*c^7*j*z^5 + 8748*a^3*b^4*c^6*j*
z^5 - 729*a^2*b^6*c^5*j*z^5 - 46656*a^5*b*c^7*m*z^5 + 46656*a^5*c^8*j*z^5 +
34992*a^5*b*c^6*j*m*z^4 - 11664*a^5*b*c^6*k*l*z^4 + 3888*a^4*b*c^7*f*j*z^4
+ 3888*a^4*b*c^7*e*k*z^4 + 3888*a^4*b*c^7*d*l*z^4 + 3888*a^4*b*c^7*g*h*z^4
+ 3888*a^3*b*c^8*d*e*z^4 + 243*a*b^5*c^6*d*e*z^4 - 25272*a^4*b^3*c^5*j*m*z
^4 + 9720*a^4*b^3*c^5*k*l*z^4 + 6075*a^3*b^5*c^4*j*m*z^4 - 2673*a^3*b^5*c^4
*k*l*z^4 - 486*a^2*b^7*c^3*j*m*z^4 + 243*a^2*b^7*c^3*k*l*z^4 - 7776*a^4*b^2
*c^6*h*k*z^4 - 7776*a^4*b^2*c^6*g*l*z^4 - 7776*a^4*b^2*c^6*f*m*z^4 + 2430*a
^3*b^4*c^5*h*k*z^4 + 2430*a^3*b^4*c^5*g*l*z^4 + 2430*a^3*b^4*c^5*f*m*z^4 -
243*a^2*b^6*c^4*h*k*z^4 - 243*a^2*b^6*c^4*g*l*z^4 - 243*a^2*b^6*c^4*f*m*z^4
- 1944*a^3*b^3*c^6*f*j*z^4 - 1944*a^3*b^3*c^6*e*k*z^4 - 1944*a^3*b^3*c^6*d
*l*z^4 + 243*a^2*b^5*c^5*f*j*z^4 + 243*a^2*b^5*c^5*e*k*z^4 + 243*a^2*b^5*c^
5*d*l*z^4 - 1944*a^3*b^3*c^6*g*h*z^4 + 243*a^2*b^5*c^5*g*h*z^4 + 3888*a^3*b
^2*c^7*e*g*z^4 + 3888*a^3*b^2*c^7*d*h*z^4 - 486*a^2*b^4*c^6*e*g*z^4 - 486*a
^2*b^4*c^6*d*h*z^4 - 1944*a^2*b^3*c^7*d*e*z^4 + 7776*a^5*c^7*h*k*z^4 + 7776
*a^5*c^7*g*l*z^4 + 7776*a^5*c^7*f*m*z^4 - 7776*a^4*c^8*e*g*z^4 - 7776*a^4*c
^8*d*h*z^4 - 13608*a^5*b^2*c^5*m^2*z^4 + 11421*a^4*b^4*c^4*m^2*z^4 - 2916*a
^3*b^6*c^3*m^2*z^4 + 243*a^2*b^8*c^2*m^2*z^4 + 13608*a^4*b^2*c^6*j^2*z^4 -
3159*a^3*b^4*c^5*j^2*z^4 + 243*a^2*b^6*c^4*j^2*z^4 + 1944*a^3*b^2*c^7*f^2*z
^4 - 243*a^2*b^4*c^6*f^2*z^4 - 3888*a^6*c^6*m^2*z^4 - 19440*a^5*c^7*j^2*z^4
- 3888*a^4*c^8*f^2*z^4 + 3078*a^4*b^4*c^3*k*l*m*z^3 - 2592*a^5*b^2*c^4*k*l
*m*z^3 - 891*a^3*b^6*c^2*k*l*m*z^3 - 4536*a^4*b^3*c^4*j*k*l*z^3 + 1053*a^3*
b^5*c^3*j*k*l*z^3 - 81*a^2*b^7*c^2*j*k*l*z^3 - 2592*a^4*b^3*c^4*h*k*m*z^3 -
2592*a^4*b^3*c^4*g*l*m*z^3 + 810*a^3*b^5*c^3*h*k*m*z^3 + 810*a^3*b^5*c^3*g
*l*m*z^3 - 81*a^2*b^7*c^2*h*k*m*z^3 - 81*a^2*b^7*c^2*g*l*m*z^3 + 7776*a^4*b
^2*c^5*f*j*m*z^3 + 3888*a^4*b^2*c^5*h*j*k*z^3 + 3888*a^4*b^2*c^5*g*j*l*z^3

- 3888*a^4*b^2*c^5*f*k*l*z^3 - 2916*a^3*b^4*c^4*f*j*m*z^3 + 1458*a^3*b^4*c^
4*f*k*l*z^3 - 972*a^3*b^4*c^4*h*j*k*z^3 - 972*a^3*b^4*c^4*g*j*l*z^3 - 486*a
^3*b^4*c^4*e*k*m*z^3 - 486*a^3*b^4*c^4*d*l*m*z^3 + 324*a^2*b^6*c^3*f*j*m*z^
3 - 162*a^2*b^6*c^3*f*k*l*z^3 + 81*a^2*b^6*c^3*h*j*k*z^3 + 81*a^2*b^6*c^3*g
*j*l*z^3 + 81*a^2*b^6*c^3*e*k*m*z^3 + 81*a^2*b^6*c^3*d*l*m*z^3 - 486*a^3*b^
4*c^4*g*h*m*z^3 + 81*a^2*b^6*c^3*g*h*m*z^3 + 648*a^3*b^3*c^5*e*j*k*z^3 + 64
8*a^3*b^3*c^5*d*j*l*z^3 - 81*a^2*b^5*c^4*e*j*k*z^3 - 81*a^2*b^5*c^4*d*j*l*z
^3 + 2592*a^3*b^3*c^5*e*g*m*z^3 + 2592*a^3*b^3*c^5*d*h*m*z^3 - 1296*a^3*b^3
*c^5*f*h*k*z^3 - 1296*a^3*b^3*c^5*f*g*l*z^3 - 1296*a^3*b^3*c^5*e*h*l*z^3 +
648*a^3*b^3*c^5*g*h*j*z^3 - 324*a^2*b^5*c^4*e*g*m*z^3 - 324*a^2*b^5*c^4*d*h
*m*z^3 + 162*a^2*b^5*c^4*f*h*k*z^3 + 162*a^2*b^5*c^4*f*g*l*z^3 + 162*a^2*b^
5*c^4*e*h*l*z^3 - 81*a^2*b^5*c^4*g*h*j*z^3 + 5184*a^3*b^2*c^6*d*e*m*z^3 - 2
592*a^3*b^2*c^6*e*g*j*z^3 - 2592*a^3*b^2*c^6*d*h*j*z^3 - 2106*a^2*b^4*c^5*d
*e*m*z^3 + 1296*a^3*b^2*c^6*e*f*k*z^3 + 1296*a^3*b^2*c^6*d*g*k*z^3 + 1296*a
^3*b^2*c^6*d*f*l*z^3 + 324*a^2*b^4*c^5*e*g*j*z^3 + 324*a^2*b^4*c^5*d*h*j*z^
3 - 162*a^2*b^4*c^5*e*f*k*z^3 - 162*a^2*b^4*c^5*d*g*k*z^3 - 162*a^2*b^4*c^5
*d*f*l*z^3 + 1296*a^3*b^2*c^6*f*g*h*z^3 - 162*a^2*b^4*c^5*f*g*h*z^3 + 1944*
a^2*b^3*c^6*d*e*j*z^3 - 1296*a^2*b^2*c^7*d*e*f*z^3 + 81*a^2*b^8*c*k*l*m*z^3
 + 6480*a^5*b*c^5*j*k*l*z^3 + 2592*a^5*b*c^5*h*k*m*z^3 + 2592*a^5*b*c^5*g*l
*m*z^3 - 1296*a^4*b*c^6*e*j*k*z^3 - 1296*a^4*b*c^6*d*j*l*z^3 - 5184*a^4*b*c
^6*e*g*m*z^3 - 5184*a^4*b*c^6*d*h*m*z^3 + 2592*a^4*b*c^6*f*h*k*z^3 + 2592*a
^4*b*c^6*f*g*l*z^3 + 2592*a^4*b*c^6*e*h*l*z^3 - 1296*a^4*b*c^6*g*h*j*z^3 +
243*a*b^6*c^4*d*e*m*z^3 - 3888*a^3*b*c^7*d*e*j*z^3 - 243*a*b^5*c^5*d*e*j*z^
3 + 162*a*b^4*c^6*d*e*f*z^3 - 2592*a^6*c^5*k*l*m*z^3 - 5184*a^5*c^6*h*j*k*z
^3 - 5184*a^5*c^6*g*j*l*z^3 - 5184*a^5*c^6*f*j*m*z^3 + 2592*a^5*c^6*f*k*l*z
^3 + 2592*a^5*c^6*e*k*m*z^3 + 2592*a^5*c^6*d*l*m*z^3 + 2592*a^5*c^6*g*h*m*z
^3 + 5184*a^4*c^7*e*g*j*z^3 + 5184*a^4*c^7*d*h*j*z^3 - 2592*a^4*c^7*e*f*k*z
^3 - 2592*a^4*c^7*d*g*k*z^3 - 2592*a^4*c^7*d*f*l*z^3 - 2592*a^4*c^7*d*e*m*z
^3 - 2592*a^4*c^7*f*g*h*z^3 + 2592*a^3*c^8*d*e*f*z^3 + 6480*a^5*b^2*c^4*j*m
^2*z^3 + 6480*a^4*b^3*c^4*j^2*m*z^3 - 5022*a^4*b^4*c^3*j*m^2*z^3 - 1296*a^3
*b^5*c^3*j^2*m*z^3 + 1134*a^3*b^6*c^2*j*m^2*z^3 + 81*a^2*b^7*c^2*j^2*m*z^3
+ 2592*a^4*b^3*c^4*h*l^2*z^3 - 1944*a^4*b^2*c^5*h^2*l*z^3 - 810*a^3*b^5*c^3
*h*l^2*z^3 + 729*a^3*b^4*c^4*h^2*l*z^3 + 81*a^2*b^7*c^2*h*l^2*z^3 - 81*a^2*
b^6*c^3*h^2*l*z^3 - 5184*a^4*b^3*c^4*f*m^2*z^3 + 1620*a^3*b^5*c^3*f*m^2*z^3
 + 1296*a^3*b^3*c^5*f^2*m*z^3 - 162*a^2*b^7*c^2*f*m^2*z^3 - 162*a^2*b^5*c^4
*f^2*m*z^3 - 1944*a^4*b^2*c^5*g*k^2*z^3 + 729*a^3*b^4*c^4*g*k^2*z^3 - 648*a
^3*b^3*c^5*g^2*k*z^3 - 81*a^2*b^6*c^3*g*k^2*z^3 + 81*a^2*b^5*c^4*g^2*k*z^3
- 1944*a^4*b^2*c^5*e*l^2*z^3 + 729*a^3*b^4*c^4*e*l^2*z^3 + 648*a^3*b^2*c^6*
e^2*l*z^3 - 81*a^2*b^6*c^3*e*l^2*z^3 - 81*a^2*b^4*c^5*e^2*l*z^3 + 1296*a^3*
b^3*c^5*f*j^2*z^3 - 1296*a^3*b^2*c^6*f^2*j*z^3 - 162*a^2*b^5*c^4*f*j^2*z^3
+ 162*a^2*b^4*c^5*f^2*j*z^3 - 648*a^3*b^3*c^5*d*k^2*z^3 + 81*a^2*b^5*c^4*d*
k^2*z^3 + 648*a^3*b^2*c^6*e*h^2*z^3 - 81*a^2*b^4*c^5*e*h^2*z^3 - 648*a^2*b^
2*c^7*d^2*g*z^3 - 10368*a^5*b*c^5*j^2*m*z^3 - 81*a^2*b^8*c*j*m^2*z^3 - 2592
*a^5*b*c^5*h*l^2*z^3 + 5184*a^5*b*c^5*f*m^2*z^3 - 2592*a^4*b*c^6*f^2*m*z^3
+ 1296*a^4*b*c^6*g^2*k*z^3 - 2592*a^4*b*c^6*f*j^2*z^3 + 1296*a^4*b*c^6*d*k^

2*z^3 + 81*a*b^4*c^6*d^2*g*z^3 + 2592*a^6*c^5*j*m^2*z^3 + 1296*a^5*c^6*h^2*
l*z^3 + 1296*a^5*c^6*g*k^2*z^3 + 1296*a^5*c^6*e*l^2*z^3 - 1296*a^4*c^7*e^2*
l*z^3 + 2592*a^4*c^7*f^2*j*z^3 - 2592*a^6*b*c^4*m^3*z^3 - 324*a^3*b^7*c*m^3
*z^3 - 27*a^2*b^8*c*l^3*z^3 - 1296*a^4*c^7*e*h^2*z^3 - 864*a^5*b*c^5*k^3*z^
3 + 1296*a^3*c^8*d^2*g*z^3 + 432*a^4*b*c^6*h^3*z^3 + 27*a*b^4*c^6*e^3*z^3 -
432*a^2*b*c^8*d^3*z^3 + 216*a*b^3*c^7*d^3*z^3 + 1134*a^4*b^5*c^2*m^3*z^3 -
432*a^5*b^3*c^3*m^3*z^3 + 1512*a^5*b^2*c^4*l^3*z^3 - 1107*a^4*b^4*c^3*l^3*
z^3 + 297*a^3*b^6*c^2*l^3*z^3 + 864*a^4*b^3*c^4*k^3*z^3 - 270*a^3*b^5*c^3*k
^3*z^3 + 27*a^2*b^7*c^2*k^3*z^3 - 2592*a^4*b^2*c^5*j^3*z^3 + 486*a^3*b^4*c^
4*j^3*z^3 - 27*a^2*b^6*c^3*j^3*z^3 - 216*a^3*b^3*c^5*h^3*z^3 + 27*a^2*b^5*c
^4*h^3*z^3 + 216*a^3*b^2*c^6*g^3*z^3 - 27*a^2*b^4*c^5*g^3*z^3 - 216*a^2*b^2
*c^7*e^3*z^3 - 432*a^6*c^5*l^3*z^3 + 27*a^2*b^9*m^3*z^3 + 4320*a^5*c^6*j^3*
z^3 - 432*a^4*c^7*g^3*z^3 + 432*a^3*c^8*e^3*z^3 - 27*b^5*c^6*d^3*z^3 + 81*a
^3*b^6*c*j*k*l*m*z^2 - 1296*a^5*b*c^4*h*j*k*m*z^2 - 1296*a^5*b*c^4*g*j*l*m*
z^2 + 1296*a^5*b*c^4*f*k*l*m*z^2 - 81*a^2*b^7*c*f*k*l*m*z^2 + 2592*a^4*b*c^
5*e*g*j*m*z^2 + 2592*a^4*b*c^5*d*h*j*m*z^2 - 1296*a^4*b*c^5*f*h*j*k*z^2 - 1
296*a^4*b*c^5*f*g*j*l*z^2 - 1296*a^4*b*c^5*e*f*k*m*z^2 - 1296*a^4*b*c^5*d*f
*l*m*z^2 - 648*a^4*b*c^5*e*h*j*l*z^2 - 648*a^4*b*c^5*e*g*k*l*z^2 - 648*a^4*
b*c^5*d*h*k*l*z^2 - 648*a^4*b*c^5*d*g*k*m*z^2 - 1296*a^4*b*c^5*f*g*h*m*z^2
- 162*a*b^6*c^3*d*e*j*m*z^2 + 81*a*b^6*c^3*d*e*k*l*z^2 + 1296*a^3*b*c^6*d*e
*f*m*z^2 - 648*a^3*b*c^6*d*f*g*k*z^2 - 648*a^3*b*c^6*d*e*h*k*z^2 - 648*a^3*
b*c^6*d*e*g*l*z^2 - 81*a*b^5*c^4*d*e*h*k*z^2 - 81*a*b^5*c^4*d*e*g*l*z^2 + 8
1*a*b^5*c^4*d*e*f*m*z^2 - 81*a*b^4*c^5*d*e*f*j*z^2 + 81*a*b^4*c^5*d*e*g*h*z
^2 + 648*a^5*b^2*c^3*j*k*l*m*z^2 - 567*a^4*b^4*c^2*j*k*l*m*z^2 - 1944*a^4*b
^3*c^3*f*k*l*m*z^2 + 729*a^3*b^5*c^2*f*k*l*m*z^2 + 648*a^4*b^3*c^3*h*j*k*m*
z^2 + 648*a^4*b^3*c^3*g*j*l*m*z^2 - 81*a^3*b^5*c^2*h*j*k*m*z^2 - 81*a^3*b^5
*c^2*g*j*l*m*z^2 + 1944*a^4*b^2*c^4*f*j*k*l*z^2 - 729*a^3*b^4*c^3*f*j*k*l*z
^2 + 648*a^4*b^2*c^4*e*j*k*m*z^2 + 648*a^4*b^2*c^4*d*j*l*m*z^2 - 81*a^3*b^4
*c^3*e*j*k*m*z^2 - 81*a^3*b^4*c^3*d*j*l*m*z^2 + 81*a^2*b^6*c^2*f*j*k*l*z^2
+ 1296*a^4*b^2*c^4*f*h*k*m*z^2 + 1296*a^4*b^2*c^4*f*g*l*m*z^2 + 648*a^4*b^2
*c^4*g*h*j*m*z^2 - 648*a^3*b^4*c^3*f*h*k*m*z^2 - 648*a^3*b^4*c^3*f*g*l*m*z^
2 - 324*a^4*b^2*c^4*g*h*k*l*z^2 - 324*a^4*b^2*c^4*e*h*l*m*z^2 + 81*a^3*b^4*
c^3*g*h*k*l*z^2 - 81*a^3*b^4*c^3*g*h*j*m*z^2 + 81*a^2*b^6*c^2*f*h*k*m*z^2 +
81*a^2*b^6*c^2*f*g*l*m*z^2 - 1296*a^3*b^3*c^4*e*g*j*m*z^2 - 1296*a^3*b^3*c
^4*d*h*j*m*z^2 + 648*a^3*b^3*c^4*f*h*j*k*z^2 + 648*a^3*b^3*c^4*f*g*j*l*z^2
+ 648*a^3*b^3*c^4*e*f*k*m*z^2 + 648*a^3*b^3*c^4*d*f*l*m*z^2 + 486*a^3*b^3*c
^4*e*g*k*l*z^2 + 486*a^3*b^3*c^4*d*h*k*l*z^2 + 162*a^3*b^3*c^4*e*h*j*l*z^2
+ 162*a^3*b^3*c^4*d*g*k*m*z^2 + 162*a^2*b^5*c^3*e*g*j*m*z^2 + 162*a^2*b^5*c
^3*d*h*j*m*z^2 - 81*a^2*b^5*c^3*f*h*j*k*z^2 - 81*a^2*b^5*c^3*f*g*j*l*z^2 -
81*a^2*b^5*c^3*e*g*k*l*z^2 - 81*a^2*b^5*c^3*e*f*k*m*z^2 - 81*a^2*b^5*c^3*d*
h*k*l*z^2 - 81*a^2*b^5*c^3*d*f*l*m*z^2 + 648*a^3*b^3*c^4*f*g*h*m*z^2 - 81*a
^2*b^5*c^3*f*g*h*m*z^2 - 3240*a^3*b^2*c^5*d*e*j*m*z^2 + 1620*a^3*b^2*c^5*d*
e*k*l*z^2 + 1377*a^2*b^4*c^4*d*e*j*m*z^2 - 648*a^3*b^2*c^5*e*f*j*k*z^2 - 64
8*a^3*b^2*c^5*d*f*j*l*z^2 - 648*a^2*b^4*c^4*d*e*k*l*z^2 - 324*a^3*b^2*c^5*d
*g*j*k*z^2 + 81*a^2*b^4*c^4*e*f*j*k*z^2 + 81*a^2*b^4*c^4*d*f*j*l*z^2 + 972*

a^3*b^2*c^5*e*f*h*l*z^2 - 648*a^3*b^2*c^5*f*g*h*j*z^2 - 324*a^3*b^2*c^5*e*g
*h*k*z^2 - 324*a^3*b^2*c^5*d*g*h*l*z^2 - 162*a^2*b^4*c^4*e*f*h*l*z^2 + 81*a
^2*b^4*c^4*f*g*h*j*z^2 + 81*a^2*b^4*c^4*e*g*h*k*z^2 + 81*a^2*b^4*c^4*d*g*h*
l*z^2 - 648*a^2*b^3*c^5*d*e*f*m*z^2 + 486*a^2*b^3*c^5*d*e*h*k*z^2 + 486*a^2
*b^3*c^5*d*e*g*l*z^2 + 162*a^2*b^3*c^5*d*f*g*k*z^2 + 648*a^2*b^2*c^6*d*e*f*
j*z^2 - 324*a^2*b^2*c^6*d*e*g*h*z^2 - 1296*a^6*b*c^3*k*l*m^2*z^2 - 81*a^4*b
^5*c*k*l*m^2*z^2 - 1296*a^5*b*c^4*j^2*k*l*z^2 - 324*a^5*b*c^4*h^2*l*m*z^2 +
 324*a^5*b*c^4*h*k^2*l*z^2 - 324*a^5*b*c^4*g*k^2*m*z^2 + 972*a^5*b*c^4*h*j*
l^2*z^2 + 324*a^5*b*c^4*g*k*l^2*z^2 - 324*a^5*b*c^4*e*l^2*m*z^2 - 324*a^4*b
*c^5*e^2*l*m*z^2 - 1944*a^5*b*c^4*f*j*m^2*z^2 + 1296*a^5*b*c^4*e*k*m^2*z^2
+ 1296*a^5*b*c^4*d*l*m^2*z^2 + 648*a^4*b*c^5*f^2*j*m*z^2 + 81*a^2*b^7*c*f*j
*m^2*z^2 + 1296*a^5*b*c^4*g*h*m^2*z^2 - 324*a^4*b*c^5*g^2*j*k*z^2 + 324*a^4
*b*c^5*g^2*h*l*z^2 + 972*a^4*b*c^5*f*h^2*l*z^2 + 324*a^4*b*c^5*g*h^2*k*z^2
- 324*a^4*b*c^5*e*h^2*m*z^2 - 324*a^4*b*c^5*d*j*k^2*z^2 - 324*a^3*b*c^6*d^2
*j*k*z^2 + 972*a^4*b*c^5*f*g*k^2*z^2 + 972*a^3*b*c^6*d^2*g*m*z^2 + 324*a^4*
b*c^5*e*h*k^2*z^2 + 324*a^3*b*c^6*d^2*h*l*z^2 + 81*a*b^5*c^4*d^2*g*m*z^2 +
972*a^4*b*c^5*e*f*l^2*z^2 + 324*a^4*b*c^5*d*g*l^2*z^2 - 324*a^3*b*c^6*e^2*h
*j*z^2 + 324*a^3*b*c^6*e^2*g*k*z^2 - 324*a^3*b*c^6*e^2*f*l*z^2 - 1296*a^4*b
*c^5*d*e*m^2*z^2 + 81*a*b^7*c^2*d*e*m^2*z^2 - 324*a^3*b*c^6*d*g^2*j*z^2 - 8
1*a*b^4*c^5*d^2*g*j*z^2 + 81*a*b^4*c^5*d^2*e*l*z^2 + 324*a^3*b*c^6*e*g^2*h*
z^2 + 81*a*b^4*c^5*d*e^2*k*z^2 + 1296*a^3*b*c^6*d*e*j^2*z^2 - 324*a^3*b*c^6
*e*f*h^2*z^2 + 324*a^3*b*c^6*d*g*h^2*z^2 + 81*a*b^5*c^4*d*e*j^2*z^2 - 324*a
^2*b*c^7*d^2*f*g*z^2 + 324*a^2*b*c^7*d^2*e*h*z^2 + 81*a*b^3*c^6*d^2*f*g*z^2
 - 81*a*b^3*c^6*d^2*e*h*z^2 + 324*a^2*b*c^7*d*e^2*g*z^2 - 81*a*b^3*c^6*d*e^
2*g*z^2 + 1296*a^6*c^4*j*k*l*m*z^2 - 1296*a^5*c^5*f*j*k*l*z^2 - 1296*a^5*c^
5*e*j*k*m*z^2 - 1296*a^5*c^5*d*j*l*m*z^2 - 1296*a^5*c^5*g*h*j*m*z^2 + 1296*
a^5*c^5*e*h*l*m*z^2 + 1296*a^4*c^6*e*f*j*k*z^2 + 1296*a^4*c^6*d*g*j*k*z^2 +
 1296*a^4*c^6*d*f*j*l*z^2 - 1296*a^4*c^6*d*e*k*l*z^2 + 1296*a^4*c^6*d*e*j*m
*z^2 + 1296*a^4*c^6*f*g*h*j*z^2 - 1296*a^4*c^6*e*f*h*l*z^2 - 1296*a^3*c^7*d
*e*f*j*z^2 + 648*a^5*b^3*c^2*k*l*m^2*z^2 + 648*a^4*b^3*c^3*j^2*k*l*z^2 + 48
6*a^5*b^2*c^3*h*l^2*m*z^2 - 81*a^4*b^4*c^2*h*l^2*m*z^2 + 81*a^4*b^3*c^3*h^2
*l*m*z^2 - 81*a^3*b^5*c^2*j^2*k*l*z^2 - 162*a^4*b^2*c^4*g^2*k*m*z^2 - 81*a^
4*b^3*c^3*h*k^2*l*z^2 + 81*a^4*b^3*c^3*g*k^2*m*z^2 - 567*a^4*b^3*c^3*h*j*l^
2*z^2 + 486*a^4*b^2*c^4*h^2*j*l*z^2 - 81*a^4*b^3*c^3*g*k*l^2*z^2 + 81*a^4*b
^3*c^3*e*l^2*m*z^2 + 81*a^3*b^5*c^2*h*j*l^2*z^2 - 81*a^3*b^4*c^3*h^2*j*l*z^
2 + 81*a^3*b^3*c^4*e^2*l*m*z^2 + 2430*a^4*b^3*c^3*f*j*m^2*z^2 - 2268*a^4*b^
2*c^4*f*j^2*m*z^2 - 810*a^3*b^5*c^2*f*j*m^2*z^2 + 810*a^3*b^4*c^3*f*j^2*m*z
^2 - 648*a^4*b^3*c^3*e*k*m^2*z^2 - 648*a^4*b^3*c^3*d*l*m^2*z^2 - 648*a^4*b^
2*c^4*h*j^2*k*z^2 - 648*a^4*b^2*c^4*g*j^2*l*z^2 - 162*a^3*b^3*c^4*f^2*j*m*z
^2 + 81*a^3*b^5*c^2*e*k*m^2*z^2 + 81*a^3*b^5*c^2*d*l*m^2*z^2 + 81*a^3*b^4*c
^3*h*j^2*k*z^2 + 81*a^3*b^4*c^3*g*j^2*l*z^2 - 81*a^2*b^6*c^2*f*j^2*m*z^2 -
648*a^4*b^3*c^3*g*h*m^2*z^2 + 486*a^4*b^2*c^4*g*j*k^2*z^2 - 486*a^4*b^2*c^4
*e*k^2*l*z^2 + 486*a^3*b^2*c^5*d^2*k*m*z^2 - 162*a^4*b^2*c^4*d*k^2*m*z^2 +
81*a^3*b^5*c^2*g*h*m^2*z^2 - 81*a^3*b^4*c^3*g*j*k^2*z^2 + 81*a^3*b^4*c^3*e*
k^2*l*z^2 + 81*a^3*b^3*c^4*g^2*j*k*z^2 - 81*a^2*b^4*c^4*d^2*k*m*z^2 + 486*a

^4*b^2*c^4*e*j*l^2*z^2 - 486*a^4*b^2*c^4*d*k*l^2*z^2 - 162*a^3*b^2*c^5*e^2*
j*l*z^2 - 81*a^3*b^4*c^3*e*j*l^2*z^2 + 81*a^3*b^4*c^3*d*k*l^2*z^2 - 81*a^3*
b^3*c^4*g^2*h*l*z^2 - 1458*a^4*b^2*c^4*f*h*l^2*z^2 + 648*a^3*b^4*c^3*f*h*l^
2*z^2 - 567*a^3*b^3*c^4*f*h^2*l*z^2 + 486*a^3*b^2*c^5*e^2*h*m*z^2 - 81*a^3*
b^3*c^4*g*h^2*k*z^2 + 81*a^3*b^3*c^4*e*h^2*m*z^2 - 81*a^2*b^6*c^2*f*h*l^2*z
^2 + 81*a^2*b^5*c^3*f*h^2*l*z^2 - 81*a^2*b^4*c^4*e^2*h*m*z^2 - 1296*a^4*b^2
*c^4*e*g*m^2*z^2 - 1296*a^4*b^2*c^4*d*h*m^2*z^2 + 648*a^3*b^4*c^3*e*g*m^2*z
^2 + 648*a^3*b^4*c^3*d*h*m^2*z^2 + 81*a^3*b^3*c^4*d*j*k^2*z^2 - 81*a^2*b^6*
c^2*e*g*m^2*z^2 - 81*a^2*b^6*c^2*d*h*m^2*z^2 + 81*a^2*b^3*c^5*d^2*j*k*z^2 -
 567*a^3*b^3*c^4*f*g*k^2*z^2 - 567*a^2*b^3*c^5*d^2*g*m*z^2 + 486*a^3*b^2*c^
5*f*g^2*k*z^2 - 486*a^3*b^2*c^5*e*g^2*l*z^2 + 486*a^3*b^2*c^5*d*g^2*m*z^2 -
 81*a^3*b^3*c^4*e*h*k^2*z^2 + 81*a^2*b^5*c^3*f*g*k^2*z^2 - 81*a^2*b^4*c^4*f
*g^2*k*z^2 + 81*a^2*b^4*c^4*e*g^2*l*z^2 - 81*a^2*b^4*c^4*d*g^2*m*z^2 - 81*a
^2*b^3*c^5*d^2*h*l*z^2 - 567*a^3*b^3*c^4*e*f*l^2*z^2 - 486*a^3*b^2*c^5*d*h^
2*k*z^2 - 162*a^3*b^2*c^5*e*h^2*j*z^2 - 81*a^3*b^3*c^4*d*g*l^2*z^2 + 81*a^2
*b^5*c^3*e*f*l^2*z^2 + 81*a^2*b^4*c^4*d*h^2*k*z^2 + 81*a^2*b^3*c^5*e^2*h*j*
z^2 - 81*a^2*b^3*c^5*e^2*g*k*z^2 + 81*a^2*b^3*c^5*e^2*f*l*z^2 + 1944*a^3*b^
3*c^4*d*e*m^2*z^2 - 729*a^2*b^5*c^3*d*e*m^2*z^2 + 648*a^3*b^2*c^5*e*g*j^2*z
^2 + 648*a^3*b^2*c^5*d*h*j^2*z^2 - 81*a^2*b^4*c^4*e*g*j^2*z^2 - 81*a^2*b^4*
c^4*d*h*j^2*z^2 + 486*a^3*b^2*c^5*d*f*k^2*z^2 + 486*a^2*b^2*c^6*d^2*g*j*z^2
 - 486*a^2*b^2*c^6*d^2*e*l*z^2 - 162*a^2*b^2*c^6*d^2*f*k*z^2 - 81*a^2*b^4*c
^4*d*f*k^2*z^2 + 81*a^2*b^3*c^5*d*g^2*j*z^2 - 486*a^2*b^2*c^6*d*e^2*k*z^2 -
 81*a^2*b^3*c^5*e*g^2*h*z^2 - 648*a^2*b^3*c^5*d*e*j^2*z^2 - 162*a^2*b^2*c^6
*e^2*f*h*z^2 + 81*a^2*b^3*c^5*e*f*h^2*z^2 - 81*a^2*b^3*c^5*d*g*h^2*z^2 - 16
2*a^2*b^2*c^6*d*f*g^2*z^2 - 189*a^5*b^3*c^2*l^3*m*z^2 + 162*a^5*b^2*c^3*k^3
*m*z^2 - 27*a^4*b^4*c^2*k^3*m*z^2 - 702*a^4*b^3*c^3*j^3*m*z^2 - 81*a^3*b^6*
c*j^2*m^2*z^2 + 81*a^3*b^5*c^2*j^3*m*z^2 - 54*a^5*b^3*c^2*j*m^3*z^2 - 486*a
^5*b^2*c^3*j*l^3*z^2 + 216*a^4*b^4*c^2*j*l^3*z^2 - 189*a^4*b^3*c^3*j*k^3*z^
2 - 54*a^4*b^2*c^4*h^3*m*z^2 + 27*a^3*b^5*c^2*j*k^3*z^2 + 27*a^3*b^3*c^4*g^
3*m*z^2 - 810*a^4*b^4*c^2*f*m^3*z^2 + 540*a^5*b^2*c^3*f*m^3*z^2 - 324*a^3*b
^2*c^5*f^3*m*z^2 + 54*a^2*b^4*c^4*f^3*m*z^2 + 675*a^4*b^3*c^3*f*l^3*z^2 - 2
43*a^3*b^5*c^2*f*l^3*z^2 - 189*a^2*b^3*c^5*e^3*m*z^2 + 27*a^3*b^3*c^4*h^3*j
*z^2 - 486*a^4*b^2*c^4*f*k^3*z^2 - 486*a^2*b^2*c^6*d^3*m*z^2 + 216*a^3*b^4*
c^3*f*k^3*z^2 - 54*a^3*b^2*c^5*g^3*j*z^2 - 27*a^2*b^6*c^2*f*k^3*z^2 - 270*a
^3*b^3*c^4*f*j^3*z^2 - 54*a^2*b^3*c^5*f^3*j*z^2 + 27*a^2*b^5*c^3*f*j^3*z^2
+ 162*a^2*b^2*c^6*e^3*j*z^2 + 162*a^3*b^2*c^5*f*h^3*z^2 - 27*a^2*b^4*c^4*f*
h^3*z^2 + 27*a^2*b^3*c^5*f*g^3*z^2 + 81*a*b^2*c^7*d^2*e^2*z^2 - 648*a^6*c^4
*h*l^2*m*z^2 + 648*a^5*c^5*g^2*k*m*z^2 - 648*a^5*c^5*h^2*j*l*z^2 + 1296*a^5
*c^5*h*j^2*k*z^2 + 1296*a^5*c^5*g*j^2*l*z^2 + 1296*a^5*c^5*f*j^2*m*z^2 - 64
8*a^5*c^5*g*j*k^2*z^2 + 648*a^5*c^5*e*k^2*l*z^2 + 648*a^5*c^5*d*k^2*m*z^2 -
 648*a^4*c^6*d^2*k*m*z^2 - 648*a^5*c^5*e*j*l^2*z^2 + 648*a^5*c^5*d*k*l^2*z^
2 + 648*a^4*c^6*e^2*j*l*z^2 + 324*a^6*b*c^3*l^3*m*z^2 + 27*a^4*b^5*c*l^3*m*
z^2 + 648*a^5*c^5*f*h*l^2*z^2 - 648*a^4*c^6*e^2*h*m*z^2 + 1512*a^5*b*c^4*j^
3*m*z^2 + 1080*a^6*b*c^3*j*m^3*z^2 - 162*a^4*b^5*c*j*m^3*z^2 - 648*a^4*c^6*
f*g^2*k*z^2 + 648*a^4*c^6*e*g^2*l*z^2 - 648*a^4*c^6*d*g^2*m*z^2 - 27*a^3*b^

6*c*j*l^3*z^2 + 648*a^4*c^6*e*h^2*j*z^2 + 648*a^4*c^6*d*h^2*k*z^2 + 324*a^5
*b*c^4*j*k^3*z^2 - 1296*a^4*c^6*e*g*j^2*z^2 - 1296*a^4*c^6*d*h*j^2*z^2 - 10
8*a^4*b*c^5*g^3*m*z^2 - 648*a^4*c^6*d*f*k^2*z^2 - 648*a^3*c^7*d^2*g*j*z^2 +
 648*a^3*c^7*d^2*f*k*z^2 + 648*a^3*c^7*d^2*e*l*z^2 + 270*a^3*b^6*c*f*m^3*z^
2 + 648*a^3*c^7*d*e^2*k*z^2 - 540*a^5*b*c^4*f*l^3*z^2 + 324*a^3*b*c^6*e^3*m
*z^2 - 108*a^4*b*c^5*h^3*j*z^2 + 27*a^2*b^7*c*f*l^3*z^2 + 27*a*b^5*c^4*e^3*
m*z^2 + 648*a^3*c^7*e^2*f*h*z^2 + 216*a*b^4*c^5*d^3*m*z^2 + 648*a^4*b*c^5*f
*j^3*z^2 + 216*a^3*b*c^6*f^3*j*z^2 + 648*a^3*c^7*d*f*g^2*z^2 - 27*a*b^4*c^5
*e^3*j*z^2 + 324*a^2*b*c^7*d^3*j*z^2 - 189*a*b^3*c^6*d^3*j*z^2 - 108*a^3*b*
c^6*f*g^3*z^2 - 108*a^2*b*c^7*e^3*f*z^2 + 27*a*b^3*c^6*e^3*f*z^2 + 162*a*b^
2*c^7*d^3*f*z^2 - 1134*a^5*b^2*c^3*j^2*m^2*z^2 + 648*a^4*b^4*c^2*j^2*m^2*z^
2 + 81*a^5*b^2*c^3*k^2*l^2*z^2 + 162*a^4*b^2*c^4*f^2*m^2*z^2 + 81*a^4*b^2*c
^4*h^2*k^2*z^2 + 81*a^4*b^2*c^4*g^2*l^2*z^2 + 162*a^3*b^2*c^5*f^2*j^2*z^2 +
 81*a^3*b^2*c^5*e^2*k^2*z^2 + 81*a^3*b^2*c^5*d^2*l^2*z^2 + 81*a^3*b^2*c^5*g
^2*h^2*z^2 + 81*a^2*b^2*c^6*e^2*g^2*z^2 + 81*a^2*b^2*c^6*d^2*h^2*z^2 - 216*
a^6*c^4*k^3*m*z^2 + 216*a^6*c^4*j*l^3*z^2 + 27*a^3*b^7*j*m^3*z^2 + 216*a^5*
c^5*h^3*m*z^2 + 432*a^6*c^4*f*m^3*z^2 + 432*a^4*c^6*f^3*m*z^2 - 27*b^6*c^4*
d^3*m*z^2 - 27*a^2*b^8*f*m^3*z^2 + 216*a^5*c^5*f*k^3*z^2 + 216*a^4*c^6*g^3*
j*z^2 + 216*a^3*c^7*d^3*m*z^2 + 216*a^5*b^4*c*m^4*z^2 - 216*a^3*c^7*e^3*j*z
^2 + 27*b^5*c^5*d^3*j*z^2 - 216*a^4*c^6*f*h^3*z^2 - 27*b^4*c^6*d^3*f*z^2 -
216*a^2*c^8*d^3*f*z^2 - 648*a^6*c^4*j^2*m^2*z^2 - 324*a^6*c^4*k^2*l^2*z^2 -
 648*a^5*c^5*f^2*m^2*z^2 - 324*a^5*c^5*h^2*k^2*z^2 - 324*a^5*c^5*g^2*l^2*z^
2 - 648*a^4*c^6*f^2*j^2*z^2 - 324*a^4*c^6*e^2*k^2*z^2 - 324*a^4*c^6*d^2*l^2
*z^2 - 405*a^6*b^2*c^2*m^4*z^2 - 324*a^4*c^6*g^2*h^2*z^2 - 324*a^3*c^7*e^2*
g^2*z^2 - 324*a^3*c^7*d^2*h^2*z^2 + 243*a^4*b^2*c^4*j^4*z^2 - 27*a^3*b^4*c^
3*j^4*z^2 - 324*a^2*c^8*d^2*e^2*z^2 + 27*a^2*b^2*c^6*f^4*z^2 - 108*a^7*c^3*
m^4*z^2 - 27*a^4*b^6*m^4*z^2 - 540*a^5*c^5*j^4*z^2 - 108*a^3*c^7*f^4*z^2 -
216*a^5*b*c^3*f*j*k*l*m*z - 54*a^3*b^5*c*f*j*k*l*m*z + 27*a^3*b^5*c*g*h*k*l
*m*z - 27*a^2*b^6*c*e*g*k*l*m*z - 27*a^2*b^6*c*d*h*k*l*m*z + 432*a^4*b*c^4*
d*g*j*k*m*z - 432*a^4*b*c^4*d*e*k*l*m*z + 216*a^4*b*c^4*e*g*j*k*l*z + 216*a
^4*b*c^4*e*f*j*k*m*z + 216*a^4*b*c^4*d*h*j*k*l*z + 216*a^4*b*c^4*d*f*j*l*m*
z + 216*a^4*b*c^4*f*g*h*j*m*z - 27*a*b^6*c^2*d*e*j*k*l*z - 27*a*b^6*c^2*d*e
*h*k*m*z - 27*a*b^6*c^2*d*e*g*l*m*z + 216*a^3*b*c^5*d*e*h*j*k*z + 216*a^3*b
*c^5*d*e*g*j*l*z - 216*a^3*b*c^5*d*e*f*j*m*z + 27*a*b^5*c^3*d*e*h*j*k*z + 2
7*a*b^5*c^3*d*e*g*j*l*z + 27*a*b^5*c^3*d*e*g*h*m*z - 27*a*b^4*c^4*d*e*g*h*j
*z + 27*a*b^7*c*d*e*k*l*m*z + 270*a^4*b^3*c^2*f*j*k*l*m*z - 108*a^4*b^3*c^2
*g*h*k*l*m*z - 216*a^4*b^2*c^3*f*h*j*k*m*z - 216*a^4*b^2*c^3*f*g*j*l*m*z -
216*a^4*b^2*c^3*e*g*k*l*m*z - 216*a^4*b^2*c^3*d*h*k*l*m*z + 162*a^3*b^4*c^2
*e*g*k*l*m*z + 162*a^3*b^4*c^2*d*h*k*l*m*z + 108*a^4*b^2*c^3*g*h*j*k*l*z +
108*a^4*b^2*c^3*e*h*j*l*m*z + 54*a^3*b^4*c^2*f*h*j*k*m*z + 54*a^3*b^4*c^2*f
*g*j*l*m*z - 27*a^3*b^4*c^2*g*h*j*k*l*z + 540*a^3*b^3*c^3*d*e*k*l*m*z - 216
*a^2*b^5*c^2*d*e*k*l*m*z - 162*a^3*b^3*c^3*e*g*j*k*l*z - 162*a^3*b^3*c^3*d*
h*j*k*l*z - 108*a^3*b^3*c^3*d*g*j*k*m*z - 54*a^3*b^3*c^3*e*f*j*k*m*z - 54*a
^3*b^3*c^3*d*f*j*l*m*z + 27*a^2*b^5*c^2*e*g*j*k*l*z + 27*a^2*b^5*c^2*d*h*j*
k*l*z - 108*a^3*b^3*c^3*e*g*h*k*m*z - 108*a^3*b^3*c^3*d*g*h*l*m*z - 54*a^3*

b^3*c^3*f*g*h*j*m*z + 27*a^2*b^5*c^2*e*g*h*k*m*z + 27*a^2*b^5*c^2*d*g*h*l*m
*z - 540*a^3*b^2*c^4*d*e*j*k*l*z + 216*a^2*b^4*c^3*d*e*j*k*l*z - 216*a^3*b^
2*c^4*d*e*h*k*m*z - 216*a^3*b^2*c^4*d*e*g*l*m*z + 162*a^2*b^4*c^3*d*e*h*k*m
*z + 162*a^2*b^4*c^3*d*e*g*l*m*z + 108*a^3*b^2*c^4*e*g*h*j*k*z - 108*a^3*b^
2*c^4*e*f*h*j*l*z + 108*a^3*b^2*c^4*d*g*h*j*l*z + 108*a^3*b^2*c^4*d*f*g*k*m
*z - 27*a^2*b^4*c^3*e*g*h*j*k*z - 27*a^2*b^4*c^3*d*g*h*j*l*z - 162*a^2*b^3*
c^4*d*e*h*j*k*z - 162*a^2*b^3*c^4*d*e*g*j*l*z + 54*a^2*b^3*c^4*d*e*f*j*m*z
- 108*a^2*b^3*c^4*d*e*g*h*m*z + 108*a^2*b^2*c^5*d*e*g*h*j*z + 324*a^6*b*c^2
*j*k*l*m^2*z - 81*a^5*b^3*c*j*k*l*m^2*z + 27*a^4*b^4*c*j^2*k*l*m*z - 27*a^4
*b^4*c*h*k^2*l*m*z - 27*a^4*b^4*c*g*k*l^2*m*z + 216*a^5*b*c^3*h*j^2*k*m*z +
216*a^5*b*c^3*g*j^2*l*m*z + 54*a^4*b^4*c*f*k*l*m^2*z + 27*a^4*b^4*c*h*j*k*
m^2*z + 27*a^4*b^4*c*g*j*l*m^2*z + 27*a^2*b^6*c*f^2*k*l*m*z + 216*a^5*b*c^3
*e*k^2*l*m*z - 108*a^5*b*c^3*h*j*k^2*l*z + 27*a^3*b^5*c*e*k^2*l*m*z + 216*a
^5*b*c^3*d*k*l^2*m*z + 216*a^4*b*c^4*e^2*j*l*m*z - 108*a^5*b*c^3*g*j*k*l^2*
z + 27*a^3*b^5*c*d*k*l^2*m*z - 324*a^5*b*c^3*e*j*k*m^2*z - 324*a^5*b*c^3*d*
j*l*m^2*z - 216*a^5*b*c^3*f*h*l^2*m*z - 108*a^4*b*c^4*f^2*j*k*l*z - 27*a^3*
b^5*c*e*j*k*m^2*z - 27*a^3*b^5*c*d*j*l*m^2*z - 324*a^5*b*c^3*g*h*j*m^2*z +
216*a^5*b*c^3*f*h*k*m^2*z + 216*a^5*b*c^3*f*g*l*m^2*z + 216*a^5*b*c^3*e*h*l
*m^2*z - 216*a^4*b*c^4*f^2*h*k*m*z - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5
*c*g*h*j*m^2*z + 216*a^4*b*c^4*e*g^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 21
6*a^4*b*c^4*f*h^2*j*l*z + 216*a^4*b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k
*m*z - 108*a^4*b*c^4*g*h^2*j*k*z - 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^
4*d*h*j^2*m*z + 216*a^4*b*c^4*f*h*j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*
a^2*b^6*c*e*g*j*m^2*z + 27*a^2*b^6*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*
z - 216*a^4*b*c^4*f*g*j*k^2*z + 216*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d
^2*e*l*m*z - 108*a^4*b*c^4*e*h*j*k^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^
3*b*c^5*d^2*h*j*l*z + 108*a^3*b*c^5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z
+ 27*a*b^5*c^3*d^2*g*k*l*z + 27*a*b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j
*l^2*z + 216*a^3*b*c^5*d*e^2*k*m*z - 108*a^4*b*c^4*d*g*j*l^2*z - 108*a^3*b*
c^5*e^2*g*j*k*z + 27*a*b^5*c^3*d*e^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^2*z + 21
6*a^3*b*c^5*e^2*f*h*m*z - 108*a^4*b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5*e^2*g*h
*l*z + 108*a^3*b*c^5*e*f^2*j*k*z + 108*a^3*b*c^5*d*f^2*j*l*z + 27*a*b^6*c^2
*d*e*j^2*m*z - 216*a^3*b*c^5*e*f^2*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*z - 27*a
*b^4*c^4*d^2*e*j*l*z + 216*a^3*b*c^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*g^2*h*j*
z + 54*a*b^4*c^4*d^2*f*g*m*z - 27*a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*c^4*d^2*
e*h*m*z - 27*a*b^4*c^4*d*e^2*j*k*z - 108*a^3*b*c^5*d*g*h^2*j*z + 54*a*b^4*c
^4*d*e^2*h*l*z + 27*a*b^6*c^2*d*e*h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*z - 27*a
*b^4*c^4*d*e^2*g*m*z - 27*a*b^4*c^4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2*f*g*j*z
- 108*a^3*b*c^5*d*e*g*k^2*z - 108*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*b*c^6*d^
2*e*g*k*z - 54*a*b^3*c^5*d^2*f*g*j*z - 27*a*b^5*c^3*d*e*g*k^2*z + 27*a*b^4*
c^4*d*e*g^2*k*z + 27*a*b^3*c^5*d^2*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k*z - 108
*a^2*b*c^6*d*e^2*g*j*z + 27*a*b^3*c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d*e*f^2*j
*z + 27*a*b^3*c^5*d*e*f^2*j*z - 432*a^5*c^4*e*h*j*l*m*z + 432*a^4*c^5*d*e*j
*k*l*z + 432*a^4*c^5*e*f*h*j*l*z - 432*a^4*c^5*d*f*g*k*m*z - 27*a*b^7*c*d*e
*j*m^2*z - 54*a^5*b^2*c^2*j^2*k*l*m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z + 108*a

^5*b^2*c^2*g*k*l^2*m*z - 54*a^5*b^2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c^3*f^2*k
*l*m*z - 270*a^5*b^2*c^2*f*k*l*m^2*z - 189*a^3*b^4*c^2*f^2*k*l*m*z - 108*a^
5*b^2*c^2*h*j*k*m^2*z - 108*a^5*b^2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^2*h*j^2*
k*m*z - 54*a^4*b^3*c^2*g*j^2*l*m*z - 162*a^4*b^3*c^2*e*k^2*l*m*z + 54*a^4*b
^2*c^3*g^2*j*k*m*z + 27*a^4*b^3*c^2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d*k*l^2*m
*z + 108*a^4*b^2*c^3*g^2*h*l*m*z - 54*a^3*b^3*c^3*e^2*j*l*m*z + 27*a^4*b^3*
c^2*g*j*k*l^2*z - 27*a^3*b^4*c^2*g^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^2*k*l*z
+ 189*a^4*b^3*c^2*e*j*k*m^2*z + 189*a^4*b^3*c^2*d*j*l*m^2*z - 162*a^4*b^2*c
^3*e*j^2*k*m*z - 162*a^4*b^2*c^3*d*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*j*k*l*z
+ 108*a^4*b^2*c^3*g*h^2*k*m*z + 54*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4*b^2*c^3
*f*h^2*l*m*z + 54*a^3*b^4*c^2*f*j^2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*m*z + 27
*a^3*b^4*c^2*e*j^2*k*m*z + 27*a^3*b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*c^2*f^2*
j*k*l*z - 270*a^3*b^2*c^4*d^2*j*k*m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z - 162*a
^4*b^2*c^3*g*h*j^2*m*z + 162*a^4*b^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*c^3*f^2*
h*k*m*z + 162*a^3*b^3*c^3*f^2*g*l*m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z - 54*a^4
*b^3*c^2*f*g*l*m^2*z - 54*a^4*b^3*c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*d*j*k^2*
m*z + 54*a^2*b^4*c^3*d^2*j*k*m*z + 27*a^3*b^4*c^2*g*h*j^2*m*z - 27*a^3*b^4*
c^2*e*j*k^2*l*z - 27*a^2*b^5*c^2*f^2*h*k*m*z - 27*a^2*b^5*c^2*f^2*g*l*m*z +
162*a^4*b^2*c^3*d*j*k*l^2*z - 162*a^3*b^3*c^3*e*g^2*l*m*z + 108*a^4*b^2*c^
3*e*h*k^2*m*z + 108*a^3*b^2*c^4*d^2*h*l*m*z - 54*a^4*b^2*c^3*f*g*k^2*m*z -
27*a^3*b^4*c^2*e*h*k^2*m*z - 27*a^3*b^4*c^2*d*j*k*l^2*z + 27*a^3*b^3*c^3*g^
2*h*j*l*z + 27*a^2*b^5*c^2*e*g^2*l*m*z - 27*a^2*b^4*c^3*d^2*h*l*m*z + 270*a
^4*b^2*c^3*f*h*j*l^2*z - 270*a^3*b^2*c^4*e^2*h*j*m*z - 162*a^4*b^2*c^3*e*h*
k*l^2*z - 162*a^3*b^3*c^3*d*h^2*k*m*z + 162*a^3*b^2*c^4*e^2*h*k*l*z + 108*a
^4*b^2*c^3*d*g*l^2*m*z + 108*a^3*b^2*c^4*e^2*g*k*m*z - 54*a^4*b^2*c^3*e*f*l
^2*m*z - 54*a^3*b^4*c^2*f*h*j*l^2*z + 54*a^3*b^3*c^3*f*h^2*j*l*z - 54*a^3*b
^3*c^3*e*h^2*j*m*z + 54*a^3*b^2*c^4*e^2*f*l*m*z + 54*a^2*b^4*c^3*e^2*h*j*m*
z + 27*a^3*b^4*c^2*e*h*k*l^2*z - 27*a^3*b^4*c^2*d*g*l^2*m*z + 27*a^3*b^3*c^
3*g*h^2*j*k*z + 27*a^2*b^5*c^2*d*h^2*k*m*z - 27*a^2*b^4*c^3*e^2*h*k*l*z - 2
7*a^2*b^4*c^3*e^2*g*k*m*z + 432*a^4*b^2*c^3*e*g*j*m^2*z + 432*a^4*b^2*c^3*d
*h*j*m^2*z - 270*a^4*b^2*c^3*d*g*k*m^2*z - 216*a^3*b^4*c^2*e*g*j*m^2*z - 21
6*a^3*b^4*c^2*d*h*j*m^2*z + 216*a^3*b^3*c^3*e*g*j^2*m*z + 216*a^3*b^3*c^3*d
*h*j^2*m*z - 162*a^3*b^2*c^4*e*f^2*k*m*z - 162*a^3*b^2*c^4*d*f^2*l*m*z - 10
8*a^3*b^2*c^4*f^2*h*j*k*z - 108*a^3*b^2*c^4*f^2*g*j*l*z + 54*a^4*b^2*c^3*e*
f*k*m^2*z + 54*a^4*b^2*c^3*d*f*l*m^2*z + 54*a^3*b^4*c^2*d*g*k*m^2*z - 54*a^
3*b^3*c^3*f*h*j^2*k*z - 54*a^3*b^3*c^3*f*g*j^2*l*z - 27*a^2*b^5*c^2*e*g*j^2
*m*z - 27*a^2*b^5*c^2*d*h*j^2*m*z + 27*a^2*b^4*c^3*f^2*h*j*k*z + 27*a^2*b^4
*c^3*f^2*g*j*l*z + 27*a^2*b^4*c^3*e*f^2*k*m*z + 27*a^2*b^4*c^3*d*f^2*l*m*z
+ 324*a^2*b^3*c^4*d^2*g*j*m*z - 270*a^3*b^2*c^4*d*g^2*j*m*z - 162*a^3*b^2*c
^4*f^2*g*h*m*z + 162*a^3*b^2*c^4*e*g^2*j*l*z - 162*a^2*b^3*c^4*d^2*e*l*m*z
- 135*a^2*b^3*c^4*d^2*g*k*l*z + 108*a^3*b^2*c^4*d*g^2*k*l*z + 54*a^4*b^2*c^
3*f*g*h*m^2*z + 54*a^3*b^3*c^3*f*g*j*k^2*z - 54*a^3*b^2*c^4*f*g^2*j*k*z + 5
4*a^2*b^4*c^3*d*g^2*j*m*z - 54*a^2*b^3*c^4*d^2*f*k*m*z + 27*a^3*b^3*c^3*e*h
*j*k^2*z + 27*a^3*b^3*c^3*d*g*k^2*l*z + 27*a^2*b^4*c^3*f^2*g*h*m*z - 27*a^2
*b^4*c^3*e*g^2*j*l*z - 27*a^2*b^4*c^3*d*g^2*k*l*z + 27*a^2*b^3*c^4*d^2*h*j*

l*z + 162*a^3*b^2*c^4*d*h^2*j*k*z - 162*a^2*b^3*c^4*d*e^2*k*m*z + 108*a^3*b
^2*c^4*e*g^2*h*m*z + 54*a^3*b^3*c^3*e*f*j*l^2*z + 27*a^3*b^3*c^3*d*g*j*l^2*
z - 27*a^2*b^4*c^3*e*g^2*h*m*z - 27*a^2*b^4*c^3*d*h^2*j*k*z + 27*a^2*b^3*c^
4*e^2*g*j*k*z - 621*a^3*b^3*c^3*d*e*j*m^2*z + 594*a^3*b^2*c^4*d*e*j^2*m*z +
243*a^2*b^5*c^2*d*e*j*m^2*z - 243*a^2*b^4*c^3*d*e*j^2*m*z + 135*a^3*b^3*c^
3*e*g*h*l^2*z - 108*a^3*b^2*c^4*e*g*h^2*l*z + 108*a^3*b^2*c^4*d*g*h^2*m*z +
54*a^3*b^2*c^4*e*f*j^2*k*z + 54*a^3*b^2*c^4*e*f*h^2*m*z + 54*a^3*b^2*c^4*d
*g*j^2*k*z + 54*a^3*b^2*c^4*d*f*j^2*l*z - 54*a^2*b^3*c^4*e^2*f*h*m*z - 27*a
^2*b^5*c^2*e*g*h*l^2*z + 27*a^2*b^4*c^3*e*g*h^2*l*z - 27*a^2*b^4*c^3*d*g*h^
2*m*z - 27*a^2*b^3*c^4*e^2*g*h*l*z - 27*a^2*b^3*c^4*e*f^2*j*k*z - 27*a^2*b^
3*c^4*d*f^2*j*l*z + 162*a^2*b^2*c^5*d^2*e*j*l*z + 54*a^3*b^2*c^4*f*g*h*j^2*
z - 54*a^3*b^2*c^4*d*f*j*k^2*z + 54*a^2*b^3*c^4*e*f^2*h*l*z + 54*a^2*b^2*c^
5*d^2*f*j*k*z - 27*a^2*b^3*c^4*f^2*g*h*j*z - 270*a^2*b^2*c^5*d^2*f*g*m*z -
162*a^3*b^2*c^4*d*g*h*k^2*z + 162*a^2*b^2*c^5*d^2*g*h*k*z + 162*a^2*b^2*c^5
*d*e^2*j*k*z + 108*a^2*b^2*c^5*d^2*e*h*m*z - 54*a^2*b^3*c^4*d*f*g^2*m*z + 2
7*a^2*b^4*c^3*d*g*h*k^2*z + 27*a^2*b^3*c^4*e*g^2*h*j*z + 270*a^3*b^2*c^4*d*
e*h*l^2*z - 270*a^2*b^2*c^5*d*e^2*h*l*z - 162*a^2*b^4*c^3*d*e*h*l^2*z + 108
*a^2*b^3*c^4*d*e*h^2*l*z + 108*a^2*b^2*c^5*d*e^2*g*m*z + 54*a^2*b^2*c^5*e^2
*f*h*j*z + 27*a^2*b^3*c^4*d*g*h^2*j*z + 162*a^2*b^2*c^5*d*e*f^2*m*z - 54*a^
3*b^2*c^4*d*e*f*m^2*z - 54*a^2*b^2*c^5*d*f^2*g*k*z + 135*a^2*b^3*c^4*d*e*g*
k^2*z - 108*a^2*b^2*c^5*d*e*g^2*k*z + 54*a^2*b^2*c^5*d*f*g^2*j*z - 54*a^2*b
^2*c^5*d*e*f*j^2*z - 9*a*b^7*c*d*e*l^3*z - 36*a*b*c^7*d^3*e*g*z - 108*a^6*b
*c^2*k^2*l^2*m*z + 27*a^5*b^3*c*k^2*l^2*m*z - 18*a^5*b^2*c^2*j*k^3*m*z - 27
*a^4*b^3*c^2*j^3*k*l*z - 108*a^5*b*c^3*h^2*k^2*m*z - 108*a^5*b*c^3*g^2*l^2*
m*z + 108*a^5*b*c^3*h^2*k*l^2*z + 108*a^5*b*c^3*g^2*k*m^2*z + 90*a^5*b^2*c^
2*f*l^3*m*z - 18*a^5*b^2*c^2*h*k*l^3*z + 18*a^4*b^2*c^3*h^3*k*l*z + 18*a^4*
b^2*c^3*h^3*j*m*z - 108*a^5*b*c^3*h*j^2*l^2*z + 18*a^4*b^3*c^2*f*k^3*m*z -
18*a^3*b^3*c^3*g^3*j*m*z - 9*a^4*b^3*c^2*g*k^3*l*z + 9*a^3*b^3*c^3*g^3*k*l*
z + 252*a^4*b^2*c^3*f*j^3*m*z + 216*a^5*b*c^3*f*j^2*m^2*z + 180*a^3*b^2*c^4
*f^3*j*m*z - 108*a^4*b*c^4*e^2*k^2*m*z - 108*a^4*b*c^4*d^2*l^2*m*z + 90*a^5
*b^2*c^2*e*k*m^3*z + 90*a^5*b^2*c^2*d*l*m^3*z - 90*a^3*b^2*c^4*f^3*k*l*z +
54*a^3*b^5*c*f*j^2*m^2*z - 54*a^3*b^4*c^2*f*j^3*m*z + 36*a^5*b^2*c^2*f*j*m^
3*z + 36*a^4*b^2*c^3*h*j^3*k*z + 36*a^4*b^2*c^3*g*j^3*l*z - 36*a^2*b^4*c^3*
f^3*j*m*z - 27*a^2*b^6*c*f^2*j*m^2*z + 18*a^2*b^4*c^3*f^3*k*l*z - 216*a^4*b
*c^4*d^2*k*m^2*z + 108*a^5*b*c^3*d*k^2*m^2*z - 108*a^4*b^3*c^2*f*j*l^3*z -
108*a^4*b*c^4*g^2*h^2*m*z + 108*a^2*b^3*c^4*e^3*j*m*z + 90*a^5*b^2*c^2*g*h*
m^3*z + 54*a^4*b^3*c^2*e*k*l^3*z - 54*a^2*b^3*c^4*e^3*k*l*z + 234*a^2*b^2*c
^5*d^3*j*m*z - 144*a^2*b^2*c^5*d^3*k*l*z + 90*a^4*b^2*c^3*f*j*k^3*z - 72*a^
4*b^2*c^3*d*k^3*l*z + 27*a^4*b^3*c^2*g*h*l^3*z - 27*a^3*b^3*c^3*g*h^3*l*z -
18*a^3*b^4*c^2*f*j*k^3*z + 9*a^3*b^4*c^2*d*k^3*l*z + 216*a^4*b*c^4*f^2*h*l
^2*z - 216*a^4*b*c^4*e^2*h*m^2*z + 108*a^4*b*c^4*g^2*h*k^2*z - 18*a^4*b^2*c
^3*g*h*k^3*z + 18*a^3*b^2*c^4*g^3*h*k*z + 18*a^3*b^2*c^4*f*g^3*m*z + 9*a^3*
b^4*c^2*g*h*k^3*z - 9*a^3*b^3*c^3*e*j^3*k*z - 9*a^3*b^3*c^3*d*j^3*l*z - 144
*a^4*b^3*c^2*e*g*m^3*z - 144*a^4*b^3*c^2*d*h*m^3*z - 108*a^3*b*c^5*e^2*g^2*
m*z + 108*a^3*b*c^5*d^2*j^2*k*z - 108*a^3*b*c^5*d^2*h^2*m*z - 18*a^2*b^3*c^

4*f^3*h*k*z - 18*a^2*b^3*c^4*f^3*g*l*z - 9*a^3*b^3*c^3*g*h*j^3*z - 216*a^4*
b*c^4*d*g^2*m^2*z + 144*a^4*b^2*c^3*e*g*l^3*z - 126*a^3*b^2*c^4*d*h^3*l*z -
108*a^4*b*c^4*d*h^2*l^2*z - 108*a^3*b*c^5*f^2*g^2*k*z - 108*a^3*b*c^5*e^2*
h^2*k*z - 90*a^2*b^2*c^5*e^3*f*m*z + 72*a^2*b^2*c^5*e^3*g*l*z - 63*a^3*b^4*
c^2*e*g*l^3*z - 36*a^3*b^4*c^2*d*h*l^3*z + 27*a^2*b^4*c^3*d*h^3*l*z + 27*a*
b^6*c^2*d^2*g*m^2*z - 18*a^4*b^2*c^3*d*h*l^3*z - 18*a^3*b^2*c^4*f*h^3*j*z -
18*a^3*b^2*c^4*e*h^3*k*z + 18*a^2*b^2*c^5*e^3*h*k*z + 108*a^3*b*c^5*e^2*h*
j^2*z + 54*a^3*b^3*c^3*d*h*k^3*z + 27*a^3*b^3*c^3*e*g*k^3*z - 27*a^2*b^3*c^
4*e*g^3*k*z + 27*a^2*b^3*c^4*d*g^3*l*z - 27*a*b^4*c^4*d^2*g^2*l*z - 9*a^2*b
^5*c^2*e*g*k^3*z - 9*a^2*b^5*c^2*d*h*k^3*z + 207*a^3*b^4*c^2*d*e*m^3*z - 10
8*a^2*b*c^6*d^2*e^2*m*z - 90*a^4*b^2*c^3*d*e*m^3*z - 72*a^3*b^2*c^4*e*g*j^3
*z - 72*a^3*b^2*c^4*d*h*j^3*z + 27*a*b^3*c^5*d^2*e^2*m*z + 18*a^2*b^2*c^5*e
*f^3*k*z + 18*a^2*b^2*c^5*d*f^3*l*z + 9*a^2*b^4*c^3*e*g*j^3*z + 9*a^2*b^4*c
^3*d*h*j^3*z - 216*a^3*b*c^5*d*e^2*l^2*z - 198*a^3*b^3*c^3*d*e*l^3*z + 108*
a^3*b*c^5*d*g^2*j^2*z - 108*a^3*b*c^5*d*f^2*k^2*z + 72*a^2*b^5*c^2*d*e*l^3*
z - 27*a*b^5*c^3*d*e^2*l^2*z + 27*a*b^4*c^4*d^2*g*j^2*z + 18*a^2*b^2*c^5*f^
3*g*h*z + 144*a^3*b^2*c^4*d*e*k^3*z - 63*a^2*b^4*c^3*d*e*k^3*z + 27*a*b^4*c
^4*d^2*e*k^2*z - 9*a^2*b^3*c^4*e*g*h^3*z - 108*a^2*b*c^6*d^2*g^2*h*z + 81*a
^2*b^3*c^4*d*e*j^3*z + 27*a*b^3*c^5*d^2*g^2*h*z - 27*a*b^2*c^6*d^2*e^2*j*z
- 18*a^2*b^2*c^5*d*g^3*h*z + 108*a^2*b*c^6*d*e^2*h^2*z - 27*a*b^3*c^5*d*e^2
*h^2*z + 27*a*b^2*c^6*d^2*f^2*g*z - 18*a^2*b^2*c^5*d*e*h^3*z - 216*a^6*c^3*
j^2*k*l*m*z + 216*a^6*c^3*h*j*l^2*m*z + 216*a^6*c^3*f*k*l*m^2*z - 216*a^5*c
^4*f^2*k*l*m*z - 216*a^5*c^4*g^2*j*k*m*z + 216*a^5*c^4*f*j^2*k*l*z + 216*a^
5*c^4*f*h^2*l*m*z + 216*a^5*c^4*e*j^2*k*m*z + 216*a^5*c^4*d*j^2*l*m*z + 216
*a^5*c^4*g*h*j^2*m*z - 216*a^5*c^4*e*j*k^2*l*z - 216*a^5*c^4*d*j*k^2*m*z +
216*a^4*c^5*d^2*j*k*m*z - 18*a^6*b^2*c*k*l*m^3*z + 216*a^5*c^4*f*g*k^2*m*z
- 216*a^5*c^4*d*j*k*l^2*z - 72*a^6*b*c^2*j*l^3*m*z + 18*a^5*b^3*c*j*l^3*m*z
- 216*a^5*c^4*f*h*j*l^2*z + 216*a^5*c^4*e*h*k*l^2*z + 216*a^5*c^4*e*f*l^2*
m*z - 216*a^4*c^5*e^2*h*k*l*z + 216*a^4*c^5*e^2*h*j*m*z - 216*a^4*c^5*e^2*f
*l*m*z - 216*a^5*c^4*e*f*k*m^2*z + 216*a^5*c^4*d*g*k*m^2*z - 216*a^5*c^4*d*
f*l*m^2*z + 216*a^4*c^5*e*f^2*k*m*z + 216*a^4*c^5*d*f^2*l*m*z + 108*a^5*b*c
^3*j^3*k*l*z - 216*a^5*c^4*f*g*h*m^2*z + 216*a^4*c^5*f^2*g*h*m*z + 216*a^4*
c^5*f*g^2*j*k*z - 216*a^4*c^5*e*g^2*j*l*z + 216*a^4*c^5*d*g^2*j*m*z - 72*a^
6*b*c^2*h*k*m^3*z - 72*a^6*b*c^2*g*l*m^3*z + 54*a^5*b^3*c*h*k*m^3*z + 54*a^
5*b^3*c*g*l*m^3*z - 216*a^4*c^5*d*h^2*j*k*z - 18*a^4*b^4*c*f*l^3*m*z + 9*a^
4*b^4*c*h*k*l^3*z - 216*a^4*c^5*e*f*j^2*k*z - 216*a^4*c^5*e*f*h^2*m*z - 216
*a^4*c^5*d*g*j^2*k*z - 216*a^4*c^5*d*f*j^2*l*z - 216*a^4*c^5*d*e*j^2*m*z -
72*a^5*b*c^3*f*k^3*m*z + 72*a^4*b*c^4*g^3*j*m*z + 36*a^5*b*c^3*g*k^3*l*z -
36*a^4*b*c^4*g^3*k*l*z - 216*a^4*c^5*f*g*h*j^2*z + 216*a^4*c^5*d*f*j*k^2*z
- 216*a^3*c^6*d^2*f*j*k*z - 216*a^3*c^6*d^2*e*j*l*z + 72*a^4*b^4*c*f*j*m^3*
z - 63*a^4*b^4*c*e*k*m^3*z - 63*a^4*b^4*c*d*l*m^3*z + 216*a^4*c^5*d*g*h*k^2
*z - 216*a^3*c^6*d^2*g*h*k*z + 216*a^3*c^6*d^2*f*g*m*z - 216*a^3*c^6*d*e^2*
j*k*z + 144*a^5*b*c^3*f*j*l^3*z - 144*a^3*b*c^5*e^3*j*m*z - 72*a^5*b*c^3*e*
k*l^3*z + 72*a^3*b*c^5*e^3*k*l*z - 63*a^4*b^4*c*g*h*m^3*z + 18*a^3*b^5*c*f*
j*l^3*z - 18*a*b^5*c^3*e^3*j*m*z - 9*a^3*b^5*c*e*k*l^3*z + 9*a*b^5*c^3*e^3*

k*l*z - 216*a^4*c^5*d*e*h*l^2*z - 216*a^3*c^6*e^2*f*h*j*z + 216*a^3*c^6*d*e
^2*h*l*z - 126*a*b^4*c^4*d^3*j*m*z + 108*a^4*b*c^4*g*h^3*l*z + 63*a*b^4*c^4
*d^3*k*l*z + 36*a^5*b*c^3*g*h*l^3*z - 9*a^3*b^5*c*g*h*l^3*z + 216*a^4*c^5*d
*e*f*m^2*z + 216*a^3*c^6*d*f^2*g*k*z - 216*a^3*c^6*d*e*f^2*m*z + 36*a^4*b*c
^4*e*j^3*k*z + 36*a^4*b*c^4*d*j^3*l*z - 216*a^3*c^6*d*f*g^2*j*z + 72*a^3*b^
5*c*e*g*m^3*z + 72*a^3*b^5*c*d*h*m^3*z + 72*a^3*b*c^5*f^3*h*k*z + 72*a^3*b*
c^5*f^3*g*l*z + 36*a^4*b*c^4*g*h*j^3*z + 18*a*b^4*c^4*e^3*f*m*z + 9*a^2*b^6
*c*e*g*l^3*z + 9*a^2*b^6*c*d*h*l^3*z - 9*a*b^4*c^4*e^3*h*k*z - 9*a*b^4*c^4*
e^3*g*l*z + 216*a^3*c^6*d*e*f*j^2*z - 144*a^2*b*c^6*d^3*f*m*z + 108*a^3*b*c
^5*e*g^3*k*z - 108*a^3*b*c^5*d*g^3*l*z + 108*a*b^3*c^5*d^3*f*m*z - 72*a^4*b
*c^4*d*h*k^3*z + 72*a^2*b*c^6*d^3*h*k*z - 54*a*b^3*c^5*d^3*h*k*z + 36*a^4*b
*c^4*e*g*k^3*z - 36*a^2*b*c^6*d^3*g*l*z - 27*a*b^3*c^5*d^3*g*l*z - 81*a^2*b
^6*c*d*e*m^3*z + 216*a^4*b*c^4*d*e*l^3*z + 72*a^2*b*c^6*e^3*f*j*z + 72*a^2*
b*c^6*d*e^3*l*z - 18*a*b^3*c^5*e^3*f*j*z - 18*a*b^3*c^5*d*e^3*l*z - 90*a*b^
2*c^6*d^3*f*j*z + 72*a*b^2*c^6*d^3*e*k*z + 36*a^3*b*c^5*e*g*h^3*z - 36*a^2*
b*c^6*e^3*g*h*z + 9*a*b^6*c^2*d*e*k^3*z + 9*a*b^3*c^5*e^3*g*h*z - 180*a^3*b
*c^5*d*e*j^3*z + 18*a*b^2*c^6*d^3*g*h*z - 9*a*b^5*c^3*d*e*j^3*z + 18*a*b^2*
c^6*d*e^3*h*z + 9*a*b^4*c^4*d*e*h^3*z + 36*a^2*b*c^6*d*e*g^3*z - 9*a*b^3*c^
5*d*e*g^3*z - 18*a*b^2*c^6*d*e*f^3*z + 27*a^5*b^2*c^2*h^2*l*m^2*z - 27*a^5*
b^2*c^2*j*k^2*l^2*z + 27*a^4*b^3*c^2*h^2*k^2*m*z + 27*a^4*b^3*c^2*g^2*l^2*m
*z + 27*a^5*b^2*c^2*g*k^2*m^2*z - 27*a^4*b^3*c^2*h^2*k*l^2*z - 27*a^4*b^3*c
^2*g^2*k*m^2*z - 135*a^4*b^2*c^3*e^2*l*m^2*z + 27*a^5*b^2*c^2*e*l^2*m^2*z +
27*a^4*b^3*c^2*h*j^2*l^2*z - 27*a^4*b^2*c^3*h^2*j^2*l*z + 27*a^3*b^4*c^2*e
^2*l*m^2*z - 270*a^4*b^3*c^2*f*j^2*m^2*z - 270*a^4*b^2*c^3*f^2*j*m^2*z + 16
2*a^3*b^4*c^2*f^2*j*m^2*z - 108*a^3*b^3*c^3*f^2*j^2*m*z - 27*a^4*b^2*c^3*h^
2*j*k^2*z - 27*a^4*b^2*c^3*g^2*j*l^2*z + 27*a^3*b^3*c^3*e^2*k^2*m*z + 27*a^
3*b^3*c^3*d^2*l^2*m*z + 27*a^2*b^5*c^2*f^2*j^2*m*z + 162*a^3*b^3*c^3*d^2*k*
m^2*z - 27*a^4*b^3*c^2*d*k^2*m^2*z - 27*a^4*b^2*c^3*g*j^2*k^2*z + 27*a^3*b^
3*c^3*g^2*h^2*m*z - 27*a^2*b^5*c^2*d^2*k*m^2*z + 162*a^3*b^2*c^4*d^2*k^2*l*
z - 108*a^4*b^2*c^3*g*h^2*l^2*z - 27*a^4*b^2*c^3*e*j^2*l^2*z + 27*a^3*b^4*c
^2*g*h^2*l^2*z + 27*a^3*b^2*c^4*e^2*j^2*l*z - 27*a^2*b^4*c^3*d^2*k^2*l*z -
162*a^3*b^3*c^3*f^2*h*l^2*z + 162*a^3*b^3*c^3*e^2*h*m^2*z - 135*a^4*b^2*c^3
*e*h^2*m^2*z + 135*a^3*b^2*c^4*f^2*h^2*l*z + 27*a^3*b^4*c^2*e*h^2*m^2*z - 2
7*a^3*b^3*c^3*g^2*h*k^2*z - 27*a^3*b^2*c^4*e^2*j*k^2*z - 27*a^3*b^2*c^4*d^2
*j*l^2*z + 27*a^2*b^5*c^2*f^2*h*l^2*z - 27*a^2*b^5*c^2*e^2*h*m^2*z - 27*a^2
*b^4*c^3*f^2*h^2*l*z - 27*a^3*b^2*c^4*g^2*h^2*j*z + 27*a^2*b^3*c^4*e^2*g^2*
m*z - 27*a^2*b^3*c^4*d^2*j^2*k*z + 27*a^2*b^3*c^4*d^2*h^2*m*z + 351*a^3*b^2
*c^4*d^2*g*m^2*z - 189*a^2*b^4*c^3*d^2*g*m^2*z + 162*a^3*b^3*c^3*d*g^2*m^2*
z - 162*a^3*b^2*c^4*e^2*g*l^2*z + 135*a^3*b^3*c^3*d*h^2*l^2*z + 135*a^3*b^2
*c^4*f^2*g*k^2*z - 27*a^2*b^5*c^2*d*h^2*l^2*z - 27*a^2*b^5*c^2*d*g^2*m^2*z
- 27*a^2*b^4*c^3*f^2*g*k^2*z + 27*a^2*b^4*c^3*e^2*g*l^2*z + 27*a^2*b^3*c^4*
f^2*g^2*k*z + 27*a^2*b^3*c^4*e^2*h^2*k*z + 135*a^3*b^2*c^4*e*f^2*l^2*z - 10
8*a^3*b^2*c^4*e*g^2*k^2*z + 108*a^2*b^2*c^5*d^2*g^2*l*z + 27*a^3*b^2*c^4*e*
h^2*j^2*z + 27*a^2*b^4*c^3*e*g^2*k^2*z - 27*a^2*b^4*c^3*e*f^2*l^2*z - 27*a^
2*b^3*c^4*e^2*h*j^2*z - 27*a^2*b^2*c^5*e^2*f^2*l*z - 27*a^2*b^2*c^5*e^2*g^2

*j*z - 27*a^2*b^2*c^5*d^2*h^2*j*z + 162*a^2*b^3*c^4*d*e^2*l^2*z - 135*a^2*b
^2*c^5*d^2*g*j^2*z - 27*a^2*b^3*c^4*d*g^2*j^2*z + 27*a^2*b^3*c^4*d*f^2*k^2*
z - 162*a^2*b^2*c^5*d^2*e*k^2*z - 27*a^2*b^2*c^5*e*f^2*h^2*z - 72*a^7*c^2*k
*l*m^3*z + 9*a^5*b^4*k*l*m^3*z + 72*a^6*c^3*j*k^3*m*z - 72*a^6*c^3*h*k*l^3*
z - 72*a^6*c^3*f*l^3*m*z - 72*a^5*c^4*h^3*k*l*z - 72*a^5*c^4*h^3*j*m*z - 9*
a^4*b^5*h*k*m^3*z - 9*a^4*b^5*g*l*m^3*z - 144*a^6*c^3*f*j*m^3*z - 144*a^5*c
^4*h*j^3*k*z - 144*a^5*c^4*g*j^3*l*z - 144*a^5*c^4*f*j^3*m*z - 144*a^4*c^5*
f^3*j*m*z + 72*a^6*c^3*e*k*m^3*z + 72*a^6*c^3*d*l*m^3*z + 72*a^4*c^5*f^3*k*
l*z + 72*a^6*c^3*g*h*m^3*z + 18*b^6*c^3*d^3*j*m*z - 18*a^3*b^6*f*j*m^3*z -
9*b^6*c^3*d^3*k*l*z + 9*a^3*b^6*e*k*m^3*z + 9*a^3*b^6*d*l*m^3*z + 144*a^5*c
^4*d*k^3*l*z + 144*a^3*c^6*d^3*k*l*z - 72*a^5*c^4*f*j*k^3*z - 72*a^3*c^6*d^
3*j*m*z + 9*a^3*b^6*g*h*m^3*z - 72*a^5*c^4*g*h*k^3*z - 72*a^4*c^5*g^3*h*k*z
 - 72*a^4*c^5*f*g^3*m*z - 108*a^5*b*c^3*j^4*m*z + 63*a^6*b^2*c*j*m^4*z + 36
*a^6*b*c^2*k*l^4*z - 9*a^5*b^3*c*k*l^4*z - 144*a^5*c^4*e*g*l^3*z - 144*a^3*
c^6*e^3*g*l*z + 72*a^5*c^4*d*h*l^3*z + 72*a^4*c^5*f*h^3*j*z + 72*a^4*c^5*e*
h^3*k*z + 72*a^4*c^5*d*h^3*l*z + 72*a^3*c^6*e^3*h*k*z + 72*a^3*c^6*e^3*f*m*
z - 18*b^5*c^4*d^3*f*m*z + 9*b^5*c^4*d^3*h*k*z + 9*b^5*c^4*d^3*g*l*z - 9*a^
2*b^7*e*g*m^3*z - 9*a^2*b^7*d*h*m^3*z + 144*a^4*c^5*e*g*j^3*z + 144*a^4*c^5
*d*h*j^3*z - 72*a^5*c^4*d*e*m^3*z - 72*a^3*c^6*e*f^3*k*z - 72*a^3*c^6*d*f^3
*l*z + 144*a^6*b*c^2*f*m^4*z - 108*a^5*b^3*c*f*m^4*z - 72*a^3*c^6*f^3*g*h*z
 + 36*a^5*b*c^3*h*k^4*z - 36*a^3*b*c^5*f^4*m*z + 18*b^4*c^5*d^3*f*j*z - 9*b
^4*c^5*d^3*e*k*z + 9*a^4*b^4*c*g*l^4*z - 144*a^4*c^5*d*e*k^3*z - 144*a^2*c^
7*d^3*e*k*z + 72*a^2*c^7*d^3*f*j*z - 9*b^4*c^5*d^3*g*h*z + 72*a^3*c^6*d*g^3
*h*z + 72*a^2*c^7*d^3*g*h*z - 72*a^5*b*c^3*d*l^4*z - 72*a^4*b*c^4*f*j^4*z +
 45*a*b^2*c^6*d^4*l*z - 36*a^2*b*c^6*e^4*k*z - 9*a^3*b^5*c*d*l^4*z + 9*a*b^
3*c^5*e^4*k*z - 72*a^3*c^6*d*e*h^3*z - 72*a^2*c^7*d*e^3*h*z + 9*b^3*c^6*d^3
*e*g*z + 72*a^2*c^7*d*e*f^3*z + 36*a^3*b*c^5*d*h^4*z - 9*a*b^2*c^6*e^4*g*z
+ 36*a*b*c^7*d^3*f^2*z + 90*a^5*b^2*c^2*j^3*m^2*z + 45*a^5*b^2*c^2*j^2*l^3*
z + 9*a^4*b^3*c^2*j^2*k^3*z - 9*a^4*b^3*c^2*h^3*m^2*z - 45*a^4*b^2*c^3*g^3*
m^2*z + 9*a^3*b^4*c^2*g^3*m^2*z + 198*a^4*b^3*c^2*f^2*m^3*z - 108*a^3*b^3*c
^3*f^3*m^2*z + 18*a^2*b^5*c^2*f^3*m^2*z - 117*a^4*b^2*c^3*f^2*l^3*z + 117*a
^3*b^2*c^4*e^3*m^2*z + 63*a^3*b^4*c^2*f^2*l^3*z - 63*a^2*b^4*c^3*e^3*m^2*z
- 171*a^2*b^3*c^4*d^3*m^2*z - 54*a^3*b^3*c^3*f^2*k^3*z + 9*a^3*b^2*c^4*g^3*
j^2*z + 9*a^2*b^5*c^2*f^2*k^3*z + 18*a^3*b^2*c^4*f^2*j^3*z + 18*a^2*b^3*c^4
*f^3*j^2*z - 9*a^2*b^4*c^3*f^2*j^3*z - 45*a^2*b^2*c^5*e^3*j^2*z + 9*a^2*b^3
*c^4*f^2*h^3*z - 9*a^2*b^2*c^5*f^2*g^3*z + 9*a*b^8*d*e*m^3*z - 36*a*b*c^7*d
^4*h*z - 108*a^6*c^3*h^2*l*m^2*z + 108*a^6*c^3*j*k^2*l^2*z - 108*a^6*c^3*g*
k^2*m^2*z - 108*a^6*c^3*e*l^2*m^2*z + 108*a^5*c^4*h^2*j^2*l*z + 108*a^5*c^4
*e^2*l*m^2*z + 216*a^5*c^4*f^2*j*m^2*z + 108*a^5*c^4*h^2*j*k^2*z + 108*a^5*
c^4*g^2*j*l^2*z + 108*a^5*c^4*g*j^2*k^2*z - 216*a^4*c^5*d^2*k^2*l*z + 108*a
^5*c^4*e*j^2*l^2*z - 108*a^4*c^5*e^2*j^2*l*z - 9*a^6*b^2*c*l^3*m^2*z + 108*
a^5*c^4*e*h^2*m^2*z - 108*a^4*c^5*f^2*h^2*l*z + 108*a^4*c^5*e^2*j*k^2*z + 1
08*a^4*c^5*d^2*j*l^2*z - 144*a^6*b*c^2*j^2*m^3*z + 108*a^4*c^5*g^2*h^2*j*z
- 27*a^4*b^4*c*j^3*m^2*z + 27*a^4*b^3*c^2*j^4*m*z + 9*a^5*b^2*c^2*k^4*l*z +
 216*a^4*c^5*e^2*g*l^2*z - 108*a^4*c^5*f^2*g*k^2*z - 108*a^4*c^5*d^2*g*m^2*

```
z - 9*a^4*b^4*c*j^2*l^3*z - 108*a^4*c^5*e*h^2*j^2*z - 108*a^4*c^5*e*f^2*l^2
*z + 108*a^3*c^6*e^2*f^2*l*z - 36*a^5*b*c^3*j^2*k^3*z + 36*a^5*b*c^3*h^3*m^
2*z + 108*a^3*c^6*e^2*g^2*j*z + 108*a^3*c^6*d^2*h^2*j*z - 216*a^5*b*c^3*f^2
*m^3*z + 144*a^4*b*c^4*f^3*m^2*z + 108*a^3*c^6*d^2*g*j^2*z - 72*a^3*b^5*c*f
^2*m^3*z - 45*a^5*b^2*c^2*g*l^4*z - 9*a^4*b^3*c^2*h*k^4*z - 9*a^3*b^2*c^4*g
^4*l*z + 9*a^2*b^3*c^4*f^4*m*z + 216*a^3*c^6*d^2*e*k^2*z - 9*a^2*b^6*c*f^2*
l^3*z + 9*a*b^6*c^2*e^3*m^2*z + 108*a^3*c^6*e*f^2*h^2*z + 108*a^3*b*c^5*d^3
*m^2*z + 108*a^2*c^7*d^2*e^2*j*z + 72*a^4*b*c^4*f^2*k^3*z + 72*a*b^5*c^3*d^
3*m^2*z - 72*a^3*b*c^5*f^3*j^2*z + 54*a^4*b^3*c^2*d*l^4*z - 45*a^4*b^2*c^3*
e*k^4*z + 18*a^3*b^3*c^3*f*j^4*z + 9*a^3*b^4*c^2*e*k^4*z - 9*a^2*b^2*c^5*f^
4*j*z - 108*a^2*c^7*d^2*f^2*g*z + 9*a^3*b^2*c^4*g*h^4*z + 9*a*b^4*c^4*e^3*j
^2*z - 72*a^2*b*c^6*d^3*j^2*z + 54*a*b^3*c^5*d^3*j^2*z - 36*a^3*b*c^5*f^2*h
^3*z - 9*a^2*b^3*c^4*d*h^4*z + 9*a^2*b^2*c^5*e*g^4*z + 9*a*b^2*c^6*e^3*f^2*
z + 36*a^7*c^2*l^3*m^2*z + 72*a^6*c^3*j^3*m^2*z - 36*a^6*c^3*j^2*l^3*z + 9*
a^4*b^5*j^2*m^3*z + 36*a^5*c^4*g^3*m^2*z + 36*a^5*c^4*f^2*l^3*z - 36*a^4*c^
5*e^3*m^2*z - 9*b^7*c^2*d^3*m^2*z + 9*a^2*b^7*f^2*m^3*z - 36*a^4*c^5*g^3*j^
2*z + 72*a^4*c^5*f^2*j^3*z + 36*a^3*c^6*e^3*j^2*z - 9*b^5*c^4*d^3*j^2*z + 3
6*a^3*c^6*f^2*g^3*z - 9*a^4*b^2*c^3*j^5*z - 36*a^2*c^7*e^3*f^2*z - 9*b^3*c^
6*d^3*f^2*z + 36*a^7*c^2*j*m^4*z - 36*a^6*c^3*k^4*l*z - 18*a^5*b^4*j*m^4*z
+ 36*a^6*c^3*g*l^4*z + 36*a^4*c^5*g^4*l*z + 18*a^4*b^5*f*m^4*z - 9*b^4*c^5*
d^4*l*z + 36*a^5*c^4*e*k^4*z + 36*a^3*c^6*f^4*j*z - 36*a^2*c^7*d^4*l*z - 36
*a^4*c^5*g*h^4*z + 9*b^3*c^6*d^4*h*z - 36*a^3*c^6*e*g^4*z + 36*a^2*c^7*e^4*
g*z - 9*b^2*c^7*d^4*e*z - 36*a^7*b*c*m^5*z + 36*a*c^8*d^4*e*z + 9*a^6*b^3*m
^5*z + 36*a^5*c^4*j^5*z + 9*a^4*b^3*c*g*h*j*k*l*m - 9*a^3*b^4*c*e*g*j*k*l*m
 - 9*a^3*b^4*c*d*h*j*k*l*m - 9*a^3*b^4*c*f*g*h*k*l*m + 36*a^4*b*c^3*d*e*j*k
*l*m + 9*a^2*b^5*c*d*e*j*k*l*m + 36*a^4*b*c^3*e*f*h*j*l*m + 36*a^4*b*c^3*e*
f*g*k*l*m + 36*a^4*b*c^3*d*f*h*k*l*m + 9*a^2*b^5*c*e*f*g*k*l*m + 9*a^2*b^5*
c*d*f*h*k*l*m + 36*a^3*b*c^4*d*e*f*j*k*l + 9*a*b^5*c^2*d*e*f*j*k*l + 36*a^3
*b*c^4*d*e*g*h*k*l + 36*a^3*b*c^4*d*e*f*h*k*m + 36*a^3*b*c^4*d*e*f*g*l*m +
9*a*b^5*c^2*d*e*f*h*k*m + 9*a*b^5*c^2*d*e*f*g*l*m - 9*a*b^4*c^3*d*e*f*h*j*k
 - 9*a*b^4*c^3*d*e*f*g*j*l - 9*a*b^4*c^3*d*e*f*g*h*m + 9*a*b^3*c^4*d*e*f*g*
h*j - 9*a*b^6*c*d*e*f*k*l*m + 18*a^4*b^2*c^2*e*g*j*k*l*m + 18*a^4*b^2*c^2*d
*h*j*k*l*m + 18*a^4*b^2*c^2*f*g*h*k*l*m - 36*a^3*b^3*c^2*d*e*j*k*l*m - 36*a
^3*b^3*c^2*e*f*g*k*l*m - 36*a^3*b^3*c^2*d*f*h*k*l*m + 9*a^3*b^3*c^2*f*g*h*j
*k*l + 9*a^3*b^3*c^2*e*g*h*j*k*m + 9*a^3*b^3*c^2*d*g*h*j*l*m - 108*a^3*b^2*
c^3*d*e*f*k*l*m + 54*a^2*b^4*c^2*d*e*f*k*l*m - 36*a^3*b^2*c^3*d*f*g*j*k*m +
 18*a^3*b^2*c^3*e*f*g*j*k*l + 18*a^3*b^2*c^3*d*f*h*j*k*l + 18*a^3*b^2*c^3*d
*e*h*j*k*m + 18*a^3*b^2*c^3*d*e*g*j*l*m - 9*a^2*b^4*c^2*e*f*g*j*k*l - 9*a^2
*b^4*c^2*d*f*h*j*k*l - 9*a^2*b^4*c^2*d*e*h*j*k*m - 9*a^2*b^4*c^2*d*e*g*j*l*
m + 18*a^3*b^2*c^3*e*f*g*h*k*m + 18*a^3*b^2*c^3*d*f*g*h*l*m - 9*a^2*b^4*c^2
*e*f*g*h*k*m - 9*a^2*b^4*c^2*d*f*g*h*l*m - 36*a^2*b^3*c^3*d*e*f*j*k*l - 36*
a^2*b^3*c^3*d*e*f*h*k*m - 36*a^2*b^3*c^3*d*e*f*g*l*m + 9*a^2*b^3*c^3*e*f*g*
h*j*k + 9*a^2*b^3*c^3*d*f*g*h*j*l + 9*a^2*b^3*c^3*d*e*g*h*j*m + 18*a^2*b^2*
c^4*d*e*f*h*j*k + 18*a^2*b^2*c^4*d*e*f*g*j*l + 18*a^2*b^2*c^4*d*e*f*g*h*m -
 9*a^5*b^2*c*h*j*k^2*l*m - 9*a^5*b^2*c*g*j*k*l^2*m + 27*a^5*b^2*c*f*j*k*l*m
```

^2 - 9*a^4*b^3*c*f*j^2*k*l*m + 9*a^3*b^4*c*f^2*j*k*l*m - 18*a^5*b*c^2*e*j*k
^2*l*m - 9*a^5*b^2*c*g*h*k*l*m^2 + 9*a^4*b^3*c*e*j*k^2*l*m - 18*a^5*b*c^2*f
*h*k^2*l*m - 18*a^5*b*c^2*d*j*k*l^2*m + 9*a^4*b^3*c*f*h*k^2*l*m + 9*a^4*b^3
*c*d*j*k*l^2*m + 36*a^5*b*c^2*e*h*k*l^2*m - 36*a^4*b*c^3*e^2*h*k*l*m + 18*a
^5*b*c^2*f*h*j*l^2*m - 18*a^5*b*c^2*f*g*k*l^2*m - 18*a^4*b^3*c*e*h*k*l^2*m
+ 9*a^4*b^3*c*f*g*k*l^2*m + 9*a^3*b^4*c*e*h^2*k*l*m - 9*a^2*b^5*c*e^2*h*k*l
*m - 54*a^5*b*c^2*e*h*j*l*m^2 - 18*a^5*b*c^2*e*g*k*l*m^2 - 18*a^5*b*c^2*d*h
*k*l*m^2 + 18*a^4*b^3*c*e*h*j*l*m^2 - 9*a^4*b^3*c*f*h*j*k*m^2 - 9*a^4*b^3*c
*f*g*j*l*m^2 + 9*a^4*b^3*c*e*g*k*l*m^2 + 9*a^4*b^3*c*d*h*k*l*m^2 + 18*a^4*b
*c^3*f*g^2*j*k*m - 18*a^4*b*c^3*e*g^2*j*l*m + 18*a^3*b^4*c*d*g*k^2*l*m - 9*
a^3*b^4*c*e*f*k^2*l*m - 9*a^2*b^5*c*d*g^2*k*l*m - 18*a^4*b*c^3*f*g^2*h*l*m
- 18*a^4*b*c^3*d*h^2*j*k*m - 9*a^3*b^4*c*d*f*k*l^2*m - 54*a^4*b*c^3*d*g*j^2
*k*m - 18*a^4*b*c^3*f*g*h^2*k*m - 18*a^4*b*c^3*e*g*j^2*k*l - 18*a^4*b*c^3*d
*h*j^2*k*l - 18*a^3*b^4*c*d*g*j*k*m^2 + 9*a^3*b^4*c*e*f*j*k*m^2 + 9*a^3*b^4
*c*d*f*j*l*m^2 - 9*a^3*b^4*c*d*e*k*l*m^2 - 54*a^3*b*c^4*d^2*f*j*k*m + 36*a^
4*b*c^3*d*g*j*k^2*l - 36*a^3*b*c^4*d^2*g*j*k*l - 18*a^4*b*c^3*e*f*j*k^2*l +
 18*a^4*b*c^3*d*f*j*k^2*m - 18*a^3*b*c^4*d^2*e*j*l*m + 9*a^3*b^4*c*f*g*h*j*
m^2 - 9*a*b^5*c^2*d^2*g*j*k*l + 36*a^4*b*c^3*d*g*h*k^2*m - 36*a^3*b*c^4*d^2
*g*h*k*m + 18*a^4*b*c^3*e*g*h*k^2*l - 18*a^4*b*c^3*e*f*h*k^2*m - 18*a^4*b*c
^3*d*f*j*k*l^2 - 18*a^3*b*c^4*d^2*f*h*l*m - 18*a^3*b*c^4*d*e^2*j*k*m - 9*a*
b^5*c^2*d^2*g*h*k*m - 54*a^4*b*c^3*d*g*h*k*l^2 - 54*a^3*b*c^4*e^2*f*h*j*m -
 18*a^4*b*c^3*d*f*g*l^2*m - 18*a^3*b*c^4*e^2*f*g*k*m - 54*a^4*b*c^3*d*f*g*k
*m^2 - 36*a^4*b*c^3*e*f*g*j*m^2 - 36*a^4*b*c^3*d*f*h*j*m^2 + 36*a^3*b*c^4*e
*f^2*g*j*m + 36*a^3*b*c^4*d*f^2*h*j*m - 18*a^4*b*c^3*d*e*h*k*m^2 - 18*a^4*b
*c^3*d*e*g*l*m^2 + 18*a^3*b*c^4*e*f^2*h*j*l - 18*a^3*b*c^4*e*f^2*g*k*l - 18
*a^3*b*c^4*d*f^2*h*k*l + 18*a^3*b*c^4*d*f^2*g*k*m - 9*a^2*b^5*c*e*f*g*j*m^2
 - 9*a^2*b^5*c*d*f*h*j*m^2 - 54*a^3*b*c^4*d*f*g^2*j*m - 18*a^3*b*c^4*e*f*g^
2*j*l - 18*a*b^4*c^3*d^2*f*g*j*m + 9*a*b^4*c^3*d^2*g*h*j*k + 9*a*b^4*c^3*d^
2*f*g*k*l + 9*a*b^4*c^3*d^2*e*g*k*m - 9*a*b^4*c^3*d^2*e*f*l*m - 18*a^3*b*c^
4*e*f*g^2*h*m - 18*a^3*b*c^4*d*f*h^2*j*k - 9*a*b^4*c^3*d*e^2*f*k*m + 18*a^3
*b*c^4*d*f*g*j^2*k - 18*a^3*b*c^4*d*f*g*h^2*m - 18*a^3*b*c^4*d*e*h*j^2*k -
18*a^3*b*c^4*d*e*g*j^2*l + 18*a*b^4*c^3*d*e*f^2*j*m - 9*a*b^5*c^2*d*e*f*j^2
*m - 9*a*b^4*c^3*d*e*f^2*k*l - 18*a^2*b*c^5*d^2*e*f*j*l - 9*a*b^3*c^4*d^2*e
*g*j*k + 9*a*b^3*c^4*d^2*e*f*j*l - 54*a^2*b*c^5*d^2*e*g*h*l - 18*a^2*b*c^5*
d^2*e*f*h*m - 18*a^2*b*c^5*d*e^2*f*j*k + 18*a*b^3*c^4*d^2*e*g*h*l - 9*a*b^3
*c^4*d^2*f*g*h*k + 9*a*b^3*c^4*d^2*e*f*h*m + 9*a*b^3*c^4*d*e^2*f*j*k - 36*a
^3*b*c^4*d*e*f*h*l^2 + 36*a^2*b*c^5*d*e^2*f*h*l + 18*a^2*b*c^5*d*e^2*g*h*k
- 18*a^2*b*c^5*d*e^2*f*g*m - 18*a*b^3*c^4*d*e^2*f*h*l - 9*a*b^5*c^2*d*e*f*h
*l^2 + 9*a*b^4*c^3*d*e*f*h^2*l + 9*a*b^3*c^4*d*e^2*f*g*m - 18*a^2*b*c^5*d*e
*f^2*h*k - 18*a^2*b*c^5*d*e*f^2*g*l + 9*a*b^3*c^4*d*e*f^2*h*k + 9*a*b^3*c^4
*d*e*f^2*g*l + 27*a*b^2*c^5*d^2*e*f*g*k + 9*a*b^4*c^3*d*e*f*g*k^2 - 9*a*b^3
*c^4*d*e*f*g^2*k - 9*a*b^2*c^5*d^2*e*f*h*j - 9*a*b^2*c^5*d*e^2*f*g*j - 9*a*
b^2*c^5*d*e*f^2*g*h + 72*a^4*c^4*d*f*g*j*k*m + 72*a^4*c^4*d*e*f*k*l*m + 9*a
*b^6*c*d^2*g*k*l*m + 9*a*b^6*c*d*e*f*j*m^2 - 27*a^4*b^2*c^2*f^2*j*k*l*m - 9
*a^4*b^2*c^2*g^2*h*j*l*m + 36*a^3*b^3*c^2*e^2*h*k*l*m - 18*a^4*b^2*c^2*e*h^

2*k*l*m - 9*a^4*b^2*c^2*g*h^2*j*k*m + 18*a^4*b^2*c^2*f*h*j^2*k*m + 18*a^4*b
^2*c^2*f*g*j^2*l*m - 18*a^4*b^2*c^2*e*h*j^2*l*m - 9*a^4*b^2*c^2*g*h*j^2*k*l
 - 9*a^3*b^3*c^2*f^2*h*j*k*m - 9*a^3*b^3*c^2*f^2*g*j*l*m - 63*a^4*b^2*c^2*d
*g*k^2*l*m + 63*a^3*b^2*c^3*d^2*g*k*l*m - 45*a^2*b^4*c^2*d^2*g*k*l*m + 36*a
^4*b^2*c^2*e*f*k^2*l*m + 27*a^3*b^3*c^2*d*g^2*k*l*m - 9*a^4*b^2*c^2*f*h*j*k
^2*l - 9*a^4*b^2*c^2*e*h*j*k^2*m + 9*a^3*b^3*c^2*e*g^2*j*l*m - 9*a^3*b^2*c^
3*d^2*h*j*l*m + 36*a^4*b^2*c^2*d*f*k*l^2*m + 27*a^4*b^2*c^2*e*h*j*k*l^2 - 2
7*a^3*b^2*c^3*e^2*h*j*k*l - 18*a^3*b^2*c^3*e^2*f*j*l*m - 9*a^4*b^2*c^2*f*g*
j*k*l^2 - 9*a^4*b^2*c^2*d*g*j*l^2*m + 9*a^3*b^3*c^2*f*g^2*h*l*m - 9*a^3*b^3
*c^2*e*h^2*j*k*l + 9*a^3*b^3*c^2*d*h^2*j*k*m - 9*a^3*b^2*c^3*e^2*g*j*k*m +
9*a^2*b^4*c^2*e^2*h*j*k*l + 72*a^4*b^2*c^2*d*g*j*k*m^2 + 36*a^4*b^2*c^2*d*e
*k*l*m^2 + 27*a^4*b^2*c^2*e*g*h*l^2*m - 27*a^4*b^2*c^2*e*f*j*k*m^2 - 27*a^4
*b^2*c^2*d*f*j*l*m^2 - 27*a^3*b^2*c^3*e^2*g*h*l*m + 27*a^3*b^2*c^3*e*f^2*j*
k*m + 27*a^3*b^2*c^3*d*f^2*j*l*m + 18*a^3*b^3*c^2*d*g*j^2*k*m + 9*a^3*b^3*c
^2*f*g*h^2*k*m + 9*a^3*b^3*c^2*e*g*j^2*k*l - 9*a^3*b^3*c^2*e*g*h^2*l*m - 9*
a^3*b^3*c^2*e*f*j^2*k*m + 9*a^3*b^3*c^2*d*h*j^2*k*l - 9*a^3*b^3*c^2*d*f*j^2
*l*m + 9*a^2*b^4*c^2*e^2*g*h*l*m + 36*a^2*b^3*c^3*d^2*g*j*k*l - 27*a^4*b^2*
c^2*f*g*h*j*m^2 + 27*a^3*b^2*c^3*f^2*g*h*j*m - 18*a^4*b^2*c^2*e*f*h*l*m^2 -
 18*a^3*b^3*c^2*d*g*j*k^2*l - 18*a^3*b^2*c^3*d*g^2*j*k*l + 18*a^2*b^3*c^3*d
^2*f*j*k*m - 9*a^4*b^2*c^2*e*g*h*k*m^2 - 9*a^4*b^2*c^2*d*g*h*l*m^2 - 9*a^3*
b^3*c^2*f*g*h*j^2*m + 9*a^3*b^3*c^2*e*f*j*k^2*l - 9*a^3*b^2*c^3*f^2*g*h*k*l
 + 9*a^2*b^4*c^2*d*g^2*j*k*l + 9*a^2*b^3*c^3*d^2*e*j*l*m + 36*a^3*b^2*c^3*e
*f*g^2*l*m + 36*a^2*b^3*c^3*d^2*g*h*k*m - 18*a^3*b^3*c^2*d*g*h*k^2*m - 18*a
^3*b^2*c^3*d*g^2*h*k*m + 9*a^3*b^3*c^2*e*f*h*k^2*m + 9*a^3*b^3*c^2*d*f*j*k*
l^2 - 9*a^3*b^2*c^3*f*g^2*h*j*l - 9*a^3*b^2*c^3*e*g^2*h*j*m - 9*a^2*b^4*c^2
*e*f*g^2*l*m + 9*a^2*b^4*c^2*d*g^2*h*k*m + 9*a^2*b^3*c^3*d^2*f*h*l*m + 9*a^
2*b^3*c^3*d*e^2*j*k*m + 36*a^3*b^2*c^3*d*f*h^2*k*m + 36*a^3*b^2*c^3*d*e*j^2
*k*l + 18*a^3*b^3*c^2*d*g*h*k*l^2 + 18*a^3*b^2*c^3*e*g*h^2*j*l + 18*a^3*b^2
*c^3*e*f*h^2*k*l - 18*a^3*b^2*c^3*e*f*h^2*j*m - 18*a^3*b^2*c^3*d*g*h^2*k*l
+ 18*a^3*b^2*c^3*d*e*h^2*l*m + 18*a^2*b^3*c^3*e^2*f*h*j*m - 9*a^3*b^3*c^2*e
*g*h*j*l^2 - 9*a^3*b^3*c^2*e*f*h*k*l^2 + 9*a^3*b^3*c^2*d*f*g*l^2*m - 9*a^3*
b^3*c^2*d*e*h*l^2*m - 9*a^3*b^2*c^3*f*g*h^2*j*k - 9*a^3*b^2*c^3*d*g*h^2*j*m
 - 9*a^2*b^4*c^2*d*f*h^2*k*m - 9*a^2*b^4*c^2*d*e*j^2*k*l - 9*a^2*b^3*c^3*e^
2*g*h*j*l - 9*a^2*b^3*c^3*e^2*f*h*k*l + 9*a^2*b^3*c^3*e^2*f*g*k*m - 9*a^2*b
^3*c^3*d*e^2*h*l*m + 36*a^3*b^3*c^2*e*f*g*j*m^2 + 36*a^3*b^3*c^2*d*f*h*j*m^
2 + 18*a^3*b^3*c^2*d*f*g*k*m^2 - 18*a^3*b^2*c^3*e*f*g*j^2*m - 18*a^3*b^2*c^
3*d*f*h*j^2*m - 18*a^2*b^3*c^3*e*f^2*g*j*m - 18*a^2*b^3*c^3*d*f^2*h*j*m + 9
*a^3*b^3*c^2*d*e*h*k*m^2 + 9*a^3*b^3*c^2*d*e*g*l*m^2 - 9*a^3*b^2*c^3*e*g*h*
j^2*k - 9*a^3*b^2*c^3*d*g*h*j^2*l + 9*a^2*b^4*c^2*e*f*g*j^2*m + 9*a^2*b^4*c
^2*d*f*h*j^2*m + 9*a^2*b^3*c^3*e*f^2*g*k*l + 9*a^2*b^3*c^3*d*f^2*h*k*l + 72
*a^2*b^2*c^4*d^2*f*g*j*m + 36*a^2*b^2*c^4*d^2*e*f*l*m + 27*a^3*b^2*c^3*d*g*
h*j*k^2 + 27*a^3*b^2*c^3*d*f*g*k^2*l + 27*a^3*b^2*c^3*d*e*g*k^2*m - 27*a^2*
b^2*c^4*d^2*g*h*j*k - 27*a^2*b^2*c^4*d^2*f*g*k*l - 27*a^2*b^2*c^4*d^2*e*g*k
*m + 18*a^2*b^3*c^3*d*f*g^2*j*m - 18*a^2*b^2*c^4*d^2*e*h*k*l - 9*a^3*b^2*c^
3*e*f*h*j*k^2 + 9*a^2*b^3*c^3*e*f*g^2*j*l - 9*a^2*b^3*c^3*d*g^2*h*j*k - 9*a

^2*b^3*c^3*d*f*g^2*k*l - 9*a^2*b^3*c^3*d*e*g^2*k*m - 9*a^2*b^2*c^4*d^2*f*h*
j*l - 9*a^2*b^2*c^4*d^2*e*h*j*m + 36*a^2*b^2*c^4*d*e^2*f*k*m - 27*a^3*b^2*c
^3*d*e*h*j*l^2 + 27*a^2*b^2*c^4*d*e^2*h*j*l - 18*a^3*b^2*c^3*d*e*g*k*l^2 -
9*a^3*b^2*c^3*d*f*g*j*l^2 + 9*a^2*b^4*c^2*d*e*h*j*l^2 + 9*a^2*b^3*c^3*e*f*g
^2*h*m + 9*a^2*b^3*c^3*d*f*h^2*j*k - 9*a^2*b^3*c^3*d*e*h^2*j*l - 9*a^2*b^2*
c^4*e^2*f*g*j*k - 9*a^2*b^2*c^4*d*e^2*g*j*m + 63*a^3*b^2*c^3*d*e*f*j*m^2 -
63*a^2*b^2*c^4*d*e*f^2*j*m - 45*a^2*b^4*c^2*d*e*f*j*m^2 + 36*a^2*b^2*c^4*d*
e*f^2*k*l - 27*a^3*b^2*c^3*e*f*g*h*l^2 + 27*a^2*b^3*c^3*d*e*f*j^2*m + 27*a^
2*b^2*c^4*e^2*f*g*h*l + 9*a^2*b^4*c^2*e*f*g*h*l^2 - 9*a^2*b^3*c^3*e*f*g*h^2
*l + 9*a^2*b^3*c^3*d*f*g*h^2*m + 9*a^2*b^3*c^3*d*e*h*j^2*k + 9*a^2*b^3*c^3*
d*e*g*j^2*l + 18*a^2*b^2*c^4*d*e*g^2*j*k - 9*a^3*b^2*c^3*d*e*g*h*m^2 - 9*a^
2*b^3*c^3*d*e*g*j*k^2 - 9*a^2*b^2*c^4*e*f^2*g*h*k - 9*a^2*b^2*c^4*d*f^2*g*h
*l + 18*a^2*b^2*c^4*d*f*g^2*h*k - 18*a^2*b^2*c^4*d*e*g^2*h*l - 9*a^2*b^3*c^
3*d*f*g*h*k^2 - 9*a^2*b^2*c^4*e*f*g^2*h*j + 36*a^2*b^3*c^3*d*e*f*h*l^2 - 18
*a^2*b^2*c^4*d*e*f*h^2*l - 9*a^2*b^2*c^4*d*f*g*h^2*j - 9*a^2*b^2*c^4*d*e*g*
h*j^2 - 27*a^2*b^2*c^4*d*e*f*g*k^2 + 18*a^2*b^2*c^4*d^2*f*h*k^2 - 9*a^2*b^3
*c^3*e*f*g^2*k^2 - 9*a^2*b^2*c^4*e^2*f*h*j^2 - 9*a^2*b^2*c^4*d*f^2*h^2*k +
45*a^2*b^3*c^3*d*e*f^2*m^2 + 36*a^2*b^2*c^4*d^2*e*g*l^2 + 9*a^2*b^3*c^3*d*e
*g^2*l^2 + 9*a^2*b^2*c^4*e*f^2*g*j^2 + 9*a^2*b^2*c^4*d*f^2*h*j^2 - 9*a^2*b^
2*c^4*d*e^2*h*k^2 - 36*a^2*b^2*c^4*d*e^2*f*l^2 - 9*a^2*b^2*c^4*d*f*g^2*j^2
- 12*a^6*b*c*h*k*l^3*m + 3*a*b^6*c*e^3*k*l*m + 3*a*b^6*c*d*e*f*l^3 - 12*a*b
*c^6*d*e^3*f*h + 9*a^5*b^2*c*h^2*k*l^2*m + 18*a^5*b*c^2*g^2*k^2*l*m - 9*a^5
*b^2*c*h^2*j*l*m^2 + 9*a^5*b*c^2*h^2*j^2*l*m - 9*a^4*b^3*c*g^2*k^2*l*m - 3*
a^4*b^2*c^2*g^3*k*l*m + 18*a^5*b*c^2*f^2*k*l*m^2 + 15*a^3*b^3*c^2*f^3*k*l*m
 + 9*a^5*b^2*c*h*j^2*k*m^2 + 9*a^5*b^2*c*g*j^2*l*m^2 - 9*a^5*b^2*c*f*k^2*l^
2*m + 9*a^5*b*c^2*h^2*j*k^2*m + 9*a^5*b*c^2*g^2*j*l^2*m - 9*a^4*b^3*c*f^2*k
*l*m^2 + 36*a^3*b^2*c^3*e^3*k*l*m - 27*a^5*b*c^2*g^2*j*k*m^2 - 18*a^5*b*c^2
*h^2*j*k*l^2 - 18*a^2*b^4*c^2*e^3*k*l*m - 9*a^5*b^2*c*g*j*k^2*m^2 - 9*a^5*b
^2*c*e*k^2*l*m^2 + 9*a^5*b*c^2*h*j^2*k^2*l + 9*a^5*b*c^2*g*j^2*k^2*m + 9*a^
4*b^3*c*g^2*j*k*m^2 + 9*a^3*b^4*c*e^2*k*l^2*m + 3*a^4*b^2*c^2*h^3*j*k*l - 5
4*a^4*b*c^3*d^2*k^2*l*m - 51*a^2*b^3*c^3*d^3*k*l*m - 27*a^4*b*c^3*e^2*j^2*l
*m - 18*a^5*b*c^2*g*h^2*l^2*m - 9*a^5*b^2*c*e*j*l^2*m^2 - 9*a^5*b^2*c*d*k*l
^2*m^2 + 9*a^5*b*c^2*g^2*h*l*m^2 + 9*a^5*b*c^2*g*j^2*k*l^2 + 9*a^5*b*c^2*e*
j^2*l^2*m - 9*a^3*b^4*c*e^2*j*l*m^2 - 9*a^2*b^5*c*d^2*k^2*l*m + 3*a^4*b^2*c
^2*g*h^3*l*m - 3*a^3*b^3*c^2*g^3*j*k*l + 18*a^5*b*c^2*e*j^2*k*m^2 + 18*a^5*
b*c^2*d*j^2*l*m^2 + 18*a^4*b*c^3*f^2*j^2*k*l + 9*a^5*b*c^2*g*h^2*k*m^2 + 9*
a^5*b*c^2*f*h^2*l*m^2 + 9*a^5*b*c^2*f*j*k^2*l^2 - 9*a^4*b^3*c*e*j^2*k*m^2 -
 9*a^4*b^3*c*d*j^2*l*m^2 + 9*a^4*b^2*c^2*f*j^3*k*l + 9*a^4*b^2*c^2*e*j^3*k*
m + 9*a^4*b^2*c^2*d*j^3*l*m + 9*a^4*b*c^3*f^2*h^2*l*m + 9*a^4*b*c^3*e^2*j*k
^2*m + 9*a^4*b*c^3*d^2*j*l^2*m - 3*a^3*b^3*c^2*g^3*h*k*m - 3*a^3*b^2*c^3*f^
3*j*k*l + 3*a^2*b^4*c^2*f^3*j*k*l + 45*a^4*b*c^3*d^2*j*k*m^2 - 27*a^5*b*c^2
*d*j*k^2*m^2 + 18*a^5*b*c^2*g*h*j^2*m^2 + 18*a^4*b*c^3*e^2*j*k*l^2 + 15*a^2
*b^3*c^3*e^3*j*k*l - 12*a^3*b^2*c^3*f^3*h*k*m - 12*a^3*b^2*c^3*f^3*g*l*m +
9*a^5*b*c^2*g*h*k^2*l^2 - 9*a^4*b^3*c*g*h*j^2*m^2 + 9*a^4*b^3*c*d*j*k^2*m^2
 + 9*a^4*b^2*c^2*g*h*j^3*m + 9*a^4*b*c^3*g^2*h^2*k*l + 9*a^4*b*c^3*g^2*h^2*

j*m + 9*a^2*b^5*c*d^2*j*k*m^2 + 3*a^2*b^4*c^2*f^3*h*k*m + 3*a^2*b^4*c^2*f^3 *g*l*m + 36*a^2*b^2*c^4*d^3*j*k*l + 18*a^4*b*c^3*e^2*g*l^2*m + 15*a^2*b^3*c ^3*e^3*g*l*m + 12*a^4*b^2*c^2*d*j*k^3*l + 9*a^5*b*c^2*f*g*k^2*m^2 + 9*a^5*b *c^2*e*h*k^2*m^2 + 9*a^4*b*c^3*g^2*h*j^2*l + 9*a^4*b*c^3*f^2*h*k^2*l + 9*a^ 4*b*c^3*f^2*g*k^2*m + 9*a^4*b*c^3*d^2*h*l*m^2 - 9*a^3*b^3*c^2*e*h^3*k*m + 6 *a^2*b^3*c^3*e^3*h*k*m + 45*a^4*b*c^3*e^2*h*j*m^2 + 36*a^2*b^2*c^4*d^3*h*k* m - 33*a^3*b^2*c^3*d*g^3*l*m - 27*a^4*b*c^3*f^2*h*j*l^2 - 27*a^4*b*c^3*e^2* f*l*m^2 - 27*a^4*b*c^3*e*h^2*j^2*m - 18*a^4*b*c^3*g^2*h*j*k^2 - 18*a^4*b*c^ 3*f*g^2*k^2*l - 18*a^4*b*c^3*e*g^2*k^2*m - 18*a^3*b*c^4*d^2*g^2*l*m + 12*a^ 4*b^2*c^2*d*h*k^3*m + 9*a^5*b*c^2*e*f*l^2*m^2 + 9*a^5*b*c^2*d*g*l^2*m^2 + 9 *a^4*b*c^3*f^2*g*k*l^2 + 9*a^4*b*c^3*e^2*g*k*m^2 + 9*a^4*b*c^3*g*h^2*j^2*k + 9*a^4*b*c^3*f*h^2*j^2*l + 9*a^4*b*c^3*e*f^2*l^2*m - 9*a^3*b^4*c*e*h^2*j*m ^2 + 9*a^3*b*c^4*e^2*f^2*l*m + 9*a^2*b^5*c*e^2*h*j*m^2 + 9*a^2*b^4*c^2*d*g^ 3*l*m - 9*a^2*b^2*c^4*d^3*g*l*m - 9*a*b^5*c^2*d^2*g^2*l*m - 6*a^4*b^2*c^2*e *h*k^3*l - 6*a^3*b^2*c^3*f*g^3*j*m + 3*a^4*b^2*c^2*g*h*j*k^3 + 3*a^4*b^2*c^ 2*f*g*k^3*l + 3*a^4*b^2*c^2*e*g*k^3*m + 3*a^3*b^2*c^3*g^3*h*j*k + 3*a^3*b^2 *c^3*f*g^3*k*l + 3*a^3*b^2*c^3*e*g^3*k*m - 27*a^3*b*c^4*d^2*h^2*k*l + 18*a^ 4*b*c^3*e*f^2*k*m^2 + 18*a^4*b*c^3*d*f^2*l*m^2 + 9*a^4*b*c^3*f*h^2*j*k^2 + 9*a^4*b*c^3*f*g^2*j*l^2 + 9*a^4*b*c^3*e*g^2*k*l^2 + 9*a^4*b*c^3*d*h^2*k^2*l + 9*a^3*b^4*c*e*g*j^2*m^2 + 9*a^3*b^4*c*d*h*j^2*m^2 - 9*a^3*b^3*c^2*e*g*j^ 3*m - 9*a^3*b^3*c^2*d*h*j^3*m + 9*a^3*b*c^4*e^2*g^2*k*l + 9*a^3*b*c^4*e^2*g ^2*j*m + 9*a^3*b*c^4*d^2*h^2*j*m - 3*a^2*b^3*c^3*f^3*h*j*k - 3*a^2*b^3*c^3* f^3*g*j*l - 3*a^2*b^3*c^3*e*f^3*k*m - 3*a^2*b^3*c^3*d*f^3*l*m + 45*a^4*b*c^ 3*d*g^2*j*m^2 + 45*a^3*b*c^4*d^2*g*j^2*m + 24*a^4*b^2*c^2*d*g*k*l^3 + 24*a^ 2*b^2*c^4*e^3*f*j*m + 18*a^4*b*c^3*f^2*g*h*m^2 + 18*a^4*b*c^3*d*h^2*j*l^2 + 18*a^3*b*c^4*e^2*h^2*j*k - 12*a^4*b^2*c^2*e*g*j*l^3 - 12*a^4*b^2*c^2*e*f*k *l^3 - 12*a^4*b^2*c^2*d*e*l^3*m - 12*a^2*b^2*c^4*e^3*g*j*l - 12*a^2*b^2*c^4 *e^3*f*k*l - 12*a^2*b^2*c^4*d*e^3*l*m + 9*a^4*b*c^3*f*g*j^2*k^2 + 9*a^4*b*c ^3*e*h*j^2*k^2 + 9*a^3*b^2*c^3*e*h^3*j*k + 9*a^3*b^2*c^3*d*h^3*j*l + 9*a^3* b*c^4*f^2*g^2*j*k + 9*a^3*b*c^4*d^2*h*j^2*l + 9*a^2*b^5*c*d*g^2*j*m^2 + 9*a *b^5*c^2*d^2*g*j^2*m - 3*a^4*b^2*c^2*d*h*j*l^3 - 3*a^2*b^3*c^3*f^3*g*h*m - 3*a^2*b^2*c^4*e^3*h*j*k + 18*a^4*b*c^3*f*g*h^2*l^2 + 18*a^3*b*c^4*e^2*g*h^2 *m + 18*a^3*b*c^4*d^2*h*j*k^2 + 18*a^3*b*c^4*d^2*f*k^2*l + 18*a^3*b*c^4*d^2 *e*k^2*m + 9*a^4*b*c^3*e*g^2*h*m^2 + 9*a^4*b*c^3*e*f*j^2*l^2 + 9*a^4*b*c^3* d*g*j^2*l^2 + 9*a^3*b^2*c^3*f*g*h^3*l + 9*a^3*b^2*c^3*e*g*h^3*m + 9*a^3*b*c ^4*f^2*g^2*h*l + 9*a^3*b*c^4*e^2*g*j^2*k + 9*a^3*b*c^4*e^2*f*j^2*l - 9*a^2* b^3*c^3*d*g^3*j*l + 9*a*b^4*c^3*d^2*g^2*j*l - 3*a^4*b^2*c^2*f*g*h*l^3 - 3*a ^3*b^3*c^2*e*g*j*k^3 - 3*a^3*b^3*c^2*d*h*j*k^3 - 3*a^3*b^3*c^2*d*f*k^3*l - 3*a^3*b^3*c^2*d*e*k^3*m - 3*a^2*b^2*c^4*e^3*g*h*m - 33*a^3*b^2*c^3*d*e*j^3* m - 27*a^4*b*c^3*e*f*h^2*m^2 - 27*a^3*b*c^4*d^2*e*k*l^2 - 18*a^4*b*c^3*d*e* j^2*m^2 - 18*a^3*b*c^4*e*f^2*j^2*k - 18*a^3*b*c^4*d*f^2*j^2*l - 9*a^4*b^2*c ^2*d*e*j*m^3 + 9*a^4*b*c^3*d*g*h^2*m^2 + 9*a^4*b*c^3*d*e*k^2*l^2 + 9*a^3*b* c^4*f^2*g*h^2*k + 9*a^3*b*c^4*e^2*f*j*k^2 + 9*a^3*b*c^4*d^2*f*j*l^2 + 9*a^3 *b*c^4*e*f^2*h^2*m + 9*a^3*b*c^4*d*e^2*k^2*l - 9*a^2*b^5*c*d*e*j^2*m^2 + 9* a^2*b^4*c^2*d*e*j^3*m - 9*a^2*b^3*c^3*d*g^3*h*m + 9*a^2*b*c^5*d^2*e^2*k*l +

9*a^2*b*c^5*d^2*e^2*j*m + 9*a*b^4*c^3*d^2*g^2*h*m - 6*a^3*b^2*c^3*d*g*j^3*
k - 3*a^3*b^3*c^2*f*g*h*k^3 + 3*a^3*b^2*c^3*e*f*j^3*k + 3*a^3*b^2*c^3*d*f*j
^3*l + 3*a^2*b^2*c^4*e*f^3*j*k + 3*a^2*b^2*c^4*d*f^3*j*l + 45*a^3*b*c^4*d^2
*g*h*l^2 + 36*a^4*b^2*c^2*e*f*g*m^3 + 36*a^4*b^2*c^2*d*f*h*m^3 - 27*a^3*b*c
^4*e^2*g*h*k^2 - 27*a^3*b*c^4*d*g^2*h^2*l - 18*a^3*b*c^4*f^2*g*h*j^2 + 18*a
^3*b*c^4*d*e^2*j*l^2 + 15*a^3*b^3*c^2*d*e*j*l^3 + 12*a^2*b^2*c^4*e*f^3*g*m
+ 12*a^2*b^2*c^4*d*f^3*h*m + 9*a^3*b*c^4*f*g^2*h^2*j + 9*a^3*b*c^4*e*g^2*h^
2*k + 9*a^3*b*c^4*d*f^2*j*k^2 + 9*a^2*b*c^5*d^2*f^2*j*k + 9*a*b^5*c^2*d^2*g
*h*l^2 - 9*a*b^4*c^3*d^2*g*h^2*l - 6*a^2*b^2*c^4*e*f^3*h*l + 3*a^3*b^2*c^3*
f*g*h*j^3 + 3*a^2*b^2*c^4*f^3*g*h*j + 45*a^3*b*c^4*d^2*f*g*m^2 - 27*a^2*b*c
^5*d^2*f^2*g*m + 18*a^3*b*c^4*e^2*f*g*l^2 + 15*a^3*b^3*c^2*e*f*g*l^3 - 12*a
^3*b^2*c^3*d*e*j*k^3 + 9*a^3*b*c^4*d^2*e*h*m^2 + 9*a^3*b*c^4*e*g^2*h*j^2 +
9*a^3*b*c^4*e*f^2*h*k^2 - 9*a^2*b^3*c^3*d*f*h^3*l + 9*a^2*b*c^5*d^2*f^2*h*l
 + 9*a*b^5*c^2*d^2*f*g*m^2 + 9*a*b^3*c^4*d^2*f^2*g*m + 6*a^3*b^3*c^2*d*f*h*
l^3 + 3*a^2*b^4*c^2*d*e*j*k^3 + 18*a^3*b*c^4*e*f*g^2*k^2 + 18*a^2*b*c^5*d^2
*g^2*h*j + 18*a^2*b*c^5*d^2*f*g^2*l + 18*a^2*b*c^5*d^2*e*g^2*m - 12*a^3*b^2
*c^3*d*f*h*k^3 + 9*a^3*b*c^4*e*f*h^2*j^2 + 9*a^3*b*c^4*d*f^2*g*l^2 + 9*a^3*
b*c^4*d*e^2*g*m^2 + 9*a^3*b*c^4*d*g*h^2*j^2 + 9*a^2*b^2*c^4*e*f*g^3*k + 9*a
^2*b^2*c^4*d*g^3*h*j + 9*a^2*b^2*c^4*d*f*g^3*l + 9*a^2*b^2*c^4*d*e*g^3*m +
9*a^2*b*c^5*e^2*f^2*h*j + 9*a^2*b*c^5*e^2*f^2*g*k - 9*a*b^3*c^4*d^2*g^2*h*j
 - 9*a*b^3*c^4*d^2*f*g^2*l - 9*a*b^3*c^4*d^2*e*g^2*m - 3*a^3*b^2*c^3*e*f*g*
k^3 + 3*a^2*b^4*c^2*e*f*g*k^3 + 3*a^2*b^4*c^2*d*f*h*k^3 - 54*a^3*b*c^4*d*e*
f^2*m^2 - 51*a^3*b^3*c^2*d*e*f*m^3 - 27*a^3*b*c^4*d*e*g^2*l^2 + 9*a^3*b*c^4
*d*e*h^2*k^2 + 9*a^2*b*c^5*e^2*f*g^2*j + 9*a^2*b*c^5*d^2*f*h^2*j + 9*a^2*b*
c^5*d^2*e*h^2*k + 9*a^2*b*c^5*d*e^2*g^2*l - 9*a*b^5*c^2*d*e*f^2*m^2 - 9*a*b
^4*c^3*d^2*e*g*l^2 - 9*a*b^2*c^5*d^2*e^2*g*l - 9*a*b^2*c^5*d^2*e^2*f*m - 3*
a^2*b^3*c^3*e*f*g*j^3 - 3*a^2*b^3*c^3*d*f*h*j^3 + 36*a^3*b^2*c^3*d*e*f*l^3
 - 27*a^2*b*c^5*d^2*f*g*j^2 - 18*a^2*b^4*c^2*d*e*f*l^3 - 18*a^2*b*c^5*d*e^2*
h^2*j + 9*a^2*b*c^5*d^2*e*h*j^2 + 9*a^2*b*c^5*d*f^2*g^2*j + 9*a*b^4*c^3*d*e
^2*f*l^2 + 9*a*b^3*c^4*d^2*f*g*j^2 - 9*a*b^2*c^5*d^2*f^2*g*j - 9*a*b^2*c^5*
d^2*e*f^2*l + 3*a^2*b^2*c^4*d*e*h^3*j - 18*a^2*b*c^5*e^2*f*g*h^2 + 18*a^2*b
*c^5*d^2*e*f*k^2 + 15*a^2*b^3*c^3*d*e*f*k^3 + 9*a^2*b*c^5*e*f^2*g^2*h + 9*a
^2*b*c^5*d*e^2*g*j^2 - 9*a*b^3*c^4*d^2*e*f*k^2 + 9*a*b^2*c^5*d^2*e*g^2*j -
9*a*b^2*c^5*d*e^2*f^2*k + 3*a^2*b^2*c^4*e*f*g*h^3 + 18*a^2*b*c^5*d*e*f^2*j^
2 + 9*a^2*b*c^5*d*f^2*g*h^2 - 9*a*b^3*c^4*d*e*f^2*j^2 + 9*a*b^2*c^5*d^2*f*g
^2*h - 3*a^2*b^2*c^4*d*e*f*j^3 + 9*a^2*b*c^5*d*e*g^2*h^2 - 9*a*b^2*c^5*d^2*
e*g*h^2 + 9*a*b^2*c^5*d*e^2*f*h^2 - 36*a^6*c^2*f*j*k*l*m^2 + 36*a^5*c^3*f^2
*j*k*l*m - 36*a^5*c^3*f*h^2*j*l*m + 36*a^5*c^3*e*h*j^2*l*m - 18*a^6*b*c*j^2
*k*l*m^2 + 9*a^6*b*c*j*k^2*l^2*m + 3*a^5*b^2*c*j^3*k*l*m - 36*a^5*c^3*f*g*j
*k^2*m - 36*a^5*c^3*e*f*k^2*l*m + 36*a^5*c^3*d*g*k^2*l*m - 36*a^4*c^4*d^2*g
*k*l*m - 36*a^5*c^3*e*h*j*k*l^2 - 36*a^5*c^3*e*f*j*l^2*m - 36*a^5*c^3*d*f*k
*l^2*m + 36*a^4*c^4*e^2*h*j*k*l + 36*a^4*c^4*e^2*f*j*l*m + 9*a^6*b*c*h*k^2*
l*m^2 - 3*a^4*b^3*c*h^3*k*l*m - 36*a^5*c^3*e*g*h*l^2*m + 36*a^5*c^3*e*f*j*k
*m^2 - 36*a^5*c^3*d*g*j*k*m^2 + 36*a^5*c^3*d*f*j*l*m^2 - 36*a^5*c^3*d*e*k*l
*m^2 + 36*a^4*c^4*e^2*g*h*l*m - 36*a^4*c^4*e*f^2*j*k*m - 36*a^4*c^4*d*f^2*j

*l*m + 9*a^6*b*c*h*j*l^2*m^2 + 9*a^6*b*c*g*k*l^2*m^2 + 9*a^5*b^2*c*g*k^3*l*
m + 3*a^3*b^4*c*g^3*k*l*m + 36*a^5*c^3*f*g*h*j*m^2 + 36*a^5*c^3*e*f*h*l*m^2
 - 36*a^4*c^4*f^2*g*h*j*m - 36*a^4*c^4*e*f^2*h*l*m - 24*a^4*b*c^3*f^3*k*l*m
 - 12*a^5*b*c^2*h*j^3*k*m - 12*a^5*b*c^2*g*j^3*l*m - 3*a^2*b^5*c*f^3*k*l*m
- 36*a^4*c^4*e*g^2*h*k*l - 36*a^4*c^4*e*f*g^2*l*m + 12*a^5*b^2*c*e*k*l^3*m
- 6*a^5*b^2*c*f*j*l^3*m + 3*a^5*b^2*c*h*j*k*l^3 + 48*a^3*b*c^4*d^3*k*l*m +
36*a^4*c^4*e*f*h^2*j*m + 36*a^4*c^4*d*g*h^2*k*l - 36*a^4*c^4*d*f*h^2*k*m -
36*a^4*c^4*d*e*j^2*k*l + 24*a^5*b*c^2*d*k^3*l*m + 21*a*b^5*c^2*d^3*k*l*m -
12*a^5*b*c^2*g*j*k^3*l - 9*a^4*b^3*c*d*k^3*l*m + 6*a^5*b*c^2*f*j*k^3*m + 3*
a^5*b^2*c*g*h*l^3*m - 36*a^4*c^4*e*f*h*j^2*l - 12*a^5*b*c^2*g*h*k^3*m - 3*a
^5*b^2*c*e*j*k*m^3 - 3*a^5*b^2*c*d*j*l*m^3 - 36*a^4*c^4*d*g*h*j*k^2 - 36*a^
4*c^4*d*f*g*k^2*l - 36*a^4*c^4*d*e*h*k^2*l - 36*a^4*c^4*d*e*g*k^2*m + 36*a^
3*c^5*d^2*g*h*j*k + 36*a^3*c^5*d^2*f*g*k*l - 36*a^3*c^5*d^2*f*g*j*m + 36*a^
3*c^5*d^2*e*h*k*l + 36*a^3*c^5*d^2*e*g*k*m - 36*a^3*c^5*d^2*e*f*l*m + 24*a^
5*b^2*c*e*h*l*m^3 - 24*a^3*b*c^4*e^3*j*k*l - 12*a^5*b^2*c*f*h*k*m^3 - 12*a^
5*b^2*c*f*g*l*m^3 - 3*a^5*b^2*c*g*h*j*m^3 - 3*a^4*b^3*c*e*j*k*l^3 - 3*a*b^5
*c^2*e^3*j*k*l + 36*a^4*c^4*d*e*h*j*l^2 + 36*a^4*c^4*d*e*g*k*l^2 - 36*a^3*c
^5*d*e^2*h*j*l - 36*a^3*c^5*d*e^2*g*k*l - 36*a^3*c^5*d*e^2*f*k*m + 24*a^4*b
*c^3*e*h^3*k*m - 24*a^3*b*c^4*e^3*g*l*m - 18*a*b^4*c^3*d^3*j*k*l - 12*a^4*b
*c^3*g*h^3*j*l - 12*a^4*b*c^3*f*h^3*k*l - 12*a^4*b*c^3*d*h^3*l*m + 12*a^3*b
*c^4*e^3*h*k*m + 6*a^4*b*c^3*f*h^3*j*m - 3*a^4*b^3*c*g*h*j*l^3 - 3*a^4*b^3*
c*f*h*k*l^3 - 3*a^4*b^3*c*e*g*l^3*m - 3*a^4*b^3*c*d*h*l^3*m - 3*a*b^5*c^2*e
^3*h*k*m - 3*a*b^5*c^2*e^3*g*l*m + 36*a^4*c^4*e*f*g*h*l^2 - 36*a^4*c^4*d*e*
f*j*m^2 - 36*a^3*c^5*e^2*f*g*h*l - 36*a^3*c^5*d*f^2*g*j*k - 36*a^3*c^5*d*e*
f^2*k*l + 36*a^3*c^5*d*e*f^2*j*m - 18*a*b^4*c^3*d^3*h*k*m - 9*a*b^4*c^3*d^3
*g*l*m + 30*a^5*b*c^2*d*g*k*m^3 - 30*a^4*b^3*c*d*g*k*m^3 - 24*a^5*b*c^2*e*f
*k*m^3 - 24*a^5*b*c^2*d*f*l*m^3 + 24*a^4*b*c^3*e*g*j^3*m + 24*a^4*b*c^3*d*h
*j^3*m + 15*a^4*b^3*c*e*f*k*m^3 + 15*a^4*b^3*c*d*f*l*m^3 + 12*a^5*b*c^2*e*g
*j*m^3 + 12*a^5*b*c^2*d*h*j*m^3 - 12*a^4*b*c^3*f*h*j^3*k - 12*a^4*b*c^3*f*g
*j^3*l + 6*a^4*b^3*c*e*g*j*m^3 + 6*a^4*b^3*c*d*h*j*m^3 + 6*a^4*b*c^3*e*h*j^
3*l + 36*a^3*c^5*d*e*g^2*h*l - 24*a^5*b*c^2*f*g*h*m^3 + 15*a^4*b^3*c*f*g*h*
m^3 - 9*a*b^6*c*d^2*g*j*m^2 - 6*a^3*b^4*c*d*g*k*l^3 - 6*a*b^4*c^3*e^3*f*j*m
 + 3*a^3*b^4*c*e*g*j*l^3 + 3*a^3*b^4*c*e*f*k*l^3 + 3*a^3*b^4*c*d*h*j*l^3 +
3*a^3*b^4*c*d*e*l^3*m + 3*a*b^4*c^3*e^3*h*j*k + 3*a*b^4*c^3*e^3*g*j*l + 3*a
*b^4*c^3*e^3*f*k*l + 3*a*b^4*c^3*d*e^3*l*m - 36*a^3*c^5*d*e*g*h^2*k + 30*a^
2*b*c^5*d^3*f*j*m - 30*a*b^3*c^4*d^3*f*j*m + 24*a^3*b*c^4*d*g^3*j*l - 24*a^
2*b*c^5*d^3*h*j*k - 24*a^2*b*c^5*d^3*f*k*l - 24*a^2*b*c^5*d^3*e*k*m + 15*a*
b^3*c^4*d^3*h*j*k + 15*a*b^3*c^4*d^3*f*k*l + 15*a*b^3*c^4*d^3*e*k*m - 12*a^
3*b*c^4*e*g^3*j*k + 12*a^2*b*c^5*d^3*g*j*l + 6*a*b^3*c^4*d^3*g*j*l + 3*a^3*
b^4*c*f*g*h*l^3 + 3*a*b^4*c^3*e^3*g*h*m + 24*a^3*b*c^4*d*g^3*h*m - 12*a^3*b
*c^4*f*g^3*h*k + 12*a^2*b*c^5*d^3*g*h*m - 9*a^3*b^4*c*d*e*j*m^3 + 6*a^3*b*c
^4*e*g^3*h*l + 6*a*b^3*c^4*d^3*g*h*m + 36*a^3*c^5*d*e*f*g*k^2 - 36*a^2*c^6*
d^2*e*f*g*k - 24*a^4*b*c^3*d*e*j*l^3 - 18*a^3*b^4*c*e*f*g*m^3 - 18*a^3*b^4*
c*d*f*h*m^3 - 3*a^2*b^5*c*d*e*j*l^3 - 3*a*b^3*c^4*d*e^3*j*l - 24*a^4*b*c^3*
e*f*g*l^3 + 24*a^3*b*c^4*d*f*h^3*l + 12*a^4*b*c^3*d*f*h*l^3 - 12*a^3*b*c^4*

e*g*h^3*j - 12*a^3*b*c^4*e*f*h^3*k - 12*a^3*b*c^4*d*e*h^3*m - 12*a*b^2*c^5*
d^3*e*j*k + 6*a^3*b*c^4*d*g*h^3*k - 3*a^2*b^5*c*e*f*g*l^3 - 3*a^2*b^5*c*d*f
*h*l^3 - 3*a*b^3*c^4*e^3*g*h*j - 3*a*b^3*c^4*e^3*f*h*k - 3*a*b^3*c^4*e^3*f*
g*l - 3*a*b^3*c^4*d*e^3*h*m + 24*a*b^2*c^5*d^3*e*h*l - 12*a*b^2*c^5*d^3*f*h
*k - 3*a*b^2*c^5*d^3*g*h*j - 3*a*b^2*c^5*d^3*f*g*l - 3*a*b^2*c^5*d^3*e*g*m
+ 48*a^4*b*c^3*d*e*f*m^3 + 24*a^2*b*c^5*d*e*f^3*m + 21*a^2*b^5*c*d*e*f*m^3
- 12*a^2*b*c^5*e*f^3*g*j - 12*a^2*b*c^5*d*f^3*h*j - 9*a*b^3*c^4*d*e*f^3*m +
6*a^2*b*c^5*d*f^3*g*k + 12*a*b^2*c^5*d*e^3*f*l - 6*a*b^2*c^5*d*e^3*g*k + 3
*a*b^2*c^5*d*e^3*h*j - 24*a^3*b*c^4*d*e*f*k^3 - 12*a^2*b*c^5*d*e*g^3*j - 3*
a*b^5*c^2*d*e*f*k^3 + 3*a*b^2*c^5*e^3*f*g*h - 12*a^2*b*c^5*d*f*g^3*h + 9*a*
b^2*c^5*d*e*f^3*j + 9*a*b*c^6*d^2*e^2*f*j + 3*a*b^4*c^3*d*e*f*j^3 + 9*a*b*c
^6*d^2*e^2*g*h + 9*a*b*c^6*d^2*e*f^2*h - 3*a*b^3*c^4*d*e*f*h^3 - 18*a*b*c^6
*d^2*e*f*g^2 + 9*a*b*c^6*d*e^2*f^2*g + 3*a*b^2*c^5*d*e*f*g^3 - 36*a^4*b^2*c
^2*e^2*k*l^2*m - 9*a^4*b^2*c^2*g^2*j^2*k*m + 45*a^3*b^3*c^2*d^2*k^2*l*m + 3
6*a^4*b^2*c^2*e^2*j*l*m^2 + 9*a^4*b^2*c^2*g^2*j*k^2*l + 9*a^3*b^3*c^2*e^2*j
^2*l*m + 9*a^4*b^2*c^2*g^2*h*k^2*m - 9*a^4*b^2*c^2*f^2*h*l^2*m - 9*a^3*b^3*
c^2*f^2*j^2*k*l - 45*a^3*b^3*c^2*d^2*j*k*m^2 + 36*a^3*b^2*c^3*d^2*j^2*k*m +
18*a^4*b^2*c^2*f^2*h*k*m^2 + 18*a^4*b^2*c^2*f^2*g*l*m^2 - 9*a^4*b^2*c^2*g^
2*h*k*l^2 - 9*a^4*b^2*c^2*f*h^2*k^2*m - 9*a^4*b^2*c^2*f*g^2*l^2*m - 9*a^4*b
^2*c^2*e*j^2*k^2*l - 9*a^4*b^2*c^2*d*j^2*k^2*m - 9*a^3*b^3*c^2*e^2*j*k*l^2
- 9*a^2*b^4*c^2*d^2*j^2*k*m - 36*a^3*b^2*c^3*d^2*j*k^2*l - 27*a^3*b^2*c^3*e
^2*h^2*k*m + 9*a^4*b^2*c^2*g*h^2*j*l^2 + 9*a^4*b^2*c^2*f*h^2*k*l^2 - 9*a^4*
b^2*c^2*f*g^2*k*m^2 - 9*a^4*b^2*c^2*e*g^2*l*m^2 - 9*a^4*b^2*c^2*d*j^2*k*l^2
+ 9*a^4*b^2*c^2*d*h^2*l^2*m - 9*a^3*b^3*c^2*e^2*g*l^2*m + 9*a^2*b^4*c^2*e^
2*h^2*k*m + 9*a^2*b^4*c^2*d^2*j*k^2*l - 45*a^3*b^3*c^2*e^2*h*j*m^2 + 36*a^4
*b^2*c^2*e*h^2*j*m^2 + 36*a^3*b^2*c^3*e^2*h*j^2*m - 36*a^3*b^2*c^3*d^2*h*k^
2*m + 36*a^2*b^3*c^3*d^2*g^2*l*m - 9*a^4*b^2*c^2*f*h*j^2*l^2 - 9*a^4*b^2*c^
2*d*h^2*k*m^2 + 9*a^3*b^3*c^2*f^2*h*j*l^2 + 9*a^3*b^3*c^2*e^2*f*l*m^2 + 9*a
^3*b^3*c^2*e*h^2*j^2*m - 9*a^3*b^2*c^3*f^2*h^2*j*l - 9*a^2*b^4*c^2*e^2*h*j^
2*m + 9*a^2*b^4*c^2*d^2*h*k^2*m + 36*a^3*b^2*c^3*d^2*h*k*l^2 - 27*a^4*b^2*c
^2*e*g*j^2*m^2 - 27*a^4*b^2*c^2*d*h*j^2*m^2 - 9*a^4*b^2*c^2*d*h*k^2*l^2 - 9
*a^3*b^3*c^2*e*f^2*k*m^2 - 9*a^3*b^3*c^2*d*f^2*l*m^2 + 9*a^3*b^2*c^3*f^2*h*
j^2*k + 9*a^3*b^2*c^3*f^2*g*j^2*l - 9*a^3*b^2*c^3*e^2*g*k^2*l - 9*a^3*b^2*c
^3*e^2*f*k^2*m - 9*a^3*b^2*c^3*d^2*f*l^2*m - 9*a^2*b^4*c^2*d^2*h*k*l^2 + 9*
a^2*b^3*c^3*d^2*h^2*k*l - 81*a^3*b^2*c^3*d^2*g*j*m^2 + 54*a^2*b^4*c^2*d^2*g
*j*m^2 - 45*a^3*b^3*c^2*d*g^2*j*m^2 - 45*a^2*b^3*c^3*d^2*g*j^2*m + 36*a^3*b
^2*c^3*d^2*f*k*m^2 + 36*a^3*b^2*c^3*d*g^2*j^2*m + 18*a^3*b^2*c^3*e^2*g*j*l^
2 + 18*a^3*b^2*c^3*e^2*f*k*l^2 + 18*a^3*b^2*c^3*d*e^2*l^2*m - 9*a^4*b^2*c^2
*d*f*k^2*m^2 - 9*a^3*b^3*c^2*f^2*g*h*m^2 - 9*a^3*b^3*c^2*d*h^2*j*l^2 - 9*a^
3*b^2*c^3*f^2*g*j*k^2 - 9*a^3*b^2*c^3*d^2*e*l*m^2 - 9*a^3*b^2*c^3*f*g^2*h^2
*m - 9*a^3*b^2*c^3*e*g^2*j^2*l - 9*a^3*b^2*c^3*e*f^2*k^2*l - 9*a^2*b^4*c^2*
d^2*f*k*m^2 - 9*a^2*b^4*c^2*d*g^2*j^2*m - 9*a^2*b^3*c^3*e^2*h^2*j*k - 9*a^2
*b^2*c^4*d^2*f^2*k*m - 27*a^2*b^2*c^4*d^2*g^2*j*l - 9*a^3*b^3*c^2*f*g*h^2*l
^2 + 9*a^3*b^2*c^3*e*g^2*j*k^2 - 9*a^3*b^2*c^3*e*f^2*j*l^2 - 9*a^3*b^2*c^3*
d*h^2*j^2*k - 9*a^3*b^2*c^3*d*f^2*k*l^2 - 9*a^3*b^2*c^3*d*e^2*k*m^2 - 9*a^2

*b^3*c^3*e^2*g*h^2*m - 9*a^2*b^3*c^3*d^2*h*j*k^2 - 9*a^2*b^3*c^3*d^2*f*k^2*
l - 9*a^2*b^3*c^3*d^2*e*k^2*m + 36*a^3*b^3*c^2*d*e*j^2*m^2 + 36*a^3*b^2*c^3
*e^2*f*h*m^2 - 27*a^2*b^2*c^4*d^2*g^2*h*m + 9*a^3*b^3*c^2*e*f*h^2*m^2 + 9*a
^3*b^2*c^3*f*g^2*h*k^2 - 9*a^2*b^4*c^2*e^2*f*h*m^2 + 9*a^2*b^3*c^3*d^2*e*k*
l^2 - 9*a^2*b^2*c^4*e^2*f^2*h*m - 45*a^2*b^3*c^3*d^2*g*h*l^2 - 36*a^3*b^2*c
^3*e*f^2*g*m^2 + 36*a^3*b^2*c^3*d*g^2*h*l^2 - 36*a^3*b^2*c^3*d*f^2*h*m^2 +
36*a^2*b^2*c^4*d^2*g*h^2*l - 9*a^3*b^2*c^3*e*g*h^2*k^2 + 9*a^2*b^4*c^2*e*f^
2*g*m^2 - 9*a^2*b^4*c^2*d*g^2*h*l^2 + 9*a^2*b^4*c^2*d*f^2*h*m^2 + 9*a^2*b^3
*c^3*e^2*g*h*k^2 + 9*a^2*b^3*c^3*d*g^2*h^2*l - 9*a^2*b^3*c^3*d*e^2*j*l^2 -
9*a^2*b^2*c^4*e^2*g^2*h*k - 9*a^2*b^2*c^4*e^2*f*g^2*m - 9*a^2*b^2*c^4*d^2*f
*j^2*k - 9*a^2*b^2*c^4*d^2*f*h^2*m - 9*a^2*b^2*c^4*d^2*e*j^2*l - 45*a^2*b^3
*c^3*d^2*f*g*m^2 + 36*a^3*b^2*c^3*d*f*g^2*m^2 - 27*a^3*b^2*c^3*d*f*h^2*l^2
+ 18*a^2*b^2*c^4*d^2*e*j*k^2 + 9*a^2*b^4*c^2*d*f*h^2*l^2 - 9*a^2*b^4*c^2*d*
f*g^2*m^2 - 9*a^2*b^3*c^3*e^2*f*g*l^2 + 9*a^2*b^2*c^4*e^2*g*h^2*j + 9*a^2*b
^2*c^4*e^2*f*h^2*k - 9*a^2*b^2*c^4*e*f^2*g^2*l - 9*a^2*b^2*c^4*d*f^2*g^2*m
- 9*a^2*b^2*c^4*d*e^2*j^2*k + 9*a^2*b^2*c^4*d*e^2*h^2*m + 18*a^4*b^2*c^2*f^
2*j^2*m^2 + 18*a^3*b^2*c^3*e^2*h^2*l^2 - 9*a^2*b^4*c^2*e^2*h^2*l^2 + 18*a^2
*b^2*c^4*d^2*g^2*k^2 + 12*a^6*c^2*j^3*k*l*m + 3*a^6*b^2*j*k*l*m^3 - 12*a^6*
c^2*g*k^3*l*m - 12*a^5*c^3*g^3*k*l*m - 24*a^6*c^2*e*k*l^3*m - 24*a^4*c^4*e^
3*k*l*m + 12*a^6*c^2*h*j*k*l^3 + 12*a^6*c^2*f*j*l^3*m + 12*a^5*c^3*h^3*j*k*
l - 3*a^5*b^3*h*j*k*m^3 - 3*a^5*b^3*g*j*l*m^3 - 3*a^5*b^3*f*k*l*m^3 + 12*a^
6*c^2*g*h*l^3*m + 12*a^5*c^3*g*h^3*l*m - 12*a^6*c^2*e*j*k*m^3 - 12*a^6*c^2*
d*j*l*m^3 - 12*a^5*c^3*f*j^3*k*l - 12*a^5*c^3*e*j^3*k*m - 12*a^5*c^3*d*j^3*
l*m - 12*a^4*c^4*f^3*j*k*l + 24*a^6*c^2*f*h*k*m^3 + 24*a^6*c^2*f*g*l*m^3 +
24*a^4*c^4*f^3*h*k*m + 24*a^4*c^4*f^3*g*l*m - 12*a^6*c^2*g*h*j*m^3 - 12*a^6
*c^2*e*h*l*m^3 - 12*a^5*c^3*g*h*j^3*m + 3*b^6*c^2*d^3*j*k*l + 3*a^4*b^4*e*j
*k*m^3 + 3*a^4*b^4*d*j*l*m^3 - 24*a^5*c^3*d*j*k^3*l - 24*a^3*c^5*d^3*j*k*l
- 6*a^4*b^4*e*h*l*m^3 + 3*b^6*c^2*d^3*h*k*m + 3*b^6*c^2*d^3*g*l*m + 3*a^6*b
*c*j^2*l^3*m + 3*a^4*b^4*g*h*j*m^3 + 3*a^4*b^4*f*h*k*m^3 + 3*a^4*b^4*f*g*l*
m^3 - 24*a^5*c^3*d*h*k^3*m - 24*a^3*c^5*d^3*h*k*m + 12*a^5*c^3*g*h*j*k^3 +
12*a^5*c^3*f*g*k^3*l + 12*a^5*c^3*e*h*k^3*l + 12*a^5*c^3*e*g*k^3*m + 12*a^4
*c^4*g^3*h*j*k + 12*a^4*c^4*f*g^3*k*l + 12*a^4*c^4*f*g^3*j*m + 12*a^4*c^4*e
*g^3*k*m + 12*a^4*c^4*d*g^3*l*m + 12*a^3*c^5*d^3*g*l*m + 3*a^6*b*c*j*k^3*m^
2 - 9*a^6*b*c*h^2*l*m^3 - 3*a^5*b*c^2*j^4*k*l + 24*a^5*c^3*e*g*j*l^3 + 24*a
^5*c^3*e*f*k*l^3 + 24*a^5*c^3*d*e*l^3*m + 24*a^3*c^5*e^3*g*j*l + 24*a^3*c^5
*e^3*f*k*l + 24*a^3*c^5*d*e^3*l*m - 12*a^5*c^3*d*h*j*l^3 - 12*a^5*c^3*d*g*k
*l^3 - 12*a^4*c^4*e*h^3*j*k - 12*a^4*c^4*d*h^3*j*l - 12*a^3*c^5*e^3*h*j*k -
12*a^3*c^5*e^3*f*j*m + 9*a^4*b*c^3*g^4*l*m + 6*b^5*c^3*d^3*f*j*m + 6*a^3*b
^5*d*g*k*m^3 - 3*b^5*c^3*d^3*h*j*k - 3*b^5*c^3*d^3*g*j*l - 3*b^5*c^3*d^3*f*
k*l - 3*b^5*c^3*d^3*e*k*m - 3*a^3*b^5*e*g*j*m^3 - 3*a^3*b^5*e*f*k*m^3 - 3*a
^3*b^5*d*h*j*m^3 - 3*a^3*b^5*d*f*l*m^3 - 12*a^5*c^3*f*g*h*l^3 - 12*a^4*c^4*
f*g*h^3*l - 12*a^4*c^4*e*g*h^3*m - 12*a^3*c^5*e^3*g*h*m - 9*a^6*b*c*g*k^2*m
^3 - 3*b^5*c^3*d^3*g*h*m + 3*a^6*b*c*f*l^3*m^2 - 3*a^3*b^5*f*g*h*m^3 + 12*a
^5*c^3*d*e*j*m^3 + 12*a^4*c^4*e*f*j^3*k + 12*a^4*c^4*d*g*j^3*k + 12*a^4*c^4
*d*f*j^3*l + 12*a^4*c^4*d*e*j^3*m + 12*a^3*c^5*e*f^3*j*k + 12*a^3*c^5*d*f^3

*j*l - 9*a^6*b*c*e*l^2*m^3 - 24*a^5*c^3*e*f*g*m^3 - 24*a^5*c^3*d*f*h*m^3 -
24*a^3*c^5*e*f^3*g*m - 24*a^3*c^5*d*f^3*h*m - 15*a^2*b*c^5*d^4*l*m + 15*a*b
^3*c^4*d^4*l*m + 12*a^4*c^4*f*g*h*j^3 + 12*a^3*c^5*f^3*g*h*j + 12*a^3*c^5*e
*f^3*h*l + 9*a^3*b*c^4*f^4*k*l - 9*a^3*b*c^4*f^4*j*m + 3*b^4*c^4*d^3*e*j*k
+ 3*a^5*b^2*c*g*j*l^4 + 3*a^5*b^2*c*f*k*l^4 + 3*a^5*b^2*c*d*l^4*m - 3*a^5*b
*c^2*h*j*k^4 - 3*a^5*b*c^2*f*k^4*l - 3*a^5*b*c^2*e*k^4*m - 3*a^4*b*c^3*h^4*
j*k + 3*a^2*b^6*d*e*j*m^3 + 3*a*b^4*c^3*e^4*k*m + 24*a^4*c^4*d*e*j*k^3 + 24
*a^2*c^6*d^3*e*j*k - 6*b^4*c^4*d^3*e*h*l + 3*b^4*c^4*d^3*g*h*j + 3*b^4*c^4*
d^3*f*h*k + 3*b^4*c^4*d^3*f*g*l + 3*b^4*c^4*d^3*e*g*m - 3*a^4*b*c^3*g*h^4*m
 + 3*a^2*b^6*e*f*g*m^3 + 3*a^2*b^6*d*f*h*m^3 - 3*a*b^6*c*e^3*j*m^2 + 24*a^4
*c^4*d*f*h*k^3 + 24*a^2*c^6*d^3*f*h*k - 12*a^4*c^4*e*f*g*k^3 - 12*a^3*c^5*e
*f*g^3*k - 12*a^3*c^5*d*g^3*h*j - 12*a^3*c^5*d*f*g^3*l - 12*a^3*c^5*d*e*g^3
*m - 12*a^2*c^6*d^3*g*h*j - 12*a^2*c^6*d^3*f*g*l - 12*a^2*c^6*d^3*e*h*l - 1
2*a^2*c^6*d^3*e*g*m - 12*a*b^2*c^5*d^4*j*l + 9*a^5*b*c^2*d*j*l^4 + 9*a^2*b*
c^5*e^4*j*k - 3*a^4*b^3*c*d*j*l^4 - 3*a^4*b*c^3*e*j^4*k - 3*a^4*b*c^3*d*j^4
*l - 3*a*b^3*c^4*e^4*j*k - 24*a^4*c^4*d*e*f*l^3 - 24*a^2*c^6*d*e^3*f*l - 12
*a^5*b^2*c*e*g*m^4 - 12*a^5*b^2*c*d*h*m^4 + 12*a^3*c^5*d*e*h^3*j + 12*a^2*c
^6*d*e^3*h*j + 12*a^2*c^6*d*e^3*g*k - 12*a*b^2*c^5*d^4*h*m + 9*a^5*b*c^2*f*
g*l^4 - 9*a^5*b*c^2*e*h*l^4 - 9*a^2*b*c^5*e^4*h*l + 9*a^2*b*c^5*e^4*g*m + 6
*a^4*b^3*c*e*h*l^4 + 6*a*b^3*c^4*e^4*h*l - 3*b^3*c^5*d^3*e*g*j - 3*b^3*c^5*
d^3*e*f*k - 3*a^4*b^3*c*f*g*l^4 - 3*a^4*b*c^3*g*h*j^4 - 3*a^3*b*c^4*g^4*h*j
 - 3*a^3*b*c^4*f*g^4*l - 3*a^3*b*c^4*e*g^4*m - 3*a*b^3*c^4*e^4*g*m + 12*a^3
*c^5*e*f*g*h^3 + 12*a^2*c^6*e^3*f*g*h - 3*b^3*c^5*d^3*f*g*h - 12*a^3*c^5*d*
e*f*j^3 - 12*a^2*c^6*d*e*f^3*j - 3*a*b^6*c*d^2*g*l^3 - 15*a^5*b*c^2*d*e*m^4
 + 15*a^4*b^3*c*d*e*m^4 + 9*a^4*b*c^3*e*f*k^4 - 9*a^4*b*c^3*d*g*k^4 + 3*a^3
*b^4*c*d*f*l^4 - 3*a^3*b*c^4*d*h^4*j - 3*a^2*b*c^5*e*f^4*k - 3*a^2*b*c^5*d*
f^4*l + 3*a*b^2*c^5*e^4*g*j + 3*a*b^2*c^5*e^4*f*k + 3*a*b^2*c^5*d*e^4*m - 9
*a*b*c^6*d^3*e^2*l + 3*b^2*c^6*d^3*e*f*g - 3*a^3*b*c^4*f*g*h^4 - 3*a^2*b*c^
5*f^4*g*h + 12*a^2*c^6*d*e*f*g^3 - 9*a*b*c^6*d^3*f^2*j + 3*a*b*c^6*d^2*e^3*
k + 9*a^3*b*c^4*d*e*j^4 - 3*a^2*b*c^5*e*f*g^4 - 9*a*b*c^6*d^3*e*h^2 + 3*a*b
*c^6*d^2*f^3*g + 3*a*b*c^6*d*e^3*g^2 - 3*a^4*b^2*c^2*h^3*j^2*m + 12*a^4*b^2
*c^2*g^3*j*m^2 - 3*a^4*b^2*c^2*f^2*k^3*m + 3*a^3*b^3*c^2*g^3*j^2*m - 9*a^3*
b^4*c*f^2*j^2*m^2 + 9*a^3*b^3*c^2*f^2*j^3*m - 6*a^3*b^3*c^2*f^3*j*m^2 - 6*a
^3*b^2*c^3*f^3*j^2*m - 3*a^2*b^4*c^2*f^3*j^2*m - 27*a^4*b^2*c^2*d^2*k*m^3 -
 27*a^3*b^2*c^3*e^3*j*m^2 + 18*a^2*b^4*c^2*e^3*j*m^2 - 15*a^2*b^3*c^3*e^3*j
^2*m + 12*a^4*b^2*c^2*f^2*j*l^3 + 3*a^3*b^3*c^2*e^2*k^3*l + 42*a^2*b^3*c^3*
d^3*j*m^2 - 27*a^2*b^2*c^4*d^3*j^2*m - 15*a^3*b^3*c^2*d^2*k*l^3 - 3*a^4*b^2
*c^2*f*j^2*k^3 - 3*a^4*b^2*c^2*f*h^3*m^2 + 3*a^3*b^3*c^2*g^3*h*l^2 + 3*a^3*
b^3*c^2*f^2*j*k^3 - 3*a^3*b^2*c^3*g^3*h^2*l - 3*a^3*b^2*c^3*e^2*j^3*l - 27*
a^4*b^2*c^2*e^2*h*m^3 + 12*a^3*b^2*c^3*f^3*h*l^2 + 3*a^3*b^3*c^2*f*g^3*m^2
- 3*a^2*b^4*c^2*f^3*h*l^2 + 3*a^2*b^3*c^3*f^3*h^2*l + 9*a^3*b^3*c^2*e*h^3*l
^2 + 9*a^2*b^3*c^3*e^2*h^3*l - 6*a^4*b^2*c^2*e*h^2*l^3 - 6*a^3*b^3*c^2*e^2*
h*l^3 - 6*a^2*b^3*c^3*e^3*h*l^2 - 6*a^2*b^2*c^4*e^3*h^2*l + 3*a^2*b^3*c^3*d
^2*j^3*k + 42*a^3*b^3*c^2*d^2*g*m^3 - 27*a^4*b^2*c^2*d*g^2*m^3 - 27*a^2*b^2
*c^4*d^3*h*l^2 - 15*a^2*b^3*c^3*e^3*f*m^2 + 12*a^3*b^2*c^3*e^2*h*k^3 + 3*a^

3*b^3*c^2*e*h^2*k^3 - 3*a^3*b^2*c^3*e*g^3*l^2 - 3*a^2*b^4*c^2*e^2*h*k^3 + 3
*a^2*b^3*c^3*f^3*g*k^2 - 3*a^2*b^2*c^4*f^3*g^2*k - 27*a^3*b^2*c^3*d^2*g*l^3
 - 27*a^2*b^2*c^4*d^3*f*m^2 + 18*a^2*b^4*c^2*d^2*g*l^3 - 15*a^3*b^3*c^2*d*g
^2*l^3 + 12*a^2*b^2*c^4*e^3*g*k^2 - 3*a^3*b^2*c^3*e*h^2*j^3 + 3*a^2*b^3*c^3
*e^2*h*j^3 + 3*a^2*b^3*c^3*e*f^3*l^2 - 3*a^2*b^2*c^4*d^2*h^3*k + 9*a^2*b^3*
c^3*d*g^3*k^2 - 9*a*b^4*c^3*d^2*g^2*k^2 - 6*a^3*b^2*c^3*d*g^2*k^3 - 6*a^2*b
^3*c^3*d^2*g*k^3 - 3*a^2*b^4*c^2*d*g^2*k^3 + 12*a^2*b^2*c^4*d^2*g*j^3 + 3*a
^2*b^3*c^3*d*g^2*j^3 - 3*a^2*b^2*c^4*d*f^3*k^2 - 3*a^2*b^2*c^4*d*g^2*h^3 +
12*a^7*c*j*k*l*m^3 - 3*b^7*c*d^3*k*l*m - 3*a^6*b*c*k^4*l*m - 3*a^6*b*c*j*k*
l^4 - 3*a^6*b*c*g*l^4*m - 9*a^6*b*c*f*j*m^4 + 9*a^6*b*c*e*k*m^4 + 9*a^6*b*c
*d*l*m^4 + 9*a^6*b*c*g*h*m^4 - 3*a*b^7*d*e*f*m^3 + 9*a*b*c^6*d^4*h*j - 9*a*
b*c^6*d^4*g*k + 9*a*b*c^6*d^4*f*l + 9*a*b*c^6*d^4*e*m + 12*a*c^7*d^3*e*f*g
 - 3*a*b*c^6*d*e^4*j - 3*a*b*c^6*e^4*f*g - 3*a*b*c^6*d*e*f^4 + 18*a^6*c^2*h^
2*j*l*m^2 - 18*a^6*c^2*h*j^2*l^2*m + 18*a^6*c^2*f*k^2*l^2*m + 36*a^5*c^3*e^
2*k*l^2*m + 18*a^6*c^2*g*j*k^2*m^2 + 18*a^6*c^2*e*k^2*l*m^2 + 18*a^5*c^3*g^
2*j^2*k*m + 18*a^6*c^2*e*j*l^2*m^2 + 18*a^6*c^2*d*k*l^2*m^2 - 18*a^5*c^3*e^
2*j*l*m^2 - 18*a^6*c^2*f*h*l^2*m^2 + 18*a^5*c^3*f^2*h*l^2*m - 36*a^5*c^3*f^
2*h*k*m^2 - 36*a^5*c^3*f^2*g*l*m^2 + 18*a^5*c^3*g^2*h*k*l^2 - 18*a^5*c^3*g*
h^2*k^2*l + 18*a^5*c^3*f*h^2*k^2*m + 18*a^5*c^3*f*g^2*l^2*m + 18*a^5*c^3*e*
j^2*k^2*l + 18*a^5*c^3*d*j^2*k^2*m - 18*a^4*c^4*d^2*j^2*k*m + 36*a^4*c^4*d^
2*j*k^2*l + 18*a^5*c^3*f*g^2*k*m^2 + 18*a^5*c^3*e*g^2*l*m^2 + 18*a^5*c^3*d*
j^2*k*l^2 - 18*a^4*c^4*f^2*g^2*k*m + 36*a^4*c^4*d^2*h*k^2*m + 18*a^5*c^3*f*
h*j^2*l^2 - 18*a^5*c^3*e*h^2*j*m^2 + 18*a^5*c^3*d*h^2*k*m^2 + 18*a^4*c^4*f^
2*h^2*j*l - 18*a^4*c^4*e^2*h*j^2*m - 18*a^5*c^3*e*g*k^2*l^2 + 18*a^5*c^3*d*
h*k^2*l^2 + 18*a^4*c^4*e^2*g*k^2*l + 18*a^4*c^4*e^2*f*k^2*m - 18*a^4*c^4*d^
2*h*k*l^2 + 18*a^4*c^4*d^2*f*l^2*m - 36*a^4*c^4*e^2*g*j*l^2 - 36*a^4*c^4*e^
2*f*k*l^2 - 36*a^4*c^4*d*e^2*l^2*m + 18*a^5*c^3*d*f*k^2*m^2 + 18*a^4*c^4*f^
2*g*j*k^2 + 18*a^4*c^4*d^2*g*j*m^2 - 18*a^4*c^4*d^2*f*k*m^2 + 18*a^4*c^4*d^
2*e*l*m^2 - 18*a^4*c^4*f*g^2*j^2*k + 18*a^4*c^4*f*g^2*h^2*m + 18*a^4*c^4*e*
g^2*j^2*l + 18*a^4*c^4*e*f^2*k^2*l - 18*a^4*c^4*d*g^2*j^2*m - 18*a^4*c^4*d*
f^2*k^2*m + 18*a^3*c^5*d^2*f^2*k*m + 3*a^4*b^2*c^2*h^4*k*m - 3*a^3*b^3*c^2*
g^4*l*m + 18*a^4*c^4*e*f^2*j*l^2 + 18*a^4*c^4*d*h^2*j^2*k + 18*a^4*c^4*d*f^
2*k*l^2 + 18*a^4*c^4*d*e^2*k*m^2 - 18*a^3*c^5*e^2*f^2*j*l + 12*a^5*b^2*c*g^
2*k*m^3 - 9*a^5*b*c^2*h^3*j*m^2 - 9*a^5*b*c^2*f^2*l^3*m + 3*a^5*b*c^2*h^2*k
^3*l + 3*a^4*b^3*c*h^3*j*m^2 + 3*a^4*b^3*c*f^2*l^3*m - 18*a^4*c^4*e^2*f*h*m
^2 + 18*a^3*c^5*e^2*f^2*h*m + 15*a^5*b*c^2*e^2*l*m^3 - 15*a^4*b^3*c*e^2*l*m
^3 - 9*a^5*b*c^2*g^2*k*l^3 - 9*a^4*b*c^3*g^3*j^2*m - 3*a^5*b^2*c*g*k^2*l^3
+ 3*a^5*b*c^2*h*j^3*l^2 + 3*a^4*b^3*c*g^2*k*l^3 - 3*a^3*b^4*c*g^3*j*m^2 + 3
6*a^4*c^4*e*f^2*g*m^2 + 36*a^4*c^4*d*f^2*h*m^2 + 18*a^4*c^4*e*g*h^2*k^2 - 1
8*a^4*c^4*d*g^2*h*l^2 - 18*a^4*c^4*d*f*j^2*k^2 + 18*a^3*c^5*e^2*g^2*h*k + 1
8*a^3*c^5*e^2*f*g^2*m - 18*a^3*c^5*d^2*g*h^2*l + 18*a^3*c^5*d^2*f*j^2*k + 1
8*a^3*c^5*d^2*f*h^2*m + 18*a^3*c^5*d^2*e*j^2*l - 12*a^2*b^2*c^4*e^4*k*m + 9
*a^4*b^3*c*f*j^3*m^2 - 9*a^4*b^2*c^2*f*j^4*m - 6*a^5*b^2*c*f*j^2*m^3 + 6*a^
5*b*c^2*f^2*j*m^3 - 6*a^5*b*c^2*f*j^3*m^2 - 6*a^4*b^3*c*f^2*j*m^3 + 6*a^4*b
*c^3*f^3*j*m^2 - 6*a^4*b*c^3*f^2*j^3*m + 6*a^2*b^3*c^3*f^4*j*m + 3*a^3*b^2*

c^3*g^4*j*l + 3*a^2*b^5*c*f^3*j*m^2 - 3*a^2*b^3*c^3*f^4*k*l - 36*a^3*c^5*d^
2*e*j*k^2 - 18*a^4*c^4*d*f*g^2*m^2 + 18*a^3*c^5*e*f^2*g^2*l + 18*a^3*c^5*d*
f^2*g^2*m + 18*a^3*c^5*d*e^2*j^2*k + 18*a^3*b^4*c*d^2*k*m^3 + 15*a^3*b*c^4*
e^3*j^2*m + 12*a^5*b^2*c*d*k^2*m^3 - 9*a^5*b*c^2*f*j^2*l^3 - 9*a^4*b*c^3*e^
2*k^3*l + 3*a^5*b*c^2*e*k^3*l^2 + 3*a^4*b^3*c*f*j^2*l^3 + 3*a^4*b*c^3*g^2*j
^3*k - 3*a^3*b^4*c*f^2*j*l^3 + 3*a^3*b^2*c^3*g^4*h*m + 3*a*b^5*c^2*e^3*j^2*
m - 36*a^3*c^5*d^2*f*h*k^2 - 21*a^3*b*c^4*d^3*j*m^2 - 21*a*b^5*c^2*d^3*j*m^
2 + 18*a^3*c^5*e^2*f*h*j^2 - 18*a^3*c^5*e*f^2*h^2*j + 18*a^3*c^5*d*f^2*h^2*
k + 18*a*b^4*c^3*d^3*j^2*m + 15*a^4*b*c^3*d^2*k*l^3 - 9*a^5*b*c^2*d*k^2*l^3
 - 9*a^4*b*c^3*g^3*h*l^2 - 9*a^4*b*c^3*f^2*j*k^3 + 3*a^4*b^3*c*d*k^2*l^3 +
3*a^2*b^5*c*d^2*k*l^3 - 18*a^3*c^5*d^2*e*g*l^2 + 18*a^3*c^5*d*e^2*h*k^2 + 1
8*a^3*b^4*c*e^2*h*m^3 - 18*a^2*c^6*d^2*e^2*h*k + 18*a^2*c^6*d^2*e^2*g*l + 1
8*a^2*c^6*d^2*e^2*f*m + 15*a^5*b*c^2*e*h^2*m^3 - 15*a^4*b^3*c*e*h^2*m^3 - 9
*a^4*b*c^3*f*g^3*m^2 - 9*a^3*b*c^4*f^3*h^2*l + 3*a^4*b^2*c^2*e*j*k^4 + 3*a^
4*b*c^3*g*h^3*k^2 + 3*a^3*b*c^4*f^2*g^3*m + 36*a^3*c^5*d*e^2*f*l^2 + 18*a^3
*c^5*d*f*g^2*j^2 + 18*a^2*c^6*d^2*f^2*g*j + 18*a^2*c^6*d^2*e*f^2*l - 9*a^3*
b^2*c^3*e*h^4*l - 9*a^3*b*c^4*d^2*j^3*k + 6*a^4*b*c^3*e^2*h*l^3 - 6*a^4*b*c
^3*e*h^3*l^2 + 6*a^3*b*c^4*e^3*h*l^2 - 6*a^3*b*c^4*e^2*h^3*l + 3*a^4*b^2*c^
2*f*h*k^4 + 3*a^4*b*c^3*d*j^3*k^2 - 3*a^3*b^4*c*e*h^2*l^3 + 3*a^2*b^5*c*e^2
*h*l^3 + 3*a^2*b^2*c^4*f^4*h*k + 3*a^2*b^2*c^4*f^4*g*l + 3*a*b^5*c^2*e^3*h*
l^2 - 3*a*b^4*c^3*e^3*h^2*l - 21*a^4*b*c^3*d^2*g*m^3 - 21*a^2*b^5*c*d^2*g*m
^3 + 18*a^3*b^4*c*d*g^2*m^3 + 18*a^2*c^6*d*e^2*f^2*k + 18*a*b^4*c^3*d^3*h*l
^2 + 15*a^3*b*c^4*e^3*f*m^2 + 15*a^2*b*c^5*d^3*h^2*l - 15*a*b^3*c^4*d^3*h^2
*l - 9*a^4*b*c^3*e*h^2*k^3 - 9*a^3*b*c^4*f^3*g*k^2 - 9*a^2*b*c^5*e^3*f^2*m
+ 3*a^3*b*c^4*f^2*h^3*j + 3*a*b^5*c^2*e^3*f*m^2 + 3*a*b^3*c^4*e^3*f^2*m + 1
8*a*b^4*c^3*d^3*f*m^2 + 15*a^4*b*c^3*d*g^2*l^3 + 12*a*b^2*c^5*d^3*f^2*m - 9
*a^3*b*c^4*e^2*h*j^3 - 9*a^3*b*c^4*e*f^3*l^2 - 9*a^2*b*c^5*e^3*g^2*k + 3*a^
3*b*c^4*f*g^3*j^2 + 3*a^2*b^5*c*d*g^2*l^3 + 3*a^2*b*c^5*e^2*f^3*l - 3*a*b^4
*c^3*e^3*g*k^2 + 3*a*b^3*c^4*e^3*g^2*k + 18*a^2*c^6*d^2*e*g*h^2 - 18*a^2*c^
6*d*e^2*g^2*h - 12*a^4*b^2*c^2*d*f*l^4 - 9*a^2*b^2*c^4*d*g^4*k + 9*a*b^3*c^
4*d^2*g^3*k + 6*a^3*b^3*c^2*d*g*k^4 + 6*a^3*b*c^4*d^2*g*k^3 - 6*a^3*b*c^4*d
*g^3*k^2 + 6*a^2*b*c^5*d^3*g*k^2 - 6*a^2*b*c^5*d^2*g^3*k - 6*a*b^3*c^4*d^3*
g*k^2 - 6*a*b^2*c^5*d^3*g^2*k - 3*a^3*b^3*c^2*e*f*k^4 + 3*a^3*b^2*c^3*e*g*j
^4 + 3*a^3*b^2*c^3*d*h*j^4 + 3*a*b^5*c^2*d^2*g*k^3 + 15*a^2*b*c^5*d^3*e*l^2
 - 15*a*b^3*c^4*d^3*e*l^2 - 9*a^3*b*c^4*d*g^2*j^3 - 9*a^2*b*c^5*e^3*f*j^2 -
 3*a*b^4*c^3*d^2*g*j^3 + 3*a*b^3*c^4*e^3*f*j^2 - 3*a*b^2*c^5*e^3*f^2*j + 12
*a*b^2*c^5*d^3*f*j^2 - 9*a^2*b*c^5*d*e^3*k^2 + 3*a^2*b*c^5*e^2*g^3*h + 3*a*
b^3*c^4*d*e^3*k^2 - 9*a^2*b*c^5*d^2*g*h^3 - 3*a^2*b^3*c^3*d*e*j^4 + 3*a^2*b
*c^5*e*f^3*h^2 + 3*a*b^3*c^4*d^2*g*h^3 + 3*a^2*b^2*c^4*d*f*h^4 - 9*a^7*c*k^
2*l^2*m^2 - 6*a^6*c^2*j^2*k^3*m - 3*a^6*b^2*h*l^2*m^3 + 3*a^5*b^3*h^2*l*m^3
 - 6*a^6*c^2*g^2*k*m^3 - 6*a^6*c^2*h*k^3*l^2 + 6*a^5*c^3*h^3*j^2*m + 6*a^6*
c^2*g*k^2*l^3 - 6*a^6*c^2*f*k^3*m^2 - 6*a^5*c^3*h^2*j^3*l - 6*a^5*c^3*g^3*j
*m^2 + 6*a^5*c^3*f^2*k^3*m + 3*a^5*b^3*g*k^2*m^3 - 3*a^4*b^4*g^2*k*m^3 + 12
*a^6*c^2*f*j^2*m^3 + 12*a^4*c^4*f^3*j^2*m + 3*a^5*b^3*e*l^2*m^3 + 3*a^3*b^5
*e^2*l*m^3 - 6*a^6*c^2*d*k^2*m^3 - 6*a^5*c^3*f^2*j*l^3 + 6*a^5*c^3*d^2*k*m^

3 - 6*a^5*c^3*g*j^3*k^2 + 6*a^4*c^4*e^3*j*m^2 - 3*b^6*c^2*d^3*j^2*m - 3*a^4
*b^4*f*j^2*m^3 + 3*a^3*b^5*f^2*j*m^3 + 6*a^5*c^3*f*j^2*k^3 + 6*a^5*c^3*f*h^
3*m^2 - 6*a^5*c^3*e*j^3*l^2 + 6*a^4*c^4*g^3*h^2*l - 6*a^4*c^4*f^2*h^3*m + 6
*a^4*c^4*e^2*j^3*l + 6*a^3*c^5*d^3*j^2*m - 3*a^4*b^4*d*k^2*m^3 - 3*a^2*b^6*
d^2*k*m^3 + 6*a^5*c^3*e^2*h*m^3 - 6*a^4*c^4*g^2*h^3*k - 6*a^4*c^4*f^3*h*l^2
 + 12*a^5*c^3*e*h^2*l^3 + 12*a^3*c^5*e^3*h^2*l - 3*b^6*c^2*d^3*h*l^2 + 3*b^
5*c^3*d^3*h^2*l - 3*a^5*b^2*c*j^4*m^2 + 3*a^3*b^5*e*h^2*m^3 - 3*a^2*b^6*e^2
*h*m^3 + 6*a^5*c^3*d*g^2*m^3 - 6*a^4*c^4*e^2*h*k^3 - 6*a^4*c^4*f*h^3*j^2 +
6*a^4*c^4*e*g^3*l^2 + 6*a^3*c^5*f^3*g^2*k - 6*a^3*c^5*e^2*g^3*l + 6*a^3*c^5
*d^3*h*l^2 - 3*b^6*c^2*d^3*f*m^2 - 3*b^4*c^4*d^3*f^2*m + 6*a^4*c^4*d^2*g*l^
3 + 6*a^4*c^4*e*h^2*j^3 - 6*a^4*c^4*d*h^3*k^2 - 6*a^3*c^5*f^2*g^3*j - 6*a^3
*c^5*e^3*g*k^2 + 6*a^3*c^5*d^3*f*m^2 + 6*a^3*c^5*d^2*h^3*k - 6*a^2*c^6*d^3*
f^2*m + 4*a^5*b^2*c*h^3*m^3 + 3*b^5*c^3*d^3*g*k^2 - 3*b^4*c^4*d^3*g^2*k - 3
*a^2*b^6*d*g^2*m^3 + a^5*b*c^2*j^3*k^3 + 12*a^4*c^4*d*g^2*k^3 + 12*a^2*c^6*
d^3*g^2*k + 6*a^5*b*c^2*h^3*l^3 + 5*a^5*b*c^2*g^3*m^3 - 5*a^4*b^3*c*g^3*m^3
 + 3*b^5*c^3*d^3*e*l^2 + 3*b^3*c^5*d^3*e^2*l - 3*a^5*b^2*c*h^2*l^4 + a^4*b^
3*c*h^3*l^3 + 12*a^5*b^2*c*f^2*m^4 - 6*a^3*c^5*d^2*g*j^3 + 6*a^3*c^5*d*f^3*
k^2 + 6*a^3*b^4*c*f^3*m^3 + 6*a^2*c^6*e^3*f^2*j - 6*a^2*c^6*d^2*f^3*k - 3*b
^4*c^4*d^3*f*j^2 + 3*b^3*c^5*d^3*f^2*j - 3*a^2*b^2*c^4*f^5*m - 7*a^4*b*c^3*
e^3*m^3 - 7*a^2*b^5*c*e^3*m^3 + 6*a^4*b*c^3*g^3*k^3 - 6*a^3*c^5*e*g^3*h^2 -
 6*a^2*c^6*d^3*f*j^2 + 5*a^4*b*c^3*f^3*l^3 + a^4*b*c^3*h^3*j^3 + a^2*b^5*c*
f^3*l^3 + 6*a^3*c^5*d*g^2*h^3 - 6*a^2*c^6*e^2*f^3*h - 3*a^3*b^4*c*e^2*l^4 -
 3*a*b^4*c^3*e^4*l^2 - 7*a^3*b*c^4*d^3*l^3 - 7*a*b^5*c^2*d^3*l^3 + 6*a^3*b*
c^4*f^3*j^3 + 5*a^3*b*c^4*e^3*k^3 + 3*b^3*c^5*d^3*e*h^2 - 3*b^2*c^6*d^3*e^2
*h + a*b^5*c^2*e^3*k^3 + 12*a*b^2*c^5*d^4*k^2 - 6*a^2*c^6*d*f^3*g^2 + 6*a*b
^4*c^3*d^3*k^3 - 3*a^4*b^2*c^2*d*k^5 + a^3*b*c^4*g^3*h^3 + 5*a^2*b*c^5*d^3*
j^3 - 5*a*b^3*c^4*d^3*j^3 - 9*a*c^7*d^2*e^2*f^2 + 6*a^2*b*c^5*e^3*h^3 - 3*a
*b^2*c^5*e^4*h^2 + a^2*b*c^5*f^3*g^3 + a*b^3*c^4*e^3*h^3 + 4*a*b^2*c^5*d^3*
h^3 - 3*a*b^2*c^5*d^2*g^4 - 6*a^7*c*j*l^3*m^2 + 6*a^7*c*h*l^2*m^3 + 6*a^6*c
^2*j*k^4*l + 6*a^6*c^2*h*k^4*m - 6*a^5*c^3*h^4*k*m + 3*a^6*b^2*h*k*m^4 + 3*
a^6*b^2*g*l*m^4 - 3*b^5*c^3*d^4*l*m - 6*a^6*c^2*g*j*l^4 - 6*a^6*c^2*f*k*l^4
 - 6*a^6*c^2*d*l^4*m + 6*a^5*c^3*h*j^4*k + 6*a^5*c^3*g*j^4*l + 6*a^5*c^3*f*
j^4*m - 6*a^4*c^4*g^4*j*l + 6*a^3*c^5*e^4*k*m + 6*a^5*b^3*f*j*m^4 - 6*a^4*c
^4*g^4*h*m + 3*b^7*c*d^3*j*m^2 - 3*a^5*b^3*e*k*m^4 - 3*a^5*b^3*d*l*m^4 + 3*
b^4*c^4*d^4*j*l - 3*a^5*b^3*g*h*m^4 - 6*a^5*c^3*e*j*k^4 + 6*a^2*c^6*d^4*j*l
 + 3*b^4*c^4*d^4*h*m + 6*a^6*c^2*e*g*m^4 + 6*a^6*c^2*d*h*m^4 + 6*a^6*b*c*j^
3*m^3 - 6*a^5*c^3*f*h*k^4 + 6*a^4*c^4*g*h^4*j + 6*a^4*c^4*f*h^4*k + 6*a^4*c
^4*e*h^4*l + 6*a^4*c^4*d*h^4*m - 6*a^3*c^5*f^4*h*k - 6*a^3*c^5*f^4*g*l + 6*
a^2*c^6*d^4*h*m + 3*a^5*b*c^2*j^5*m + a^6*b*c*k^3*l^3 + 3*a^4*b^4*e*g*m^4 +
 3*a^4*b^4*d*h*m^4 + 6*b^3*c^5*d^4*g*k - 3*b^3*c^5*d^4*h*j - 3*b^3*c^5*d^4*
f*l - 3*b^3*c^5*d^4*e*m + 3*a*b^7*d^2*g*m^3 + 6*a^5*c^3*d*f*l^4 - 6*a^4*c^4
*e*g*j^4 - 6*a^4*c^4*d*h*j^4 + 6*a^3*c^5*e*g^4*j + 6*a^3*c^5*d*g^4*k - 6*a^
2*c^6*e^4*g*j - 6*a^2*c^6*e^4*f*k - 6*a^2*c^6*d*e^4*m + 3*a^4*b*c^3*h^5*l +
 6*a^3*c^5*f*g^4*h - 3*a^3*b^5*d*e*m^4 + 3*b^2*c^6*d^4*e*j + 3*a^5*b*c^2*g*
k^5 + 3*a^3*b*c^4*g^5*k + 8*a*b^6*c*d^3*m^3 + 3*b^2*c^6*d^4*f*h - 3*a^5*b^2

*c*e*l^5 - 3*a*b^2*c^5*e^5*l - 6*a^3*c^5*d*f*h^4 + 6*a^2*c^6*e*f^4*g + 6*a^
2*c^6*d*f^4*h + 3*a^4*b*c^3*f*j^5 + 3*a^2*b*c^5*f^5*j + 6*a*c^7*d^3*e^2*h -
 6*a*c^7*d^2*e^3*g + 3*a^3*b*c^4*e*h^5 + 6*a*b*c^6*d^3*g^3 + 3*a^2*b*c^5*d*
g^5 + a*b*c^6*e^3*f^3 - 9*a^6*c^2*j^2*k^2*l^2 - 9*a^6*c^2*h^2*k^2*m^2 - 9*a
^6*c^2*g^2*l^2*m^2 - 18*a^5*c^3*f^2*j^2*m^2 - 9*a^5*c^3*h^2*j^2*k^2 - 9*a^5
*c^3*g^2*j^2*l^2 - 9*a^5*c^3*f^2*k^2*l^2 - 9*a^5*c^3*e^2*k^2*m^2 - 9*a^5*c^
3*d^2*l^2*m^2 - 9*a^5*c^3*g^2*h^2*m^2 - 9*a^4*c^4*e^2*j^2*k^2 - 9*a^4*c^4*d
^2*j^2*l^2 - 18*a^4*c^4*e^2*h^2*l^2 - 9*a^4*c^4*g^2*h^2*j^2 - 9*a^4*c^4*f^2
*h^2*k^2 - 9*a^4*c^4*f^2*g^2*l^2 - 9*a^4*c^4*e^2*g^2*m^2 - 9*a^4*c^4*d^2*h^
2*m^2 - 18*a^3*c^5*d^2*g^2*k^2 - 9*a^3*c^5*e^2*g^2*j^2 - 9*a^3*c^5*e^2*f^2*
k^2 - 9*a^3*c^5*d^2*h^2*j^2 - 9*a^3*c^5*d^2*f^2*l^2 - 9*a^3*c^5*d^2*e^2*m^2
 - 3*a^4*b^2*c^2*h^4*l^2 - 18*a^4*b^2*c^2*f^3*m^3 + 12*a^3*b^2*c^3*f^4*m^2
- 9*a^3*c^5*f^2*g^2*h^2 + 4*a^4*b^2*c^2*g^3*l^3 - 3*a^2*b^4*c^2*f^4*m^2 + 1
4*a^3*b^3*c^2*e^3*m^3 - 5*a^3*b^3*c^2*f^3*l^3 - 3*a^4*b^2*c^2*g^2*k^4 - 3*a
^3*b^2*c^3*g^4*k^2 + a^3*b^3*c^2*g^3*k^3 - 20*a^2*b^4*c^2*d^3*m^3 - 18*a^3*
b^2*c^3*e^3*l^3 + 16*a^3*b^2*c^3*d^3*m^3 + 12*a^4*b^2*c^2*e^2*l^4 + 12*a^2*
b^2*c^4*e^4*l^2 - 9*a^2*c^6*d^2*e^2*j^2 + 6*a^2*b^4*c^2*e^3*l^3 + 4*a^3*b^2
*c^3*f^3*k^3 + 14*a^2*b^3*c^3*d^3*l^3 - 9*a^2*c^6*e^2*f^2*g^2 - 9*a^2*c^6*d
^2*f^2*h^2 - 5*a^2*b^3*c^3*e^3*k^3 - 3*a^3*b^2*c^3*f^2*j^4 - 3*a^2*b^2*c^4*
f^4*j^2 + a^2*b^3*c^3*f^3*j^3 - 18*a^2*b^2*c^4*d^3*k^3 + 12*a^3*b^2*c^3*d^2
*k^4 + 4*a^2*b^2*c^4*e^3*j^3 - 3*a^2*b^4*c^2*d^2*k^4 - 3*a^2*b^2*c^4*e^2*h^
4 + 6*a^7*c*k*l^4*m - 3*a^7*b*k*l*m^4 - 6*a^7*c*h*k*m^4 - 6*a^7*c*g*l*m^4 +
 3*a^6*b*c*h*l^5 - 6*a*c^7*d^4*e*j - 6*a*c^7*d^4*f*h - 3*b*c^7*d^4*e*f + 6*
a*c^7*d*e^4*f + 3*a*b*c^6*e^5*h - a^5*b^2*c*j^3*l^3 - a^3*b^4*c*g^3*l^3 - a
*b^4*c^3*e^3*j^3 - a*b^2*c^5*e^3*g^3 + 3*a^7*b*j*m^5 + 6*a^7*c*f*m^5 + 6*a*
c^7*d^5*k + 3*b*c^7*d^5*g - 3*a^6*c^2*j^4*m^2 - 3*a^6*b^2*j^2*m^4 + 2*a^6*c
^2*j^3*l^3 + a^5*b^3*j^3*m^3 - 2*a^6*c^2*h^3*m^3 - 3*a^6*c^2*h^2*l^4 - 3*a^
5*c^3*h^4*l^2 - a*b^6*c*e^3*l^3 + 20*a^5*c^3*f^3*m^3 - 15*a^6*c^2*f^2*m^4 -
 15*a^4*c^4*f^4*m^2 + 2*a^5*c^3*h^3*k^3 - 2*a^5*c^3*g^3*l^3 + a^3*b^5*g^3*m
^3 - 3*a^5*c^3*g^2*k^4 - 3*a^4*c^4*g^4*k^2 - 3*a^4*b^4*f^2*m^4 + 20*a^4*c^4
*e^3*l^3 - 15*a^5*c^3*e^2*l^4 - 15*a^3*c^5*e^4*l^2 + 2*a^4*c^4*g^3*j^3 - 2*
a^4*c^4*f^3*k^3 - 2*a^4*c^4*d^3*m^3 - 3*b^4*c^4*d^4*k^2 - 3*a^4*c^4*f^2*j^4
 - 3*a^3*c^5*f^4*j^2 + 20*a^3*c^5*d^3*k^3 - 15*a^4*c^4*d^2*k^4 - 15*a^2*c^6
*d^4*k^2 - 2*a^3*c^5*e^3*j^3 + b^5*c^3*d^3*j^3 + 2*a^3*c^5*f^3*h^3 - 3*a^3*
c^5*e^2*h^4 - 3*a^2*c^6*e^4*h^2 - 3*b^2*c^6*d^4*g^2 + 2*a^2*c^6*e^3*g^3 - 2
*a^2*c^6*d^3*h^3 + b^3*c^5*d^3*g^3 - 3*a^2*c^6*d^2*g^4 - a^4*b^2*c^2*h^3*k^
3 - a^3*b^2*c^3*g^3*j^3 - a^2*b^4*c^2*f^3*k^3 - a^2*b^2*c^4*f^3*h^3 + 2*a^7
*c*k^3*m^3 + a^7*b*l^3*m^3 - 3*a^7*c*j^2*m^4 + 6*a^3*c^5*f^5*m - 3*a^6*b^2*
f*m^5 + 6*a^6*c^2*e*l^5 + 6*a^2*c^6*e^5*l + b^7*c*d^3*l^3 + a*b^7*e^3*m^3 -
 3*b^2*c^6*d^5*k + 6*a^5*c^3*d*k^5 - 3*a*c^7*d^4*g^2 + 2*a*c^7*d^3*f^3 + b*
c^7*d^3*e^3 - a^6*b^2*k^3*m^3 - a^4*b^4*h^3*m^3 - a^2*b^6*f^3*m^3 - b^6*c^2
*d^3*k^3 - b^4*c^4*d^3*h^3 - b^2*c^6*d^3*f^3 - b^8*d^3*m^3 - a^6*c^2*k^6 -
a^5*c^3*j^6 - a^4*c^4*h^6 - a^3*c^5*g^6 - a^2*c^6*f^6 - a^7*c*l^6 - a*c^7*e
^6 - a^8*m^6 - c^8*d^6, z, k1)*(243*a*b^5*c^6 + 3888*a^3*b*c^8 - 1944*a^2*b
^3*c^7))/c^3 + (x*(81*b^5*c^6*d - 1296*a^3*c^8*g + 648*a^2*b^2*c^7*g - 648*

a*b^3*c^7*d + 1296*a^2*b*c^8*d - 81*a*b^4*c^6*g))/c^3) + (216*a^2*b*c^7*f^2
 - 54*a*b^3*c^6*f^2 + 81*a*b^5*c^4*j^2 + 1512*a^3*b*c^6*j^2 + 81*a*b^7*c^2*
m^2 - 648*a^4*b*c^5*m^2 - 702*a^2*b^3*c^5*j^2 - 702*a^2*b^5*c^3*m^2 + 1674*
a^3*b^3*c^4*m^2 - 432*a^2*c^8*d*e + 27*b^4*c^6*d*e + 432*a^3*c^7*g*h + 432*
a^3*c^7*d*l + 432*a^3*c^7*e*k - 864*a^3*c^7*f*j + 864*a^4*c^6*j*m - 432*a^4
*c^6*k*l - 108*a*b^3*c^6*d*h - 108*a*b^3*c^6*e*g + 432*a^2*b*c^7*d*h + 432*
a^2*b*c^7*e*g + 81*a*b^4*c^5*g*h + 81*a*b^4*c^5*d*l + 81*a*b^4*c^5*e*k - 81
*a*b^5*c^4*g*l - 81*a*b^5*c^4*h*k + 432*a^3*b*c^6*f*m - 864*a^3*b*c^6*g*l -
 864*a^3*b*c^6*h*k - 162*a*b^6*c^3*j*m + 81*a*b^6*c^3*k*l - 432*a^2*b^2*c^6
*g*h - 432*a^2*b^2*c^6*d*l - 432*a^2*b^2*c^6*e*k + 216*a^2*b^2*c^6*f*j - 10
8*a^2*b^3*c^5*f*m + 540*a^2*b^3*c^5*g*l + 540*a^2*b^3*c^5*h*k + 1404*a^2*b^
4*c^4*j*m - 621*a^2*b^4*c^4*k*l - 3240*a^3*b^2*c^5*j*m + 1296*a^3*b^2*c^5*k
l)/c^3 + (x(216*a^2*c^8*e^2 + 27*b^4*c^6*e^2 - 216*a^3*c^7*h^2 + 216*a^4*
c^6*l^2 - 162*a*b^2*c^7*e^2 + 54*a^2*b^2*c^6*h^2 + 27*a^2*b^4*c^4*l^2 - 162
*a^3*b^2*c^5*l^2 + 432*a^2*c^8*d*f + 54*b^4*c^6*d*f - 81*b^5*c^5*d*j - 432*
a^3*c^7*d*m - 432*a^3*c^7*e*l - 432*a^3*c^7*f*k + 864*a^3*c^7*g*j + 81*b^6*
c^4*d*m + 432*a^4*c^6*k*m - 324*a*b^2*c^7*d*f - 54*a*b^3*c^6*e*h - 54*a*b^3
*c^6*f*g + 216*a^2*b*c^7*e*h + 216*a^2*b*c^7*f*g + 594*a*b^3*c^6*d*j - 1080
*a^2*b*c^7*d*j - 648*a*b^4*c^5*d*m + 81*a*b^4*c^5*g*j - 81*a*b^5*c^4*g*m -
1080*a^3*b*c^6*g*m + 216*a^3*b*c^6*h*l + 216*a^3*b*c^6*j*k + 1404*a^2*b^2*c
^6*d*m + 108*a^2*b^2*c^6*e*l + 108*a^2*b^2*c^6*f*k - 540*a^2*b^2*c^6*g*j +
594*a^2*b^3*c^5*g*m - 54*a^2*b^3*c^5*h*l - 54*a^2*b^3*c^5*j*k + 54*a^2*b^4*
c^4*k*m - 324*a^3*b^2*c^5*k*m))/c^3) + (36*a*c^8*d^3 + 9*a*b^8*m^3 - 9*b^2*
c^7*d^3 + 72*a^2*c^7*f^3 + 36*a^3*c^6*h^3 - 36*a^4*c^5*k^3 - 72*a^5*c^4*m^3
 - 18*a*b^2*c^6*f^3 - 9*a*b^3*c^5*g^3 + 36*a^2*b*c^6*g^3 + 9*a*b^4*c^4*h^3
- 108*a^2*c^7*d*g^2 - 9*a*b^5*c^3*j^3 - 288*a^3*b*c^5*j^3 + 9*a*b^6*c^2*k^3
 - 108*a^2*c^7*e^2*h + 108*a^4*b*c^4*l^3 - 81*a^2*b^6*c*m^3 - 108*a^2*c^7*d
^2*k + 108*a^3*c^6*d*k^2 + 216*a^3*c^6*f*j^2 + 108*a^3*c^6*g^2*k - 216*a^3*
c^6*f^2*m + 216*a^4*c^5*f*m^2 - 108*a^4*c^5*h*l^2 - 216*a^4*c^5*j^2*m - 45*
a^2*b^2*c^5*h^3 + 108*a^2*b^3*c^4*j^3 - 63*a^2*b^4*c^3*k^3 + 117*a^3*b^2*c^
4*k^3 + 72*a^2*b^5*c^2*l^3 - 171*a^3*b^3*c^3*l^3 + 180*a^3*b^4*c^2*m^3 + 18
*a^4*b^2*c^3*m^3 - 9*a*b^7*c*l^3 + 27*b^3*c^6*d*e*f + 216*a^2*c^7*d*e*j - 2
7*b^4*c^5*d*e*j + 27*b^5*c^4*d*e*m - 27*a*b^7*c*j*m^2 + 216*a^3*c^6*e*h*l -
 216*a^3*c^6*g*h*j - 216*a^3*c^6*d*j*l - 216*a^3*c^6*e*j*k + 216*a^4*c^5*j*
k*l + 27*a*b^2*c^6*d*g^2 + 27*a*b^2*c^6*e^2*h - 27*a*b^3*c^5*e*h^2 + 108*a^
2*b*c^6*e*h^2 + 27*a*b^2*c^6*d^2*k + 27*a*b^4*c^4*d*k^2 + 54*a*b^3*c^5*f^2*
j - 27*a*b^4*c^4*f*j^2 - 216*a^2*b*c^6*f^2*j - 27*a*b^3*c^5*e^2*l - 27*a*b^
5*c^3*e*l^2 + 108*a^2*b*c^6*e^2*l - 216*a^3*b*c^5*e*l^2 + 27*a*b^4*c^4*g^2*
k - 27*a*b^5*c^3*g*k^2 - 216*a^3*b*c^5*g*k^2 - 54*a*b^4*c^4*f^2*m - 27*a*b^
6*c^2*f*m^2 - 27*a*b^5*c^3*h^2*l + 27*a*b^6*c^2*h*l^2 - 216*a^3*b*c^5*h^2*l
 + 27*a*b^6*c^2*j^2*m + 216*a^4*b*c^4*j*m^2 - 135*a^2*b^2*c^5*d*k^2 + 54*a^
2*b^2*c^5*f*j^2 + 162*a^2*b^3*c^4*e*l^2 - 135*a^2*b^2*c^5*g^2*k + 162*a^2*b
^3*c^4*g*k^2 + 270*a^2*b^2*c^5*f^2*m + 162*a^2*b^4*c^3*f*m^2 - 270*a^3*b^2*
c^4*f*m^2 + 162*a^2*b^3*c^4*h^2*l - 189*a^2*b^4*c^3*h*l^2 + 351*a^3*b^2*c^4
*h*l^2 - 297*a^2*b^4*c^3*j^2*m + 270*a^2*b^5*c^2*j*m^2 + 810*a^3*b^2*c^4*j^

2*m - 702*a^3*b^3*c^3*j*m^2 - 108*a*b*c^7*d*e*f + 27*a*b^7*c*k*l*m + 54*a*b
^2*c^6*d*e*j - 27*a*b^3*c^5*f*g*h + 108*a^2*b*c^6*f*g*h - 81*a*b^3*c^5*d*e*
m - 27*a*b^3*c^5*d*f*l - 54*a*b^3*c^5*d*g*k + 54*a*b^3*c^5*d*h*j - 27*a*b^3
*c^5*e*f*k + 54*a*b^3*c^5*e*g*j - 108*a^2*b*c^6*d*e*m + 108*a^2*b*c^6*d*f*l
 + 216*a^2*b*c^6*d*g*k - 216*a^2*b*c^6*d*h*j + 108*a^2*b*c^6*e*f*k - 216*a^
2*b*c^6*e*g*j - 54*a*b^4*c^4*d*h*m - 54*a*b^4*c^4*e*g*m + 54*a*b^4*c^4*e*h*
l + 27*a*b^4*c^4*f*g*l + 27*a*b^4*c^4*f*h*k - 27*a*b^4*c^4*g*h*j - 27*a*b^4
*c^4*d*j*l - 27*a*b^4*c^4*e*j*k + 27*a*b^5*c^3*g*h*m + 108*a^3*b*c^5*g*h*m
+ 27*a*b^5*c^3*d*l*m + 27*a*b^5*c^3*e*k*m + 54*a*b^5*c^3*f*j*m - 27*a*b^5*c
^3*f*k*l + 27*a*b^5*c^3*g*j*l + 27*a*b^5*c^3*h*j*k + 108*a^3*b*c^5*d*l*m +
108*a^3*b*c^5*e*k*m - 108*a^3*b*c^5*f*k*l + 432*a^3*b*c^5*g*j*l + 432*a^3*b
*c^5*h*j*k - 27*a*b^6*c^2*g*l*m - 27*a*b^6*c^2*h*k*m - 27*a*b^6*c^2*j*k*l -
 108*a^4*b*c^4*k*l*m + 216*a^2*b^2*c^5*d*h*m + 216*a^2*b^2*c^5*e*g*m - 270*
a^2*b^2*c^5*e*h*l - 108*a^2*b^2*c^5*f*g*l - 108*a^2*b^2*c^5*f*h*k + 162*a^2
*b^2*c^5*g*h*j + 162*a^2*b^2*c^5*d*j*l + 162*a^2*b^2*c^5*e*j*k - 135*a^2*b^
3*c^4*g*h*m - 135*a^2*b^3*c^4*d*l*m - 135*a^2*b^3*c^4*e*k*m - 216*a^2*b^3*c
^4*f*j*m + 135*a^2*b^3*c^4*f*k*l - 216*a^2*b^3*c^4*g*j*l - 216*a^2*b^3*c^4*
h*j*k + 189*a^2*b^4*c^3*g*l*m + 189*a^2*b^4*c^3*h*k*m - 324*a^3*b^2*c^4*g*l
*m - 324*a^3*b^2*c^4*h*k*m + 243*a^2*b^4*c^3*j*k*l - 594*a^3*b^2*c^4*j*k*l
- 216*a^2*b^5*c^2*k*l*m + 459*a^3*b^3*c^3*k*l*m)/c^3 + (x*(27*b^2*c^7*d^2*e
 - 108*a^2*c^7*e*g^2 + 27*b^3*c^6*e^2*f - 27*b^3*c^6*d^2*h - 108*a^2*c^7*e^
2*j + 27*b^5*c^4*d*j^2 + 108*a^2*c^7*d^2*l - 108*a^3*c^6*e*k^2 - 27*b^4*c^5
*e^2*j - 216*a^3*c^6*g*j^2 + 27*b^4*c^5*d^2*l + 108*a^3*c^6*h^2*j + 27*b^7*
c^2*d*m^2 + 108*a^3*c^6*g^2*l + 27*b^5*c^4*e^2*m - 108*a^4*c^5*j*l^2 + 108*
a^4*c^5*k^2*l - 108*a*c^8*d^2*e - 108*a*b*c^7*e^2*f + 108*a*b*c^7*d^2*h - 2
7*b^3*c^6*d*e*g + 216*a^2*c^7*e*f*h + 216*a^2*c^7*d*e*k - 216*a^2*c^7*d*f*j
 + 27*b^4*c^5*d*g*h + 27*b^4*c^5*d*e*k - 27*b^4*c^5*d*f*j + 27*b^5*c^4*d*f*
m - 27*b^5*c^4*d*g*l - 27*b^5*c^4*d*h*k - 216*a^3*c^6*e*h*m - 216*a^3*c^6*f
*h*l + 216*a^3*c^6*d*j*m - 216*a^3*c^6*d*k*l + 216*a^3*c^6*e*j*l + 216*a^3*
c^6*f*j*k - 54*b^6*c^3*d*j*m + 27*b^6*c^3*d*k*l + 216*a^4*c^5*h*l*m - 216*a
^4*c^5*j*k*m + 27*a*b^2*c^6*e*g^2 - 189*a*b^3*c^5*d*j^2 + 324*a^2*b*c^6*d*j
^2 + 135*a*b^2*c^6*e^2*j - 27*a*b^3*c^5*g^2*h + 108*a^2*b*c^6*g^2*h - 135*a
*b^2*c^6*d^2*l - 27*a*b^4*c^4*g*j^2 - 216*a*b^5*c^3*d*m^2 - 216*a^3*b*c^5*d
*m^2 - 162*a*b^3*c^5*e^2*m + 216*a^2*b*c^6*e^2*m + 108*a^3*b*c^5*f*l^2 + 27
*a*b^4*c^4*g^2*l + 108*a^3*b*c^5*h*k^2 - 27*a*b^6*c^2*g*m^2 - 108*a^3*b*c^5
*h^2*m - 108*a^3*b*c^5*j^2*k + 216*a^4*b*c^4*k*m^2 + 27*a^2*b^2*c^5*e*k^2 +
 162*a^2*b^2*c^5*g*j^2 + 486*a^2*b^3*c^4*d*m^2 - 27*a^2*b^2*c^5*h^2*j - 27*
a^2*b^3*c^4*f*l^2 - 135*a^2*b^2*c^5*g^2*l - 27*a^2*b^3*c^4*h*k^2 + 189*a^2*
b^4*c^3*g*m^2 - 324*a^3*b^2*c^4*g*m^2 + 27*a^2*b^3*c^4*h^2*m + 27*a^2*b^3*c
^4*j^2*k + 27*a^3*b^2*c^4*j*l^2 + 27*a^2*b^4*c^3*k^2*l - 135*a^3*b^2*c^4*k^
2*l + 27*a^2*b^5*c^2*k*m^2 - 162*a^3*b^3*c^3*k*m^2 + 108*a*b*c^7*d*e*g - 10
8*a*b^2*c^6*d*g*h - 54*a*b^2*c^6*e*f*h - 162*a*b^2*c^6*d*e*k + 162*a*b^2*c^
6*d*f*j - 162*a*b^3*c^5*d*f*m + 135*a*b^3*c^5*d*g*l + 162*a*b^3*c^5*d*h*k -
 27*a*b^3*c^5*e*g*k + 54*a*b^3*c^5*e*h*j + 27*a*b^3*c^5*f*g*j + 216*a^2*b*c
^6*d*f*m - 108*a^2*b*c^6*d*g*l - 216*a^2*b*c^6*d*h*k + 108*a^2*b*c^6*e*g*k

- 216*a^2*b*c^6*e*h*j - 108*a^2*b*c^6*f*g*j - 54*a*b^4*c^4*e*h*m - 27*a*b^4
*c^4*f*g*m + 27*a*b^4*c^4*g*h*k + 405*a*b^4*c^4*d*j*m - 189*a*b^4*c^4*d*k*l
+ 54*a*b^5*c^3*g*j*m - 27*a*b^5*c^3*g*k*l - 216*a^3*b*c^5*e*l*m - 216*a^3*
b*c^5*f*k*m + 540*a^3*b*c^5*g*j*m - 108*a^3*b*c^5*g*k*l + 270*a^2*b^2*c^5*e
*h*m + 108*a^2*b^2*c^5*f*g*m + 54*a^2*b^2*c^5*f*h*l - 108*a^2*b^2*c^5*g*h*k
- 810*a^2*b^2*c^5*d*j*m + 378*a^2*b^2*c^5*d*k*l - 54*a^2*b^2*c^5*e*j*l - 5
4*a^2*b^2*c^5*f*j*k + 54*a^2*b^3*c^4*e*l*m + 54*a^2*b^3*c^4*f*k*m - 351*a^2
*b^3*c^4*g*j*m + 135*a^2*b^3*c^4*g*k*l - 54*a^3*b^2*c^4*h*l*m - 54*a^2*b^4*
c^3*j*k*m + 270*a^3*b^2*c^4*j*k*m))/c^3) - (6*a^3*b^5*m^4 - 9*b*c^7*d^2*e^2
- 27*a^3*b*c^4*j^4 + 12*a^2*c^6*f*g^3 - 30*a^4*b^3*c*m^4 + 21*a^5*b*c^2*m^
4 - 6*b^2*c^6*d^3*j + 24*a^2*c^6*f^3*j + 24*a^3*c^5*f*j^3 + 12*a^2*c^6*e^3*
m + 12*a^3*c^5*h^3*j + 12*a^4*c^4*f*l^3 + 6*b^3*c^5*d^3*m - 12*a^3*c^5*g^3*
m - 12*a^4*c^4*j*k^3 - 6*a^2*b^6*j*m^3 - 24*a^4*c^4*j^3*m - 24*a^5*c^3*j*m^
3 - 12*a^5*c^3*l^3*m + 6*a^2*b^3*c^3*j^4 - 3*a*b*c^6*f^4 - 12*a*c^7*e^3*f +
6*b*c^7*d^3*f + 12*a*c^7*d^3*j + 6*a*b^7*f*m^3 + 36*a*c^7*d*e*f^2 + 6*a*b*
c^6*e^3*j - 36*a*c^7*d^2*f*g - 18*a*b*c^6*d^3*m - 6*a*b^6*c*f*l^3 - 54*a^2*
b^3*c^3*f^2*m^2 - 81*a^3*b^3*c^2*j^2*m^2 - 9*a*b*c^6*d^2*h^2 - 9*a*b*c^6*e^
2*g^2 - 6*a*b^2*c^5*f*g^3 + 6*a*b^3*c^4*f*h^3 - 18*a^2*b*c^5*f*h^3 - 9*b^2*
c^6*d*e*f^2 + 9*b^2*c^6*d*e^2*g - 6*a*b^4*c^3*f*j^3 + 9*b^2*c^6*d^2*e*h + 6
*a*b^5*c^2*f*k^3 + 6*a^2*b*c^5*g^3*j + 36*a^2*c^6*d*e*j^2 + 36*a^2*c^6*e*f*
h^2 + 30*a^3*b*c^4*f*k^3 - 6*a*b^2*c^5*e^3*m - 9*b^4*c^4*d*e*j^2 - 12*a^2*b
*c^5*f^3*m - 42*a^2*b^5*c*f*m^3 - 36*a^2*c^6*d*g^2*j - 36*a^2*c^6*f^2*g*h -
60*a^4*b*c^3*f*m^3 - 9*b^3*c^5*d*e^2*k - 36*a^2*c^6*d*f^2*l - 36*a^2*c^6*e
*f^2*k + 36*a^3*c^5*d*e*m^2 - 9*b^3*c^5*d^2*e*l + 36*a^2*c^6*e^2*f*l - 36*a
^2*c^6*e^2*h*j + 6*a^3*b*c^4*h^3*m - 36*a^3*c^5*e*f*l^2 - 9*b^6*c^2*d*e*m^2
+ 6*a^2*b^5*c*j*l^3 + 36*a^2*c^6*d^2*g*m - 36*a^3*c^5*f*g*k^2 + 30*a^4*b*c
^3*j*l^3 - 36*a^2*c^6*d^2*j*k + 18*a^3*b^4*c*j*m^3 + 36*a^3*c^5*d*j*k^2 - 3
6*a^3*c^5*g*h*j^2 - 36*a^3*c^5*d*j^2*l - 36*a^3*c^5*e*h^2*m - 36*a^3*c^5*e*
j^2*k - 36*a^3*c^5*f*h^2*l - 18*a^4*b*c^3*k^3*m - 6*a^3*b^4*c*l^3*m + 36*a^
3*c^5*g^2*j*k - 36*a^4*c^4*g*h*m^2 - 72*a^3*c^5*f^2*j*m + 36*a^3*c^5*f^2*k*
l - 36*a^4*c^4*d*l*m^2 - 36*a^4*c^4*e*k*m^2 + 72*a^4*c^4*f*j*m^2 - 36*a^3*c
^5*e^2*l*m + 36*a^4*c^4*e*l^2*m - 36*a^4*c^4*h*j*l^2 + 36*a^4*c^4*g*k^2*m +
36*a^4*c^4*h^2*l*m + 36*a^4*c^4*j^2*k*l + 36*a^5*c^3*k*l*m^2 - 9*a^2*b*c^5
*g^2*h^2 + 9*a*b^3*c^4*f^2*j^2 - 9*a^2*b*c^5*d^2*l^2 - 9*a^2*b*c^5*e^2*k^2
- 54*a^2*b*c^5*f^2*j^2 + 24*a^2*b^2*c^4*f*j^3 - 30*a^2*b^3*c^3*f*k^3 - 6*a^
2*b^2*c^4*h^3*j + 36*a^2*b^4*c^2*f*l^3 - 54*a^3*b^2*c^3*f*l^3 + 9*a*b^5*c^2
*f^2*m^2 + 54*a^3*b*c^4*f^2*m^2 - 9*a^3*b*c^4*g^2*l^2 - 9*a^3*b*c^4*h^2*k^2
+ 84*a^3*b^3*c^2*f*m^3 - 6*a^2*b^4*c^2*j*k^3 + 24*a^3*b^2*c^3*j*k^3 - 30*a
^3*b^3*c^2*j*l^3 - 18*a^2*b^4*c^2*j^3*m + 18*a^2*b^5*c*j^2*m^2 + 84*a^3*b^2
*c^3*j^3*m + 18*a^4*b*c^3*j^2*m^2 - 9*a^4*b*c^3*k^2*l^2 + 36*a^4*b^2*c^2*j*
m^3 + 6*a^3*b^3*c^2*k^3*m + 24*a^4*b^2*c^2*l^3*m - 45*a^2*b^2*c^4*d*e*m^2 +
9*a^2*b^2*c^4*d*g*l^2 + 72*a^2*b^2*c^4*e*f*l^2 + 9*a^2*b^2*c^4*e*h*k^2 + 7
2*a^2*b^2*c^4*f*g*k^2 - 18*a^2*b^2*c^4*d*j*k^2 + 9*a^2*b^2*c^4*g*h*j^2 - 36
*a^2*b^3*c^3*d*h*m^2 - 36*a^2*b^3*c^3*e*g*m^2 + 9*a^2*b^2*c^4*d*j^2*l + 9*a
^2*b^2*c^4*e*j^2*k + 72*a^2*b^2*c^4*f*h^2*l + 9*a^2*b^2*c^4*g*h^2*k - 90*a^

2*b^3*c^3*f*h*l^2 + 9*a^2*b^2*c^4*g^2*h*l - 9*a^2*b^3*c^3*d*k*l^2 + 18*a^2*
b^3*c^3*e*j*l^2 - 18*a^2*b^2*c^4*g^2*j*k - 9*a^2*b^3*c^3*e*k^2*l + 18*a^2*b
^3*c^3*g*j*k^2 - 9*a^2*b^4*c^2*g*h*m^2 + 45*a^3*b^2*c^3*g*h*m^2 + 108*a^2*b
^2*c^4*f^2*j*m - 45*a^2*b^2*c^4*f^2*k*l - 90*a^2*b^3*c^3*f*j^2*m - 18*a^2*b
^3*c^3*g*j^2*l - 18*a^2*b^3*c^3*h*j^2*k - 9*a^2*b^4*c^2*d*l*m^2 - 9*a^2*b^4
*c^2*e*k*m^2 + 108*a^2*b^4*c^2*f*j*m^2 + 45*a^3*b^2*c^3*d*l*m^2 + 45*a^3*b^
2*c^3*e*k*m^2 - 144*a^3*b^2*c^3*f*j*m^2 + 18*a^2*b^3*c^3*h^2*j*l - 18*a^2*b
^4*c^2*h*j*l^2 - 18*a^3*b^2*c^3*e*l^2*m + 9*a^3*b^2*c^3*g*k*l^2 + 72*a^3*b^
2*c^3*h*j*l^2 - 18*a^3*b^2*c^3*g*k^2*m + 9*a^3*b^2*c^3*h*k^2*l - 9*a^3*b^3*
c^2*g*l*m^2 - 9*a^3*b^3*c^2*h*k*m^2 + 18*a^2*b^4*c^2*j^2*k*l - 18*a^3*b^2*c
^3*h^2*l*m - 81*a^3*b^2*c^3*j^2*k*l + 18*a^3*b^3*c^2*h*l^2*m - 81*a^4*b^2*c
^2*k*l*m^2 + 18*a*b*c^6*d*f*g^2 + 18*a*b*c^6*e^2*f*h + 18*a*b*c^6*d*e^2*k +
 18*a*b*c^6*d^2*e*l + 18*a*b*c^6*d^2*f*k + 18*a*b*c^6*d^2*g*j - 9*b^3*c^5*d
*e*g*h + 18*b^3*c^5*d*e*f*j - 72*a^2*c^6*d*e*f*m + 72*a^2*c^6*d*f*g*k - 18*
b^4*c^4*d*e*f*m + 9*b^4*c^4*d*e*g*l + 9*b^4*c^4*d*e*h*k - 18*a*b^6*c*f*j*m^
2 + 18*b^5*c^3*d*e*j*m - 9*b^5*c^3*d*e*k*l + 72*a^3*c^5*f*g*h*m + 72*a^3*c^
5*d*f*l*m - 72*a^3*c^5*d*g*k*m + 72*a^3*c^5*e*f*k*m + 72*a^3*c^5*e*h*j*l -
72*a^4*c^4*f*k*l*m + 27*a*b^2*c^5*d*e*j^2 + 9*a*b^2*c^5*d*g*h^2 - 18*a*b^2*
c^5*e*f*h^2 + 9*a*b^2*c^5*e*g^2*h + 9*a*b^2*c^5*f^2*g*h + 18*a*b^3*c^4*d*f*
k^2 - 54*a^2*b*c^5*d*f*k^2 + 9*a*b^2*c^5*d*f^2*l + 9*a*b^2*c^5*e*f^2*k + 9*
a*b^3*c^4*d*h*j^2 + 9*a*b^3*c^4*e*g*j^2 + 45*a*b^4*c^3*d*e*m^2 - 36*a^2*b*c
^5*d*h*j^2 - 36*a^2*b*c^5*e*g*j^2 - 18*a*b^2*c^5*e^2*f*l + 9*a*b^2*c^5*e^2*
g*k - 9*a*b^3*c^4*d*h^2*k - 18*a*b^4*c^3*e*f*l^2 + 18*a^2*b*c^5*d*h^2*k + 1
8*a^2*b*c^5*e*h^2*j - 18*a*b^2*c^5*d^2*g*m + 9*a*b^2*c^5*d^2*h*l - 9*a*b^3*
c^4*e*g^2*l + 18*a*b^3*c^4*f*g^2*k - 18*a*b^4*c^3*f*g*k^2 + 18*a^2*b*c^5*d*
g^2*m + 18*a^2*b*c^5*e*g^2*l - 54*a^2*b*c^5*f*g^2*k - 9*a*b^3*c^4*f^2*g*l -
 9*a*b^3*c^4*f^2*h*k + 9*a*b^5*c^2*d*h*m^2 + 9*a*b^5*c^2*e*g*m^2 + 36*a^2*b
*c^5*f^2*g*l + 36*a^2*b*c^5*f^2*h*k - 18*a*b^4*c^3*f*h^2*l + 18*a*b^5*c^2*f
*h*l^2 + 18*a^2*b*c^5*e^2*h*m + 90*a^3*b*c^4*f*h*l^2 + 18*a^2*b*c^5*e^2*j*l
 + 18*a^3*b*c^4*d*k*l^2 - 54*a^3*b*c^4*e*j*l^2 + 18*a^2*b*c^5*d^2*k*m + 18*
a^3*b*c^4*d*k^2*m + 18*a^3*b*c^4*e*k^2*l - 54*a^3*b*c^4*g*j*k^2 - 18*a*b^4*
c^3*f^2*j*m + 9*a*b^4*c^3*f^2*k*l + 18*a*b^5*c^2*f*j^2*m + 36*a^3*b*c^4*f*j
^2*m + 72*a^3*b*c^4*g*j^2*l + 72*a^3*b*c^4*h*j^2*k - 54*a^3*b*c^4*h^2*j*l +
 18*a^3*b*c^4*g^2*k*m + 36*a^4*b*c^3*g*l*m^2 + 36*a^4*b*c^3*h*k*m^2 - 54*a^
4*b*c^3*h*l^2*m + 18*a^3*b^4*c*k*l*m^2 - 90*a^2*b^2*c^4*f*g*h*m - 90*a^2*b^
2*c^4*d*f*l*m + 72*a^2*b^2*c^4*d*h*j*m - 18*a^2*b^2*c^4*d*h*k*l - 90*a^2*b^
2*c^4*e*f*k*m + 72*a^2*b^2*c^4*e*g*j*m - 18*a^2*b^2*c^4*e*g*k*l - 36*a^2*b^
2*c^4*e*h*j*l - 72*a^2*b^2*c^4*f*g*j*l - 72*a^2*b^2*c^4*f*h*j*k + 90*a^2*b^
3*c^3*f*g*l*m + 90*a^2*b^3*c^3*f*h*k*m - 9*a^2*b^3*c^3*g*h*k*l + 90*a^2*b^3
*c^3*f*j*k*l - 108*a^2*b^4*c^2*f*k*l*m + 18*a^2*b^4*c^2*g*j*l*m + 18*a^2*b^
4*c^2*h*j*k*m + 162*a^3*b^2*c^3*f*k*l*m - 72*a^3*b^2*c^3*g*j*l*m - 72*a^3*b
^2*c^3*h*j*k*m + 72*a^3*b^3*c^2*j*k*l*m - 72*a*b*c^6*d*e*f*j + 18*a*b^6*c*f
*k*l*m + 90*a*b^2*c^5*d*e*f*m - 18*a*b^2*c^5*d*e*g*l - 18*a*b^2*c^5*d*e*h*k
 - 36*a*b^2*c^5*d*f*g*k - 9*a*b^3*c^4*d*g*h*l + 36*a*b^3*c^4*e*f*h*l - 9*a*
b^3*c^4*e*g*h*k - 18*a*b^3*c^4*f*g*h*j - 108*a^2*b*c^5*e*f*h*l + 72*a^2*b*c

^5*f*g*h*j - 72*a*b^3*c^4*d*e*j*m + 36*a*b^3*c^4*d*e*k*l - 18*a*b^3*c^4*d*f
*j*l - 18*a*b^3*c^4*e*f*j*k - 36*a^2*b*c^5*d*e*k*l + 72*a^2*b*c^5*d*f*j*l +
36*a^2*b*c^5*d*g*j*k + 72*a^2*b*c^5*e*f*j*k + 18*a*b^4*c^3*f*g*h*m + 18*a*
b^4*c^3*d*f*l*m - 18*a*b^4*c^3*d*h*j*m + 9*a*b^4*c^3*d*h*k*l + 18*a*b^4*c^3
*e*f*k*m - 18*a*b^4*c^3*e*g*j*m + 9*a*b^4*c^3*e*g*k*l + 18*a*b^4*c^3*f*g*j*
l + 18*a*b^4*c^3*f*h*j*k - 18*a*b^5*c^2*f*g*l*m - 18*a*b^5*c^2*f*h*k*m + 36
*a^3*b*c^4*e*h*l*m - 72*a^3*b*c^4*f*g*l*m - 72*a^3*b*c^4*f*h*k*m - 18*a*b^5
*c^2*f*j*k*l - 72*a^3*b*c^4*f*j*k*l - 18*a^2*b^5*c*j*k*l*m)/c^3 + (x*(6*c^8
*d^4 + 3*b^8*d*m^3 - 6*a^2*c^6*g^4 + 6*a^4*c^4*k^4 + 3*a*b^2*c^5*g^4 - 18*a
*c^7*e^2*f^2 - 6*b^2*c^6*d*f^3 - 12*a^2*c^6*d*h^3 - 3*b^3*c^5*d*g^3 - 9*b^2
*c^6*e^3*g + 3*b^4*c^4*d*h^3 - 24*a^3*c^5*d*k^3 - 3*b^5*c^3*d*j^3 + 12*b^2*
c^6*d^3*k + 3*b^6*c^2*d*k^3 - 12*a^2*c^6*f^3*k + 24*a^3*c^5*g*j^3 - 12*a^4*
c^4*d*m^3 + 9*b^3*c^5*e^3*k + 12*a^3*c^5*h^3*k - 24*a^4*c^4*g*l^3 + 3*a^2*b
^6*k*m^3 + 12*a^5*c^3*k*m^3 + 9*b^2*c^6*d^2*g^2 + 9*b^2*c^6*e^2*f^2 + 3*a^2
*b^4*c^2*k^4 - 12*a^3*b^2*c^3*k^4 + 18*a^2*c^6*f^2*h^2 + 36*a^2*c^6*d^2*k^2
+ 18*a^2*c^6*e^2*j^2 + 9*b^4*c^4*d^2*k^2 + 9*b^4*c^4*e^2*j^2 - 18*a^3*c^5*
e^2*m^2 - 18*a^3*c^5*f^2*l^2 - 18*a^3*c^5*h^2*j^2 + 9*b^6*c^2*e^2*m^2 + 18*
a^4*c^4*h^2*m^2 + 18*a^4*c^4*j^2*l^2 - 18*a^5*c^3*l^2*m^2 + 12*a*c^7*d*f^3
+ 6*b*c^7*d*e^3 + 24*a*c^7*e^3*g - 12*b*c^7*d^3*g - 24*a*c^7*d^3*k - 3*b^7*
c*d*l^3 - 3*a*b^7*g*m^3 + 6*a*b*c^6*f^3*g - 36*a*c^7*d*e^2*h - 30*a*b*c^6*e
^3*k - 24*a*b^6*c*d*m^3 + 36*a*c^7*d^2*e*j + 3*a*b^6*c*g*l^3 - 9*b^7*c*d*j*
m^2 + 81*a^2*b^2*c^4*e^2*m^2 + 9*a^2*b^2*c^4*f^2*l^2 - 27*a^2*b^2*c^4*g^2*k
^2 + 9*a^2*b^2*c^4*h^2*j^2 + 9*a^2*b^4*c^2*h^2*m^2 - 36*a^3*b^2*c^3*h^2*m^2
+ 9*a^4*b^2*c^2*l^2*m^2 - 12*a*b^2*c^5*d*h^3 + 24*a*b^3*c^4*d*j^3 - 42*a^2
*b*c^5*d*j^3 - 3*a*b^3*c^4*g*h^3 - 18*a*b^4*c^3*d*k^3 + 18*a^2*b*c^5*g*h^3
+ 21*a*b^5*c^2*d*l^3 + 30*a^3*b*c^4*d*l^3 - 9*b^3*c^5*d*e*h^2 + 3*a*b^4*c^3
*g*j^3 - 9*a*b^3*c^4*g^3*k - 3*a*b^5*c^2*g*k^3 + 24*a^2*b*c^5*g^3*k + 36*a^
2*c^6*d*f*j^2 + 12*a^3*b*c^4*g*k^3 - 9*b^2*c^6*d^2*e*j + 9*b^3*c^5*d*f^2*j
+ 9*b^3*c^5*e^2*g*h + 21*a^2*b^5*c*g*m^3 + 36*a^2*c^6*e*g^2*j - 6*a^4*b*c^3
*g*m^3 - 18*b^3*c^5*e^2*f*j - 9*b^5*c^3*d*e*l^2 - 36*a^2*c^6*d*f^2*m + 36*a
^2*c^6*e*f^2*l + 18*a^3*b*c^4*j^3*k + 36*a^3*c^5*d*f*m^2 + 9*b^3*c^5*d^2*e*
m - 18*b^3*c^5*d^2*g*k + 9*b^3*c^5*d^2*h*j + 9*b^4*c^4*d*g^2*k - 9*b^5*c^3*
d*g*k^2 + 36*a^2*c^6*e^2*f*m - 72*a^2*c^6*e^2*g*l + 36*a^2*c^6*e^2*h*k - 36
*a^3*c^5*d*h*l^2 + 72*a^3*c^5*e*g*l^2 - 9*b^4*c^4*d*f^2*m - 3*a^2*b^5*c*k*l
^3 - 6*a^4*b*c^3*k*l^3 + 18*b^4*c^4*e^2*f*m - 9*b^4*c^4*e^2*g*l - 9*b^4*c^4
*e^2*h*k - 9*b^5*c^3*d*h^2*l + 9*b^6*c^2*d*h*l^2 - 36*a^2*c^6*d^2*j*l - 18*
a^3*b^4*c*k*m^3 + 36*a^3*c^5*e*j*k^2 - 9*b^4*c^4*d^2*h*m - 36*a^3*c^5*d*j^2
*m - 36*a^3*c^5*e*j^2*l - 36*a^3*c^5*f*h^2*m - 36*a^3*c^5*f*j^2*k - 9*b^4*c
^4*d^2*j*l + 9*b^6*c^2*d*j^2*m - 36*a^3*c^5*g^2*j*l - 18*b^5*c^3*e^2*j*m +
9*b^5*c^3*e^2*k*l + 36*a^3*c^5*f^2*k*m + 36*a^4*c^4*e*l*m^2 - 36*a^4*c^4*f*
k*m^2 + 9*b^5*c^3*d^2*l*m + 36*a^4*c^4*f*l^2*m + 36*a^4*c^4*h*k*l^2 - 36*a^
4*c^4*j*k^2*l + 36*a^4*c^4*j^2*k*m - 36*a*b^2*c^5*d^2*k^2 - 36*a*b^2*c^5*e^
2*j^2 + 36*a^2*b^2*c^4*d*k^3 - 42*a^2*b^3*c^3*d*l^3 - 21*a^2*b^2*c^4*g*j^3
+ 51*a^2*b^4*c^2*d*m^3 - 12*a^3*b^2*c^3*d*m^3 - 54*a*b^4*c^3*e^2*m^2 + 9*a*
b^4*c^3*g^2*k^2 + 6*a^2*b^3*c^3*g*k^3 - 6*a^2*b^2*c^4*h^3*k - 18*a^2*b^4*c^

2*g*l^3 + 27*a^3*b^2*c^3*g*l^3 - 33*a^3*b^3*c^2*g*m^3 - 3*a^2*b^3*c^3*j^3*k
 + 15*a^3*b^3*c^2*k*l^3 + 18*a^4*b^2*c^2*k*m^3 + 9*b^7*c*d*k*l*m - 18*a^2*b
^2*c^4*d*f*m^2 + 72*a^2*b^2*c^4*d*h*l^2 - 63*a^2*b^2*c^4*e*g*l^2 - 9*a^2*b^
2*c^4*e*j*k^2 + 90*a^2*b^3*c^3*e*h*m^2 + 144*a^2*b^2*c^4*d*j^2*m + 18*a^2*b
^2*c^4*e*j^2*l + 18*a^2*b^2*c^4*f*h^2*m - 45*a^2*b^2*c^4*g*h^2*l - 153*a^2*
b^3*c^3*d*j*m^2 + 45*a^2*b^3*c^3*g*h*l^2 + 36*a^2*b^2*c^4*g^2*h*m + 9*a^2*b
^3*c^3*e*k*l^2 + 45*a^2*b^2*c^4*g^2*j*l + 9*a^2*b^3*c^3*e*k^2*m + 9*a^2*b^3
*c^3*h*j*k^2 - 18*a^2*b^2*c^4*f^2*k*m + 63*a^2*b^3*c^3*g*j^2*m + 18*a^2*b^4
*c^2*e*l*m^2 - 63*a^2*b^4*c^2*g*j*m^2 - 72*a^3*b^2*c^3*e*l*m^2 + 99*a^3*b^2
*c^3*g*j*m^2 - 18*a^2*b^3*c^3*h^2*j*m + 9*a^2*b^4*c^2*h*k*l^2 - 54*a^3*b^2*
c^3*h*k*l^2 - 45*a^2*b^3*c^3*g^2*l*m - 9*a^2*b^4*c^2*h*k^2*m + 36*a^3*b^2*c
^3*h*k^2*m - 9*a^2*b^4*c^2*j*k^2*l + 45*a^3*b^2*c^3*j*k^2*l - 18*a^3*b^3*c^
2*h*l*m^2 + 9*a^2*b^4*c^2*j^2*k*m - 54*a^3*b^2*c^3*j^2*k*m + 54*a^3*b^3*c^2
*j*k*m^2 - 45*a^3*b^3*c^2*k^2*l*m + 54*a*b*c^6*d*e*h^2 - 18*a*b*c^6*e*f^2*h
 - 18*a*b*c^6*d*f^2*j - 18*a*b*c^6*e^2*g*h + 18*a*b*c^6*d*e^2*l + 54*a*b*c^
6*e^2*f*j - 36*a*b*c^6*d^2*e*m + 36*a*b*c^6*d^2*g*k - 36*a*b*c^6*d^2*h*j +
9*b^3*c^5*d*e*g*j + 72*a^2*c^6*d*e*h*l - 72*a^2*c^6*e*f*h*j - 72*a^2*c^6*d*
e*j*k - 9*b^4*c^4*d*e*g*m + 18*b^4*c^4*d*e*h*l - 9*b^4*c^4*d*g*h*j - 9*b^4*
c^4*d*e*j*k + 9*a*b^6*c*g*j*m^2 + 9*b^5*c^3*d*g*h*m + 9*b^5*c^3*d*e*k*m + 9
*b^5*c^3*d*g*j*l + 9*b^5*c^3*d*h*j*k - 72*a^3*c^5*e*f*l*m + 72*a^3*c^5*e*h*
j*m - 72*a^3*c^5*e*h*k*l + 72*a^3*c^5*f*h*j*l + 72*a^3*c^5*d*j*k*l - 9*b^6*
c^2*d*g*l*m - 9*b^6*c^2*d*h*k*m - 9*b^6*c^2*d*j*k*l - 72*a^4*c^4*h*j*l*m -
18*a*b^2*c^5*d*f*j^2 - 9*a*b^2*c^5*e*g*h^2 + 54*a*b^3*c^4*d*e*l^2 - 54*a^2*
b*c^5*d*e*l^2 - 18*a*b^2*c^5*d*g^2*k - 9*a*b^2*c^5*e*g^2*j + 36*a*b^3*c^4*d
*g*k^2 - 36*a^2*b*c^5*d*g*k^2 + 36*a*b^2*c^5*d*f^2*m - 9*a*b^2*c^5*f^2*g*j
- 18*a*b^3*c^4*e*h*j^2 + 54*a^2*b*c^5*e*h*j^2 + 18*a^2*b*c^5*f*g*j^2 - 72*a
*b^2*c^5*e^2*f*m + 45*a*b^2*c^5*e^2*g*l + 18*a*b^2*c^5*e^2*h*k + 45*a*b^3*c
^4*d*h^2*l + 18*a*b^3*c^4*e*h^2*k - 54*a*b^4*c^3*d*h*l^2 + 9*a*b^4*c^3*e*g*
l^2 - 18*a^2*b*c^5*d*h^2*l - 54*a^2*b*c^5*e*h^2*k - 18*a^2*b*c^5*f*h^2*j +
36*a*b^2*c^5*d^2*h*m + 9*a*b^3*c^4*e*g^2*m + 9*a*b^3*c^4*g^2*h*j - 36*a^2*b
*c^5*e*g^2*m - 36*a^2*b*c^5*g^2*h*j + 45*a*b^2*c^5*d^2*j*l + 9*a*b^3*c^4*f^
2*g*m - 18*a*b^5*c^2*e*h*m^2 - 18*a^2*b*c^5*f^2*g*m - 18*a^2*b*c^5*f^2*h*l
- 90*a^3*b*c^4*e*h*m^2 + 18*a^3*b*c^4*f*g*m^2 - 72*a*b^4*c^3*d*j^2*m + 9*a*
b^4*c^3*g*h^2*l + 72*a*b^5*c^2*d*j*m^2 - 9*a*b^5*c^2*g*h*l^2 + 18*a^2*b*c^5
*f^2*j*k + 54*a^3*b*c^4*d*j*m^2 - 18*a^3*b*c^4*g*h*l^2 + 90*a*b^3*c^4*e^2*j
*m - 45*a*b^3*c^4*e^2*k*l - 9*a*b^4*c^3*g^2*h*m - 90*a^2*b*c^5*e^2*j*m + 54
*a^2*b*c^5*e^2*k*l - 18*a^3*b*c^4*e*k*l^2 - 18*a^3*b*c^4*f*j*l^2 - 45*a*b^3
*c^4*d^2*l*m - 9*a*b^4*c^3*g^2*j*l + 36*a^2*b*c^5*d^2*l*m - 36*a^3*b*c^4*e*
k^2*m - 36*a^3*b*c^4*h*j*k^2 - 9*a*b^5*c^2*g*j^2*m - 90*a^3*b*c^4*g*j^2*m -
 18*a^3*b*c^4*h*j^2*l + 54*a^3*b*c^4*h^2*j*m + 18*a^3*b*c^4*h^2*k*l + 9*a*b
^5*c^2*g^2*l*m + 36*a^3*b*c^4*g^2*l*m + 54*a^4*b*c^3*h*l*m^2 - 9*a^2*b^5*c*
j*k*m^2 - 54*a^4*b*c^3*j*k*m^2 - 18*a^4*b*c^3*j*l^2*m + 9*a^2*b^5*c*k^2*l*m
 + 36*a^4*b*c^3*k^2*l*m - 36*a^2*b^2*c^4*d*g*l*m - 72*a^2*b^2*c^4*d*h*k*m +
 36*a^2*b^2*c^4*e*f*l*m + 36*a^2*b^2*c^4*e*g*k*m - 144*a^2*b^2*c^4*e*h*j*m
+ 72*a^2*b^2*c^4*e*h*k*l - 18*a^2*b^2*c^4*f*g*j*m + 36*a^2*b^2*c^4*g*h*j*k

- 126*a^2*b^2*c^4*d*j*k*l - 36*a^2*b^3*c^3*g*h*k*m + 126*a^2*b^3*c^3*d*k*l*
m - 36*a^2*b^3*c^3*e*j*l*m - 45*a^2*b^3*c^3*g*j*k*l + 45*a^2*b^4*c^2*g*k*l*
m - 36*a^3*b^2*c^3*g*k*l*m + 36*a^3*b^2*c^3*h*j*l*m - 36*a*b*c^6*d*e*g*j -
9*a*b^6*c*g*k*l*m + 36*a*b^2*c^5*d*e*g*m - 108*a*b^2*c^5*d*e*h*l + 36*a*b^2
*c^5*d*g*h*j + 36*a*b^2*c^5*e*f*h*j + 54*a*b^2*c^5*d*e*j*k - 36*a*b^3*c^4*d
*g*h*m - 36*a*b^3*c^4*e*f*h*m + 108*a^2*b*c^5*e*f*h*m + 36*a^2*b*c^5*e*g*h*
l - 54*a*b^3*c^4*d*e*k*m + 18*a*b^3*c^4*d*f*j*m - 45*a*b^3*c^4*d*g*j*l - 54
*a*b^3*c^4*d*h*j*k + 9*a*b^3*c^4*e*g*j*k + 72*a^2*b*c^5*d*e*k*m - 36*a^2*b*
c^5*d*f*j*m + 36*a^2*b*c^5*d*g*j*l + 72*a^2*b*c^5*d*h*j*k - 36*a^2*b*c^5*e*
f*j*l - 36*a^2*b*c^5*e*g*j*k + 45*a*b^4*c^3*d*g*l*m + 54*a*b^4*c^3*d*h*k*m
- 9*a*b^4*c^3*e*g*k*m + 36*a*b^4*c^3*e*h*j*m - 18*a*b^4*c^3*e*h*k*l - 9*a*b
^4*c^3*g*h*j*k + 63*a*b^4*c^3*d*j*k*l + 9*a*b^5*c^2*g*h*k*m - 36*a^3*b*c^4*
f*h*l*m - 63*a*b^5*c^2*d*k*l*m + 9*a*b^5*c^2*g*j*k*l - 72*a^3*b*c^4*d*k*l*m
 + 108*a^3*b*c^4*e*j*l*m + 36*a^3*b*c^4*f*j*k*m + 36*a^3*b*c^4*g*j*k*l))/c^
3))*root(34992*a^4*b^2*c^8*z^6 - 8748*a^3*b^4*c^7*z^6 + 729*a^2*b^6*c^6*z^6
 - 46656*a^5*c^9*z^6 + 34992*a^4*b^3*c^6*m*z^5 - 8748*a^3*b^5*c^5*m*z^5 + 7
29*a^2*b^7*c^4*m*z^5 - 34992*a^4*b^2*c^7*j*z^5 + 8748*a^3*b^4*c^6*j*z^5 - 7
29*a^2*b^6*c^5*j*z^5 - 46656*a^5*b*c^7*m*z^5 + 46656*a^5*c^8*j*z^5 + 34992*
a^5*b*c^6*j*m*z^4 - 11664*a^5*b*c^6*k*l*z^4 + 3888*a^4*b*c^7*f*j*z^4 + 3888
*a^4*b*c^7*e*k*z^4 + 3888*a^4*b*c^7*d*l*z^4 + 3888*a^4*b*c^7*g*h*z^4 + 3888
*a^3*b*c^8*d*e*z^4 + 243*a*b^5*c^6*d*e*z^4 - 25272*a^4*b^3*c^5*j*m*z^4 + 97
20*a^4*b^3*c^5*k*l*z^4 + 6075*a^3*b^5*c^4*j*m*z^4 - 2673*a^3*b^5*c^4*k*l*z^
4 - 486*a^2*b^7*c^3*j*m*z^4 + 243*a^2*b^7*c^3*k*l*z^4 - 7776*a^4*b^2*c^6*h*
k*z^4 - 7776*a^4*b^2*c^6*g*l*z^4 - 7776*a^4*b^2*c^6*f*m*z^4 + 2430*a^3*b^4*
c^5*h*k*z^4 + 2430*a^3*b^4*c^5*g*l*z^4 + 2430*a^3*b^4*c^5*f*m*z^4 - 243*a^2
*b^6*c^4*h*k*z^4 - 243*a^2*b^6*c^4*g*l*z^4 - 243*a^2*b^6*c^4*f*m*z^4 - 1944
*a^3*b^3*c^6*f*j*z^4 - 1944*a^3*b^3*c^6*e*k*z^4 - 1944*a^3*b^3*c^6*d*l*z^4
+ 243*a^2*b^5*c^5*f*j*z^4 + 243*a^2*b^5*c^5*e*k*z^4 + 243*a^2*b^5*c^5*d*l*z
^4 - 1944*a^3*b^3*c^6*g*h*z^4 + 243*a^2*b^5*c^5*g*h*z^4 + 3888*a^3*b^2*c^7*
e*g*z^4 + 3888*a^3*b^2*c^7*d*h*z^4 - 486*a^2*b^4*c^6*e*g*z^4 - 486*a^2*b^4*
c^6*d*h*z^4 - 1944*a^2*b^3*c^7*d*e*z^4 + 7776*a^5*c^7*h*k*z^4 + 7776*a^5*c^
7*g*l*z^4 + 7776*a^5*c^7*f*m*z^4 - 7776*a^4*c^8*e*g*z^4 - 7776*a^4*c^8*d*h*
z^4 - 13608*a^5*b^2*c^5*m^2*z^4 + 11421*a^4*b^4*c^4*m^2*z^4 - 2916*a^3*b^6*
c^3*m^2*z^4 + 243*a^2*b^8*c^2*m^2*z^4 + 13608*a^4*b^2*c^6*j^2*z^4 - 3159*a^
3*b^4*c^5*j^2*z^4 + 243*a^2*b^6*c^4*j^2*z^4 + 1944*a^3*b^2*c^7*f^2*z^4 - 24
3*a^2*b^4*c^6*f^2*z^4 - 3888*a^6*c^6*m^2*z^4 - 19440*a^5*c^7*j^2*z^4 - 3888
*a^4*c^8*f^2*z^4 + 3078*a^4*b^4*c^3*k*l*m*z^3 - 2592*a^5*b^2*c^4*k*l*m*z^3
- 891*a^3*b^6*c^2*k*l*m*z^3 - 4536*a^4*b^3*c^4*j*k*l*z^3 + 1053*a^3*b^5*c^3
*j*k*l*z^3 - 81*a^2*b^7*c^2*j*k*l*z^3 - 2592*a^4*b^3*c^4*h*k*m*z^3 - 2592*a
^4*b^3*c^4*g*l*m*z^3 + 810*a^3*b^5*c^3*h*k*m*z^3 + 810*a^3*b^5*c^3*g*l*m*z^
3 - 81*a^2*b^7*c^2*h*k*m*z^3 - 81*a^2*b^7*c^2*g*l*m*z^3 + 7776*a^4*b^2*c^5*
f*j*m*z^3 + 3888*a^4*b^2*c^5*h*j*k*z^3 + 3888*a^4*b^2*c^5*g*j*l*z^3 - 3888*
a^4*b^2*c^5*f*k*l*z^3 - 2916*a^3*b^4*c^4*f*j*m*z^3 + 1458*a^3*b^4*c^4*f*k*l
*z^3 - 972*a^3*b^4*c^4*h*j*k*z^3 - 972*a^3*b^4*c^4*g*j*l*z^3 - 486*a^3*b^4*
c^4*e*k*m*z^3 - 486*a^3*b^4*c^4*d*l*m*z^3 + 324*a^2*b^6*c^3*f*j*m*z^3 - 162

*a^2*b^6*c^3*f*k*l*z^3 + 81*a^2*b^6*c^3*h*j*k*z^3 + 81*a^2*b^6*c^3*g*j*l*z^
3 + 81*a^2*b^6*c^3*e*k*m*z^3 + 81*a^2*b^6*c^3*d*l*m*z^3 - 486*a^3*b^4*c^4*g
*h*m*z^3 + 81*a^2*b^6*c^3*g*h*m*z^3 + 648*a^3*b^3*c^5*e*j*k*z^3 + 648*a^3*b
^3*c^5*d*j*l*z^3 - 81*a^2*b^5*c^4*e*j*k*z^3 - 81*a^2*b^5*c^4*d*j*l*z^3 + 25
92*a^3*b^3*c^5*e*g*m*z^3 + 2592*a^3*b^3*c^5*d*h*m*z^3 - 1296*a^3*b^3*c^5*f*
h*k*z^3 - 1296*a^3*b^3*c^5*f*g*l*z^3 - 1296*a^3*b^3*c^5*e*h*l*z^3 + 648*a^3
*b^3*c^5*g*h*j*z^3 - 324*a^2*b^5*c^4*e*g*m*z^3 - 324*a^2*b^5*c^4*d*h*m*z^3
+ 162*a^2*b^5*c^4*f*h*k*z^3 + 162*a^2*b^5*c^4*f*g*l*z^3 + 162*a^2*b^5*c^4*e
*h*l*z^3 - 81*a^2*b^5*c^4*g*h*j*z^3 + 5184*a^3*b^2*c^6*d*e*m*z^3 - 2592*a^3
*b^2*c^6*e*g*j*z^3 - 2592*a^3*b^2*c^6*d*h*j*z^3 - 2106*a^2*b^4*c^5*d*e*m*z^
3 + 1296*a^3*b^2*c^6*e*f*k*z^3 + 1296*a^3*b^2*c^6*d*g*k*z^3 + 1296*a^3*b^2*
c^6*d*f*l*z^3 + 324*a^2*b^4*c^5*e*g*j*z^3 + 324*a^2*b^4*c^5*d*h*j*z^3 - 162
*a^2*b^4*c^5*e*f*k*z^3 - 162*a^2*b^4*c^5*d*g*k*z^3 - 162*a^2*b^4*c^5*d*f*l*
z^3 + 1296*a^3*b^2*c^6*f*g*h*z^3 - 162*a^2*b^4*c^5*f*g*h*z^3 + 1944*a^2*b^3
*c^6*d*e*j*z^3 - 1296*a^2*b^2*c^7*d*e*f*z^3 + 81*a^2*b^8*c*k*l*m*z^3 + 6480
*a^5*b*c^5*j*k*l*z^3 + 2592*a^5*b*c^5*h*k*m*z^3 + 2592*a^5*b*c^5*g*l*m*z^3
- 1296*a^4*b*c^6*e*j*k*z^3 - 1296*a^4*b*c^6*d*j*l*z^3 - 5184*a^4*b*c^6*e*g*
m*z^3 - 5184*a^4*b*c^6*d*h*m*z^3 + 2592*a^4*b*c^6*f*h*k*z^3 + 2592*a^4*b*c^
6*f*g*l*z^3 + 2592*a^4*b*c^6*e*h*l*z^3 - 1296*a^4*b*c^6*g*h*j*z^3 + 243*a*b
^6*c^4*d*e*m*z^3 - 3888*a^3*b*c^7*d*e*j*z^3 - 243*a*b^5*c^5*d*e*j*z^3 + 162
*a*b^4*c^6*d*e*f*z^3 - 2592*a^6*c^5*k*l*m*z^3 - 5184*a^5*c^6*h*j*k*z^3 - 51
84*a^5*c^6*g*j*l*z^3 - 5184*a^5*c^6*f*j*m*z^3 + 2592*a^5*c^6*f*k*l*z^3 + 25
92*a^5*c^6*e*k*m*z^3 + 2592*a^5*c^6*d*l*m*z^3 + 2592*a^5*c^6*g*h*m*z^3 + 51
84*a^4*c^7*e*g*j*z^3 + 5184*a^4*c^7*d*h*j*z^3 - 2592*a^4*c^7*e*f*k*z^3 - 25
92*a^4*c^7*d*g*k*z^3 - 2592*a^4*c^7*d*f*l*z^3 - 2592*a^4*c^7*d*e*m*z^3 - 25
92*a^4*c^7*f*g*h*z^3 + 2592*a^3*c^8*d*e*f*z^3 + 6480*a^5*b^2*c^4*j*m^2*z^3
+ 6480*a^4*b^3*c^4*j^2*m*z^3 - 5022*a^4*b^4*c^3*j*m^2*z^3 - 1296*a^3*b^5*c^
3*j^2*m*z^3 + 1134*a^3*b^6*c^2*j*m^2*z^3 + 81*a^2*b^7*c^2*j^2*m*z^3 + 2592*
a^4*b^3*c^4*h*l^2*z^3 - 1944*a^4*b^2*c^5*h^2*l*z^3 - 810*a^3*b^5*c^3*h*l^2*
z^3 + 729*a^3*b^4*c^4*h^2*l*z^3 + 81*a^2*b^7*c^2*h*l^2*z^3 - 81*a^2*b^6*c^3
*h^2*l*z^3 - 5184*a^4*b^3*c^4*f*m^2*z^3 + 1620*a^3*b^5*c^3*f*m^2*z^3 + 1296
*a^3*b^3*c^5*f^2*m*z^3 - 162*a^2*b^7*c^2*f*m^2*z^3 - 162*a^2*b^5*c^4*f^2*m*
z^3 - 1944*a^4*b^2*c^5*g*k^2*z^3 + 729*a^3*b^4*c^4*g*k^2*z^3 - 648*a^3*b^3*
c^5*g^2*k*z^3 - 81*a^2*b^6*c^3*g*k^2*z^3 + 81*a^2*b^5*c^4*g^2*k*z^3 - 1944*
a^4*b^2*c^5*e*l^2*z^3 + 729*a^3*b^4*c^4*e*l^2*z^3 + 648*a^3*b^2*c^6*e^2*l*z
^3 - 81*a^2*b^6*c^3*e*l^2*z^3 - 81*a^2*b^4*c^5*e^2*l*z^3 + 1296*a^3*b^3*c^5
*f*j^2*z^3 - 1296*a^3*b^2*c^6*f^2*j*z^3 - 162*a^2*b^5*c^4*f*j^2*z^3 + 162*a
^2*b^4*c^5*f^2*j*z^3 - 648*a^3*b^3*c^5*d*k^2*z^3 + 81*a^2*b^5*c^4*d*k^2*z^3
+ 648*a^3*b^2*c^6*e*h^2*z^3 - 81*a^2*b^4*c^5*e*h^2*z^3 - 648*a^2*b^2*c^7*d
^2*g*z^3 - 10368*a^5*b*c^5*j^2*m*z^3 - 81*a^2*b^8*c*j*m^2*z^3 - 2592*a^5*b*
c^5*h*l^2*z^3 + 5184*a^5*b*c^5*f*m^2*z^3 - 2592*a^4*b*c^6*f^2*m*z^3 + 1296*
a^4*b*c^6*g^2*k*z^3 - 2592*a^4*b*c^6*f*j^2*z^3 + 1296*a^4*b*c^6*d*k^2*z^3 +
81*a*b^4*c^6*d^2*g*z^3 + 2592*a^6*c^5*j*m^2*z^3 + 1296*a^5*c^6*h^2*l*z^3 +
1296*a^5*c^6*g*k^2*z^3 + 1296*a^5*c^6*e*l^2*z^3 - 1296*a^4*c^7*e^2*l*z^3 +
2592*a^4*c^7*f^2*j*z^3 - 2592*a^6*b*c^4*m^3*z^3 - 324*a^3*b^7*c*m^3*z^3 -

27*a^2*b^8*c*l^3*z^3 - 1296*a^4*c^7*e*h^2*z^3 - 864*a^5*b*c^5*k^3*z^3 + 129
6*a^3*c^8*d^2*g*z^3 + 432*a^4*b*c^6*h^3*z^3 + 27*a*b^4*c^6*e^3*z^3 - 432*a^
2*b*c^8*d^3*z^3 + 216*a*b^3*c^7*d^3*z^3 + 1134*a^4*b^5*c^2*m^3*z^3 - 432*a^
5*b^3*c^3*m^3*z^3 + 1512*a^5*b^2*c^4*l^3*z^3 - 1107*a^4*b^4*c^3*l^3*z^3 + 2
97*a^3*b^6*c^2*l^3*z^3 + 864*a^4*b^3*c^4*k^3*z^3 - 270*a^3*b^5*c^3*k^3*z^3
+ 27*a^2*b^7*c^2*k^3*z^3 - 2592*a^4*b^2*c^5*j^3*z^3 + 486*a^3*b^4*c^4*j^3*z
^3 - 27*a^2*b^6*c^3*j^3*z^3 - 216*a^3*b^3*c^5*h^3*z^3 + 27*a^2*b^5*c^4*h^3*
z^3 + 216*a^3*b^2*c^6*g^3*z^3 - 27*a^2*b^4*c^5*g^3*z^3 - 216*a^2*b^2*c^7*e^
3*z^3 - 432*a^6*c^5*l^3*z^3 + 27*a^2*b^9*m^3*z^3 + 4320*a^5*c^6*j^3*z^3 - 4
32*a^4*c^7*g^3*z^3 + 432*a^3*c^8*e^3*z^3 - 27*b^5*c^6*d^3*z^3 + 81*a^3*b^6*
c*j*k*l*m*z^2 - 1296*a^5*b*c^4*h*j*k*m*z^2 - 1296*a^5*b*c^4*g*j*l*m*z^2 + 1
296*a^5*b*c^4*f*k*l*m*z^2 - 81*a^2*b^7*c*f*k*l*m*z^2 + 2592*a^4*b*c^5*e*g*j
*m*z^2 + 2592*a^4*b*c^5*d*h*j*m*z^2 - 1296*a^4*b*c^5*f*h*j*k*z^2 - 1296*a^4
*b*c^5*f*g*j*l*z^2 - 1296*a^4*b*c^5*e*f*k*m*z^2 - 1296*a^4*b*c^5*d*f*l*m*z^
2 - 648*a^4*b*c^5*e*h*j*l*z^2 - 648*a^4*b*c^5*e*g*k*l*z^2 - 648*a^4*b*c^5*d
*h*k*l*z^2 - 648*a^4*b*c^5*d*g*k*m*z^2 - 1296*a^4*b*c^5*f*g*h*m*z^2 - 162*a
*b^6*c^3*d*e*j*m*z^2 + 81*a*b^6*c^3*d*e*k*l*z^2 + 1296*a^3*b*c^6*d*e*f*m*z^
2 - 648*a^3*b*c^6*d*f*g*k*z^2 - 648*a^3*b*c^6*d*e*h*k*z^2 - 648*a^3*b*c^6*d
*e*g*l*z^2 - 81*a*b^5*c^4*d*e*h*k*z^2 - 81*a*b^5*c^4*d*e*g*l*z^2 + 81*a*b^5
*c^4*d*e*f*m*z^2 - 81*a*b^4*c^5*d*e*f*j*z^2 + 81*a*b^4*c^5*d*e*g*h*z^2 + 64
8*a^5*b^2*c^3*j*k*l*m*z^2 - 567*a^4*b^4*c^2*j*k*l*m*z^2 - 1944*a^4*b^3*c^3*
f*k*l*m*z^2 + 729*a^3*b^5*c^2*f*k*l*m*z^2 + 648*a^4*b^3*c^3*h*j*k*m*z^2 + 6
48*a^4*b^3*c^3*g*j*l*m*z^2 - 81*a^3*b^5*c^2*h*j*k*m*z^2 - 81*a^3*b^5*c^2*g*
j*l*m*z^2 + 1944*a^4*b^2*c^4*f*j*k*l*z^2 - 729*a^3*b^4*c^3*f*j*k*l*z^2 + 64
8*a^4*b^2*c^4*e*j*k*m*z^2 + 648*a^4*b^2*c^4*d*j*l*m*z^2 - 81*a^3*b^4*c^3*e*
j*k*m*z^2 - 81*a^3*b^4*c^3*d*j*l*m*z^2 + 81*a^2*b^6*c^2*f*j*k*l*z^2 + 1296*
a^4*b^2*c^4*f*h*k*m*z^2 + 1296*a^4*b^2*c^4*f*g*l*m*z^2 + 648*a^4*b^2*c^4*g*
h*j*m*z^2 - 648*a^3*b^4*c^3*f*h*k*m*z^2 - 648*a^3*b^4*c^3*f*g*l*m*z^2 - 324
*a^4*b^2*c^4*g*h*k*l*z^2 - 324*a^4*b^2*c^4*e*h*l*m*z^2 + 81*a^3*b^4*c^3*g*h
*k*l*z^2 - 81*a^3*b^4*c^3*g*h*j*m*z^2 + 81*a^2*b^6*c^2*f*h*k*m*z^2 + 81*a^2
*b^6*c^2*f*g*l*m*z^2 - 1296*a^3*b^3*c^4*e*g*j*m*z^2 - 1296*a^3*b^3*c^4*d*h*
j*m*z^2 + 648*a^3*b^3*c^4*f*h*j*k*z^2 + 648*a^3*b^3*c^4*f*g*j*l*z^2 + 648*a
^3*b^3*c^4*e*f*k*m*z^2 + 648*a^3*b^3*c^4*d*f*l*m*z^2 + 486*a^3*b^3*c^4*e*g*
k*l*z^2 + 486*a^3*b^3*c^4*d*h*k*l*z^2 + 162*a^3*b^3*c^4*e*h*j*l*z^2 + 162*a
^3*b^3*c^4*d*g*k*m*z^2 + 162*a^2*b^5*c^3*e*g*j*m*z^2 + 162*a^2*b^5*c^3*d*h*
j*m*z^2 - 81*a^2*b^5*c^3*f*h*j*k*z^2 - 81*a^2*b^5*c^3*f*g*j*l*z^2 - 81*a^2*
b^5*c^3*e*g*k*l*z^2 - 81*a^2*b^5*c^3*e*f*k*m*z^2 - 81*a^2*b^5*c^3*d*h*k*l*z
^2 - 81*a^2*b^5*c^3*d*f*l*m*z^2 + 648*a^3*b^3*c^4*f*g*h*m*z^2 - 81*a^2*b^5*
c^3*f*g*h*m*z^2 - 3240*a^3*b^2*c^5*d*e*j*m*z^2 + 1620*a^3*b^2*c^5*d*e*k*l*z
^2 + 1377*a^2*b^4*c^4*d*e*j*m*z^2 - 648*a^3*b^2*c^5*e*f*j*k*z^2 - 648*a^3*b
^2*c^5*d*f*j*l*z^2 - 648*a^2*b^4*c^4*d*e*k*l*z^2 - 324*a^3*b^2*c^5*d*g*j*k*
z^2 + 81*a^2*b^4*c^4*e*f*j*k*z^2 + 81*a^2*b^4*c^4*d*f*j*l*z^2 + 972*a^3*b^2
*c^5*e*f*h*l*z^2 - 648*a^3*b^2*c^5*f*g*h*j*z^2 - 324*a^3*b^2*c^5*e*g*h*k*z^
2 - 324*a^3*b^2*c^5*d*g*h*l*z^2 - 162*a^2*b^4*c^4*e*f*h*l*z^2 + 81*a^2*b^4*
c^4*f*g*h*j*z^2 + 81*a^2*b^4*c^4*e*g*h*k*z^2 + 81*a^2*b^4*c^4*d*g*h*l*z^2 -

648*a^2*b^3*c^5*d*e*f*m*z^2 + 486*a^2*b^3*c^5*d*e*h*k*z^2 + 486*a^2*b^3*c^5*d*e*g*l*z^2 + 162*a^2*b^3*c^5*d*f*g*k*z^2 + 648*a^2*b^2*c^6*d*e*f*j*z^2 - 324*a^2*b^2*c^6*d*e*g*h*z^2 - 1296*a^6*b*c^3*k*l*m^2*z^2 - 81*a^4*b^5*c*k*l*m^2*z^2 - 1296*a^5*b*c^4*j^2*k*l*z^2 - 324*a^5*b*c^4*h^2*l*m*z^2 + 324*a^5*b*c^4*h*k^2*l*z^2 - 324*a^5*b*c^4*g*k^2*m*z^2 + 972*a^5*b*c^4*h*j*l^2*z^2 + 324*a^5*b*c^4*g*k*l^2*z^2 - 324*a^5*b*c^4*e*l^2*m*z^2 - 324*a^4*b*c^5*e^2*l*m*z^2 - 1944*a^5*b*c^4*f*j*m^2*z^2 + 1296*a^5*b*c^4*e*k*m^2*z^2 + 1296*a^5*b*c^4*d*l*m^2*z^2 + 648*a^4*b*c^5*f^2*j*m*z^2 + 81*a^2*b^7*c*f*j*m^2*z^2 + 1296*a^5*b*c^4*g*h*m^2*z^2 - 324*a^4*b*c^5*g^2*j*k*z^2 + 324*a^4*b*c^5*g^2*h*l*z^2 + 972*a^4*b*c^5*f*h^2*l*z^2 + 324*a^4*b*c^5*g*h^2*k*z^2 - 324*a^4*b*c^5*e*h^2*m*z^2 - 324*a^4*b*c^5*d*j*k^2*z^2 - 324*a^3*b*c^6*d^2*j*k*z^2 + 972*a^4*b*c^5*f*g*k^2*z^2 + 972*a^3*b*c^6*d^2*g*m*z^2 + 324*a^4*b*c^5*e*h*k^2*z^2 + 324*a^3*b*c^6*d^2*h*l*z^2 + 81*a*b^5*c^4*d^2*g*m*z^2 + 972*a^4*b*c^5*e*f*l^2*z^2 + 324*a^4*b*c^5*d*g*l^2*z^2 - 324*a^3*b*c^6*e^2*h*j*z^2 + 324*a^3*b*c^6*e^2*g*k*z^2 - 324*a^3*b*c^6*e^2*f*l*z^2 - 1296*a^4*b*c^5*d*e*m^2*z^2 + 81*a*b^7*c^2*d*e*m^2*z^2 - 324*a^3*b*c^6*d*g^2*j*z^2 - 81*a*b^4*c^5*d^2*g*j*z^2 + 81*a*b^4*c^5*d^2*e*l*z^2 + 324*a^3*b*c^6*e*g^2*h*z^2 + 81*a*b^4*c^5*d*e^2*k*z^2 + 1296*a^3*b*c^6*d*e*j^2*z^2 - 324*a^3*b*c^6*e*f*h^2*z^2 + 324*a^3*b*c^6*d*g*h^2*z^2 + 81*a*b^5*c^4*d*e*j^2*z^2 - 324*a^2*b*c^7*d^2*f*g*z^2 + 324*a^2*b*c^7*d^2*e*h*z^2 + 81*a*b^3*c^6*d^2*f*g*z^2 - 81*a*b^3*c^6*d^2*e*h*z^2 + 324*a^2*b*c^7*d*e^2*g*z^2 - 81*a*b^3*c^6*d*e^2*g*z^2 + 1296*a^6*c^4*j*k*l*m*z^2 - 1296*a^5*c^5*f*j*k*l*z^2 - 1296*a^5*c^5*e*j*k*m*z^2 - 1296*a^5*c^5*d*j*l*m*z^2 - 1296*a^5*c^5*g*h*j*m*z^2 + 1296*a^5*c^5*e*h*l*m*z^2 + 1296*a^4*c^6*e*f*j*k*z^2 + 1296*a^4*c^6*d*g*j*k*z^2 + 1296*a^4*c^6*d*f*j*l*z^2 - 1296*a^4*c^6*d*e*k*l*z^2 + 1296*a^4*c^6*d*e*j*m*z^2 + 1296*a^4*c^6*f*g*h*j*z^2 - 1296*a^4*c^6*e*f*h*l*z^2 - 1296*a^3*c^7*d*e*f*j*z^2 + 648*a^5*b^3*c^2*k*l*m^2*z^2 + 648*a^4*b^3*c^3*j^2*k*l*z^2 + 486*a^5*b^2*c^3*h*l^2*m*z^2 - 81*a^4*b^4*c^2*h*l^2*m*z^2 + 81*a^4*b^3*c^3*h^2*l*m*z^2 - 81*a^3*b^5*c^2*j^2*k*l*z^2 - 162*a^4*b^2*c^4*g^2*k*m*z^2 - 81*a^4*b^3*c^3*h*k^2*l*z^2 + 81*a^4*b^3*c^3*g*k^2*m*z^2 - 567*a^4*b^3*c^3*h*j*l^2*z^2 + 486*a^4*b^2*c^4*h^2*j*l*z^2 - 81*a^4*b^3*c^3*g*k*l^2*z^2 + 81*a^4*b^3*c^3*e*l^2*m*z^2 + 81*a^3*b^5*c^2*h*j*l^2*z^2 - 81*a^3*b^4*c^3*h^2*j*l*z^2 + 81*a^3*b^3*c^4*e^2*l*m*z^2 + 2430*a^4*b^3*c^3*f*j*m^2*z^2 - 2268*a^4*b^2*c^4*f*j^2*m*z^2 - 810*a^3*b^5*c^2*f*j*m^2*z^2 + 810*a^3*b^4*c^3*f*j^2*m*z^2 - 648*a^4*b^3*c^3*e*k*m^2*z^2 - 648*a^4*b^3*c^3*d*l*m^2*z^2 - 648*a^4*b^2*c^4*h*j^2*k*z^2 - 648*a^4*b^2*c^4*g*j^2*l*z^2 - 162*a^3*b^3*c^4*f^2*j*m*z^2 + 81*a^3*b^5*c^2*e*k*m^2*z^2 + 81*a^3*b^5*c^2*d*l*m^2*z^2 + 81*a^3*b^4*c^3*h*j^2*k*z^2 + 81*a^3*b^4*c^3*g*j^2*l*z^2 - 81*a^2*b^6*c^2*f*j^2*m*z^2 - 648*a^4*b^3*c^3*g*h*m^2*z^2 + 486*a^4*b^2*c^4*g*j*k^2*z^2 - 486*a^4*b^2*c^4*e*k^2*l*z^2 + 486*a^3*b^2*c^5*d^2*k*m*z^2 - 162*a^4*b^2*c^4*d*k^2*m*z^2 + 81*a^3*b^5*c^2*g*h*m^2*z^2 - 81*a^3*b^4*c^3*g*j*k^2*z^2 + 81*a^3*b^4*c^3*e*k^2*l*z^2 + 81*a^3*b^3*c^4*g^2*j*k*z^2 - 81*a^2*b^4*c^4*d^2*k*m*z^2 + 486*a^4*b^2*c^4*e*j*l^2*z^2 - 486*a^4*b^2*c^4*d*k*l^2*z^2 - 162*a^3*b^2*c^5*e^2*j*l*z^2 - 81*a^3*b^4*c^3*e*j*l^2*z^2 + 81*a^3*b^4*c^3*d*k*l^2*z^2 - 81*a^3*b^3*c^4*g^2*h*l*z^2 - 1458*a^4*b^2*c^4*f*h*l^2*z^2 + 648*a^3*b^4*c^3*f*h*l^2*z^2 -

567*a^3*b^3*c^4*f*h^2*l*z^2 + 486*a^3*b^2*c^5*e^2*h*m*z^2 - 81*a^3*b^3*c^4
*g*h^2*k*z^2 + 81*a^3*b^3*c^4*e*h^2*m*z^2 - 81*a^2*b^6*c^2*f*h*l^2*z^2 + 81
*a^2*b^5*c^3*f*h^2*l*z^2 - 81*a^2*b^4*c^4*e^2*h*m*z^2 - 1296*a^4*b^2*c^4*e*
g*m^2*z^2 - 1296*a^4*b^2*c^4*d*h*m^2*z^2 + 648*a^3*b^4*c^3*e*g*m^2*z^2 + 64
8*a^3*b^4*c^3*d*h*m^2*z^2 + 81*a^3*b^3*c^4*d*j*k^2*z^2 - 81*a^2*b^6*c^2*e*g
*m^2*z^2 - 81*a^2*b^6*c^2*d*h*m^2*z^2 + 81*a^2*b^3*c^5*d^2*j*k*z^2 - 567*a^
3*b^3*c^4*f*g*k^2*z^2 - 567*a^2*b^3*c^5*d^2*g*m*z^2 + 486*a^3*b^2*c^5*f*g^2
*k*z^2 - 486*a^3*b^2*c^5*e*g^2*l*z^2 + 486*a^3*b^2*c^5*d*g^2*m*z^2 - 81*a^3
*b^3*c^4*e*h*k^2*z^2 + 81*a^2*b^5*c^3*f*g*k^2*z^2 - 81*a^2*b^4*c^4*f*g^2*k*
z^2 + 81*a^2*b^4*c^4*e*g^2*l*z^2 - 81*a^2*b^4*c^4*d*g^2*m*z^2 - 81*a^2*b^3*
c^5*d^2*h*l*z^2 - 567*a^3*b^3*c^4*e*f*l^2*z^2 - 486*a^3*b^2*c^5*d*h^2*k*z^2
 - 162*a^3*b^2*c^5*e*h^2*j*z^2 - 81*a^3*b^3*c^4*d*g*l^2*z^2 + 81*a^2*b^5*c^
3*e*f*l^2*z^2 + 81*a^2*b^4*c^4*d*h^2*k*z^2 + 81*a^2*b^3*c^5*e^2*h*j*z^2 - 8
1*a^2*b^3*c^5*e^2*g*k*z^2 + 81*a^2*b^3*c^5*e^2*f*l*z^2 + 1944*a^3*b^3*c^4*d
*e*m^2*z^2 - 729*a^2*b^5*c^3*d*e*m^2*z^2 + 648*a^3*b^2*c^5*e*g*j^2*z^2 + 64
8*a^3*b^2*c^5*d*h*j^2*z^2 - 81*a^2*b^4*c^4*e*g*j^2*z^2 - 81*a^2*b^4*c^4*d*h
*j^2*z^2 + 486*a^3*b^2*c^5*d*f*k^2*z^2 + 486*a^2*b^2*c^6*d^2*g*j*z^2 - 486*
a^2*b^2*c^6*d^2*e*l*z^2 - 162*a^2*b^2*c^6*d^2*f*k*z^2 - 81*a^2*b^4*c^4*d*f*
k^2*z^2 + 81*a^2*b^3*c^5*d*g^2*j*z^2 - 486*a^2*b^2*c^6*d*e^2*k*z^2 - 81*a^2
*b^3*c^5*e*g^2*h*z^2 - 648*a^2*b^3*c^5*d*e*j^2*z^2 - 162*a^2*b^2*c^6*e^2*f*
h*z^2 + 81*a^2*b^3*c^5*e*f*h^2*z^2 - 81*a^2*b^3*c^5*d*g*h^2*z^2 - 162*a^2*b
^2*c^6*d*f*g^2*z^2 - 189*a^5*b^3*c^2*l^3*m*z^2 + 162*a^5*b^2*c^3*k^3*m*z^2
 - 27*a^4*b^4*c^2*k^3*m*z^2 - 702*a^4*b^3*c^3*j^3*m*z^2 - 81*a^3*b^6*c*j^2*m
^2*z^2 + 81*a^3*b^5*c^2*j^3*m*z^2 - 54*a^5*b^3*c^2*j*m^3*z^2 - 486*a^5*b^2*
c^3*j*l^3*z^2 + 216*a^4*b^4*c^2*j*l^3*z^2 - 189*a^4*b^3*c^3*j*k^3*z^2 - 54*
a^4*b^2*c^4*h^3*m*z^2 + 27*a^3*b^5*c^2*j*k^3*z^2 + 27*a^3*b^3*c^4*g^3*m*z^2
 - 810*a^4*b^4*c^2*f*m^3*z^2 + 540*a^5*b^2*c^3*f*m^3*z^2 - 324*a^3*b^2*c^5*
f^3*m*z^2 + 54*a^2*b^4*c^4*f^3*m*z^2 + 675*a^4*b^3*c^3*f*l^3*z^2 - 243*a^3*
b^5*c^2*f*l^3*z^2 - 189*a^2*b^3*c^5*e^3*m*z^2 + 27*a^3*b^3*c^4*h^3*j*z^2 -
486*a^4*b^2*c^4*f*k^3*z^2 - 486*a^2*b^2*c^6*d^3*m*z^2 + 216*a^3*b^4*c^3*f*k
^3*z^2 - 54*a^3*b^2*c^5*g^3*j*z^2 - 27*a^2*b^6*c^2*f*k^3*z^2 - 270*a^3*b^3*
c^4*f*j^3*z^2 - 54*a^2*b^3*c^5*f^3*j*z^2 + 27*a^2*b^5*c^3*f*j^3*z^2 + 162*a
^2*b^2*c^6*e^3*j*z^2 + 162*a^3*b^2*c^5*f*h^3*z^2 - 27*a^2*b^4*c^4*f*h^3*z^2
 + 27*a^2*b^3*c^5*f*g^3*z^2 + 81*a*b^2*c^7*d^2*e^2*z^2 - 648*a^6*c^4*h*l^2*
m*z^2 + 648*a^5*c^5*g^2*k*m*z^2 - 648*a^5*c^5*h^2*j*l*z^2 + 1296*a^5*c^5*h*
j^2*k*z^2 + 1296*a^5*c^5*g*j^2*l*z^2 + 1296*a^5*c^5*f*j^2*m*z^2 - 648*a^5*c
^5*g*j*k^2*z^2 + 648*a^5*c^5*e*k^2*l*z^2 + 648*a^5*c^5*d*k^2*m*z^2 - 648*a^
4*c^6*d^2*k*m*z^2 - 648*a^5*c^5*e*j*l^2*z^2 + 648*a^5*c^5*d*k*l^2*z^2 + 648
*a^4*c^6*e^2*j*l*z^2 + 324*a^6*b*c^3*l^3*m*z^2 + 27*a^4*b^5*c*l^3*m*z^2 + 6
48*a^5*c^5*f*h*l^2*z^2 - 648*a^4*c^6*e^2*h*m*z^2 + 1512*a^5*b*c^4*j^3*m*z^2
 + 1080*a^6*b*c^3*j*m^3*z^2 - 162*a^4*b^5*c*j*m^3*z^2 - 648*a^4*c^6*f*g^2*k
*z^2 + 648*a^4*c^6*e*g^2*l*z^2 - 648*a^4*c^6*d*g^2*m*z^2 - 27*a^3*b^6*c*j*l
^3*z^2 + 648*a^4*c^6*e*h^2*j*z^2 + 648*a^4*c^6*d*h^2*k*z^2 + 324*a^5*b*c^4*
j*k^3*z^2 - 1296*a^4*c^6*e*g*j^2*z^2 - 1296*a^4*c^6*d*h*j^2*z^2 - 108*a^4*b
*c^5*g^3*m*z^2 - 648*a^4*c^6*d*f*k^2*z^2 - 648*a^3*c^7*d^2*g*j*z^2 + 648*a^

3*c^7*d^2*f*k*z^2 + 648*a^3*c^7*d^2*e*l*z^2 + 270*a^3*b^6*c*f*m^3*z^2 + 648
*a^3*c^7*d*e^2*k*z^2 - 540*a^5*b*c^4*f*l^3*z^2 + 324*a^3*b*c^6*e^3*m*z^2 -
108*a^4*b*c^5*h^3*j*z^2 + 27*a^2*b^7*c*f*l^3*z^2 + 27*a*b^5*c^4*e^3*m*z^2 +
 648*a^3*c^7*e^2*f*h*z^2 + 216*a*b^4*c^5*d^3*m*z^2 + 648*a^4*b*c^5*f*j^3*z^
2 + 216*a^3*b*c^6*f^3*j*z^2 + 648*a^3*c^7*d*f*g^2*z^2 - 27*a*b^4*c^5*e^3*j*
z^2 + 324*a^2*b*c^7*d^3*j*z^2 - 189*a*b^3*c^6*d^3*j*z^2 - 108*a^3*b*c^6*f*g
^3*z^2 - 108*a^2*b*c^7*e^3*f*z^2 + 27*a*b^3*c^6*e^3*f*z^2 + 162*a*b^2*c^7*d
^3*f*z^2 - 1134*a^5*b^2*c^3*j^2*m^2*z^2 + 648*a^4*b^4*c^2*j^2*m^2*z^2 + 81*
a^5*b^2*c^3*k^2*l^2*z^2 + 162*a^4*b^2*c^4*f^2*m^2*z^2 + 81*a^4*b^2*c^4*h^2*
k^2*z^2 + 81*a^4*b^2*c^4*g^2*l^2*z^2 + 162*a^3*b^2*c^5*f^2*j^2*z^2 + 81*a^3
*b^2*c^5*e^2*k^2*z^2 + 81*a^3*b^2*c^5*d^2*l^2*z^2 + 81*a^3*b^2*c^5*g^2*h^2*
z^2 + 81*a^2*b^2*c^6*e^2*g^2*z^2 + 81*a^2*b^2*c^6*d^2*h^2*z^2 - 216*a^6*c^4
*k^3*m*z^2 + 216*a^6*c^4*j*l^3*z^2 + 27*a^3*b^7*j*m^3*z^2 + 216*a^5*c^5*h^3
*m*z^2 + 432*a^6*c^4*f*m^3*z^2 + 432*a^4*c^6*f^3*m*z^2 - 27*b^6*c^4*d^3*m*z
^2 - 27*a^2*b^8*f*m^3*z^2 + 216*a^5*c^5*f*k^3*z^2 + 216*a^4*c^6*g^3*j*z^2 +
 216*a^3*c^7*d^3*m*z^2 + 216*a^5*b^4*c*m^4*z^2 - 216*a^3*c^7*e^3*j*z^2 + 27
*b^5*c^5*d^3*j*z^2 - 216*a^4*c^6*f*h^3*z^2 - 27*b^4*c^6*d^3*f*z^2 - 216*a^2
*c^8*d^3*f*z^2 - 648*a^6*c^4*j^2*m^2*z^2 - 324*a^6*c^4*k^2*l^2*z^2 - 648*a^
5*c^5*f^2*m^2*z^2 - 324*a^5*c^5*h^2*k^2*z^2 - 324*a^5*c^5*g^2*l^2*z^2 - 648
*a^4*c^6*f^2*j^2*z^2 - 324*a^4*c^6*e^2*k^2*z^2 - 324*a^4*c^6*d^2*l^2*z^2 -
405*a^6*b^2*c^2*m^4*z^2 - 324*a^4*c^6*g^2*h^2*z^2 - 324*a^3*c^7*e^2*g^2*z^2
 - 324*a^3*c^7*d^2*h^2*z^2 + 243*a^4*b^2*c^4*j^4*z^2 - 27*a^3*b^4*c^3*j^4*z
^2 - 324*a^2*c^8*d^2*e^2*z^2 + 27*a^2*b^2*c^6*f^4*z^2 - 108*a^7*c^3*m^4*z^2
 - 27*a^4*b^6*m^4*z^2 - 540*a^5*c^5*j^4*z^2 - 108*a^3*c^7*f^4*z^2 - 216*a^5
*b*c^3*f*j*k*l*m*z - 54*a^3*b^5*c*f*j*k*l*m*z + 27*a^3*b^5*c*g*h*k*l*m*z -
27*a^2*b^6*c*e*g*k*l*m*z - 27*a^2*b^6*c*d*h*k*l*m*z + 432*a^4*b*c^4*d*g*j*k
*m*z - 432*a^4*b*c^4*d*e*k*l*m*z + 216*a^4*b*c^4*e*g*j*k*l*z + 216*a^4*b*c^
4*e*f*j*k*m*z + 216*a^4*b*c^4*d*h*j*k*l*z + 216*a^4*b*c^4*d*f*j*l*m*z + 216
*a^4*b*c^4*f*g*h*j*m*z - 27*a*b^6*c^2*d*e*j*k*l*z - 27*a*b^6*c^2*d*e*h*k*m*
z - 27*a*b^6*c^2*d*e*g*l*m*z + 216*a^3*b*c^5*d*e*h*j*k*z + 216*a^3*b*c^5*d*
e*g*j*l*z - 216*a^3*b*c^5*d*e*f*j*m*z + 27*a*b^5*c^3*d*e*h*j*k*z + 27*a*b^5
*c^3*d*e*g*j*l*z + 27*a*b^5*c^3*d*e*g*h*m*z - 27*a*b^4*c^4*d*e*g*h*j*z + 27
*a*b^7*c*d*e*k*l*m*z + 270*a^4*b^3*c^2*f*j*k*l*m*z - 108*a^4*b^3*c^2*g*h*k*
l*m*z - 216*a^4*b^2*c^3*f*h*j*k*m*z - 216*a^4*b^2*c^3*f*g*j*l*m*z - 216*a^4
*b^2*c^3*e*g*k*l*m*z - 216*a^4*b^2*c^3*d*h*k*l*m*z + 162*a^3*b^4*c^2*e*g*k*
l*m*z + 162*a^3*b^4*c^2*d*h*k*l*m*z + 108*a^4*b^2*c^3*g*h*j*k*l*z + 108*a^4
*b^2*c^3*e*h*j*l*m*z + 54*a^3*b^4*c^2*f*h*j*k*m*z + 54*a^3*b^4*c^2*f*g*j*l*
m*z - 27*a^3*b^4*c^2*g*h*j*k*l*z + 540*a^3*b^3*c^3*d*e*k*l*m*z - 216*a^2*b^
5*c^2*d*e*k*l*m*z - 162*a^3*b^3*c^3*e*g*j*k*l*z - 162*a^3*b^3*c^3*d*h*j*k*l
*z - 108*a^3*b^3*c^3*d*g*j*k*m*z - 54*a^3*b^3*c^3*e*f*j*k*m*z - 54*a^3*b^3*
c^3*d*f*j*l*m*z + 27*a^2*b^5*c^2*e*g*j*k*l*z + 27*a^2*b^5*c^2*d*h*j*k*l*z -
 108*a^3*b^3*c^3*e*g*h*k*m*z - 108*a^3*b^3*c^3*d*g*h*l*m*z - 54*a^3*b^3*c^3
*f*g*h*j*m*z + 27*a^2*b^5*c^2*e*g*h*k*m*z + 27*a^2*b^5*c^2*d*g*h*l*m*z - 54
0*a^3*b^2*c^4*d*e*j*k*l*z + 216*a^2*b^4*c^3*d*e*j*k*l*z - 216*a^3*b^2*c^4*d
*e*h*k*m*z - 216*a^3*b^2*c^4*d*e*g*l*m*z + 162*a^2*b^4*c^3*d*e*h*k*m*z + 16

2*a^2*b^4*c^3*d*e*g*l*m*z + 108*a^3*b^2*c^4*e*g*h*j*k*z - 108*a^3*b^2*c^4*e
*f*h*j*l*z + 108*a^3*b^2*c^4*d*g*h*j*l*z + 108*a^3*b^2*c^4*d*f*g*k*m*z - 27
*a^2*b^4*c^3*e*g*h*j*k*z - 27*a^2*b^4*c^3*d*g*h*j*l*z - 162*a^2*b^3*c^4*d*e
*h*j*k*z - 162*a^2*b^3*c^4*d*e*g*j*l*z + 54*a^2*b^3*c^4*d*e*f*j*m*z - 108*a
^2*b^3*c^4*d*e*g*h*m*z + 108*a^2*b^2*c^5*d*e*g*h*j*z + 324*a^6*b*c^2*j*k*l*
m^2*z - 81*a^5*b^3*c*j*k*l*m^2*z + 27*a^4*b^4*c*j^2*k*l*m*z - 27*a^4*b^4*c*
h*k^2*l*m*z - 27*a^4*b^4*c*g*k*l^2*m*z + 216*a^5*b*c^3*h*j^2*k*m*z + 216*a^
5*b*c^3*g*j^2*l*m*z + 54*a^4*b^4*c*f*k*l*m^2*z + 27*a^4*b^4*c*h*j*k*m^2*z +
 27*a^4*b^4*c*g*j*l*m^2*z + 27*a^2*b^6*c*f^2*k*l*m*z + 216*a^5*b*c^3*e*k^2*
l*m*z - 108*a^5*b*c^3*h*j*k^2*l*z + 27*a^3*b^5*c*e*k^2*l*m*z + 216*a^5*b*c^
3*d*k*l^2*m*z + 216*a^4*b*c^4*e^2*j*l*m*z - 108*a^5*b*c^3*g*j*k*l^2*z + 27*
a^3*b^5*c*d*k*l^2*m*z - 324*a^5*b*c^3*e*j*k*m^2*z - 324*a^5*b*c^3*d*j*l*m^2
*z - 216*a^5*b*c^3*f*h*l^2*m*z - 108*a^4*b*c^4*f^2*j*k*l*z - 27*a^3*b^5*c*e
*j*k*m^2*z - 27*a^3*b^5*c*d*j*l*m^2*z - 324*a^5*b*c^3*g*h*j*m^2*z + 216*a^5
*b*c^3*f*h*k*m^2*z + 216*a^5*b*c^3*f*g*l*m^2*z + 216*a^5*b*c^3*e*h*l*m^2*z
- 216*a^4*b*c^4*f^2*h*k*m*z - 216*a^4*b*c^4*f^2*g*l*m*z - 27*a^3*b^5*c*g*h*
j*m^2*z + 216*a^4*b*c^4*e*g^2*l*m*z - 108*a^4*b*c^4*g^2*h*j*l*z - 216*a^4*b
*c^4*f*h^2*j*l*z + 216*a^4*b*c^4*e*h^2*j*m*z + 216*a^4*b*c^4*d*h^2*k*m*z -
108*a^4*b*c^4*g*h^2*j*k*z - 432*a^4*b*c^4*e*g*j^2*m*z - 432*a^4*b*c^4*d*h*j
^2*m*z + 216*a^4*b*c^4*f*h*j^2*k*z + 216*a^4*b*c^4*f*g*j^2*l*z + 27*a^2*b^6
*c*e*g*j*m^2*z + 27*a^2*b^6*c*d*h*j*m^2*z - 432*a^3*b*c^5*d^2*g*j*m*z - 216
*a^4*b*c^4*f*g*j*k^2*z + 216*a^3*b*c^5*d^2*f*k*m*z + 216*a^3*b*c^5*d^2*e*l*
m*z - 108*a^4*b*c^4*e*h*j*k^2*z - 108*a^4*b*c^4*d*g*k^2*l*z - 108*a^3*b*c^5
*d^2*h*j*l*z + 108*a^3*b*c^5*d^2*g*k*l*z - 54*a*b^5*c^3*d^2*g*j*m*z + 27*a*
b^5*c^3*d^2*g*k*l*z + 27*a*b^5*c^3*d^2*e*l*m*z - 216*a^4*b*c^4*e*f*j*l^2*z
+ 216*a^3*b*c^5*d*e^2*k*m*z - 108*a^4*b*c^4*d*g*j*l^2*z - 108*a^3*b*c^5*e^2
*g*j*k*z + 27*a*b^5*c^3*d*e^2*k*m*z + 324*a^4*b*c^4*d*e*j*m^2*z + 216*a^3*b
*c^5*e^2*f*h*m*z - 108*a^4*b*c^4*e*g*h*l^2*z + 108*a^3*b*c^5*e^2*g*h*l*z +
108*a^3*b*c^5*e*f^2*j*k*z + 108*a^3*b*c^5*d*f^2*j*l*z + 27*a*b^6*c^2*d*e*j^
2*m*z - 216*a^3*b*c^5*e*f^2*h*l*z + 108*a^3*b*c^5*f^2*g*h*j*z - 27*a*b^4*c^
4*d^2*e*j*l*z + 216*a^3*b*c^5*d*f*g^2*m*z - 108*a^3*b*c^5*e*g^2*h*j*z + 54*
a*b^4*c^4*d^2*f*g*m*z - 27*a*b^4*c^4*d^2*g*h*k*z - 27*a*b^4*c^4*d^2*e*h*m*z
 - 27*a*b^4*c^4*d*e^2*j*k*z - 108*a^3*b*c^5*d*g*h^2*j*z + 54*a*b^4*c^4*d*e^
2*h*l*z + 27*a*b^6*c^2*d*e*h*l^2*z - 27*a*b^5*c^3*d*e*h^2*l*z - 27*a*b^4*c^
4*d*e^2*g*m*z - 27*a*b^4*c^4*d*e*f^2*m*z + 216*a^2*b*c^6*d^2*f*g*j*z - 108*
a^3*b*c^5*d*e*g*k^2*z - 108*a^2*b*c^6*d^2*e*h*j*z + 108*a^2*b*c^6*d^2*e*g*k
*z - 54*a*b^3*c^5*d^2*f*g*j*z - 27*a*b^5*c^3*d*e*g*k^2*z + 27*a*b^4*c^4*d*e
*g^2*k*z + 27*a*b^3*c^5*d^2*e*h*j*z - 27*a*b^3*c^5*d^2*e*g*k*z - 108*a^2*b*
c^6*d*e^2*g*j*z + 27*a*b^3*c^5*d*e^2*g*j*z - 108*a^2*b*c^6*d*e*f^2*j*z + 27
*a*b^3*c^5*d*e*f^2*j*z - 432*a^5*c^4*e*h*j*l*m*z + 432*a^4*c^5*d*e*j*k*l*z
+ 432*a^4*c^5*e*f*h*j*l*z - 432*a^4*c^5*d*f*g*k*m*z - 27*a*b^7*c*d*e*j*m^2*
z - 54*a^5*b^2*c^2*j^2*k*l*m*z + 108*a^5*b^2*c^2*h*k^2*l*m*z + 108*a^5*b^2*
c^2*g*k*l^2*m*z - 54*a^5*b^2*c^2*h*j*l^2*m*z + 378*a^4*b^2*c^3*f^2*k*l*m*z
- 270*a^5*b^2*c^2*f*k*l*m^2*z - 189*a^3*b^4*c^2*f^2*k*l*m*z - 108*a^5*b^2*c
^2*h*j*k*m^2*z - 108*a^5*b^2*c^2*g*j*l*m^2*z - 54*a^4*b^3*c^2*h*j^2*k*m*z -

54*a^4*b^3*c^2*g*j^2*l*m*z - 162*a^4*b^3*c^2*e*k^2*l*m*z + 54*a^4*b^2*c^3*
g^2*j*k*m*z + 27*a^4*b^3*c^2*h*j*k^2*l*z - 162*a^4*b^3*c^2*d*k*l^2*m*z + 10
8*a^4*b^2*c^3*g^2*h*l*m*z - 54*a^3*b^3*c^3*e^2*j*l*m*z + 27*a^4*b^3*c^2*g*j
*k*l^2*z - 27*a^3*b^4*c^2*g^2*h*l*m*z - 270*a^4*b^2*c^3*f*j^2*k*l*z + 189*a
^4*b^3*c^2*e*j*k*m^2*z + 189*a^4*b^3*c^2*d*j*l*m^2*z - 162*a^4*b^2*c^3*e*j^
2*k*m*z - 162*a^4*b^2*c^3*d*j^2*l*m*z + 135*a^3*b^3*c^3*f^2*j*k*l*z + 108*a
^4*b^2*c^3*g*h^2*k*m*z + 54*a^4*b^3*c^2*f*h*l^2*m*z - 54*a^4*b^2*c^3*f*h^2*
l*m*z + 54*a^3*b^4*c^2*f*j^2*k*l*z - 27*a^3*b^4*c^2*g*h^2*k*m*z + 27*a^3*b^
4*c^2*e*j^2*k*m*z + 27*a^3*b^4*c^2*d*j^2*l*m*z - 27*a^2*b^5*c^2*f^2*j*k*l*z
 - 270*a^3*b^2*c^4*d^2*j*k*m*z + 189*a^4*b^3*c^2*g*h*j*m^2*z - 162*a^4*b^2*
c^3*g*h*j^2*m*z + 162*a^4*b^2*c^3*e*j*k^2*l*z + 162*a^3*b^3*c^3*f^2*h*k*m*z
 + 162*a^3*b^3*c^3*f^2*g*l*m*z - 54*a^4*b^3*c^2*f*h*k*m^2*z - 54*a^4*b^3*c^
2*f*g*l*m^2*z - 54*a^4*b^3*c^2*e*h*l*m^2*z + 54*a^4*b^2*c^3*d*j*k^2*m*z + 5
4*a^2*b^4*c^3*d^2*j*k*m*z + 27*a^3*b^4*c^2*g*h*j^2*m*z - 27*a^3*b^4*c^2*e*j
*k^2*l*z - 27*a^2*b^5*c^2*f^2*h*k*m*z - 27*a^2*b^5*c^2*f^2*g*l*m*z + 162*a^
4*b^2*c^3*d*j*k*l^2*z - 162*a^3*b^3*c^3*e*g^2*l*m*z + 108*a^4*b^2*c^3*e*h*k
^2*m*z + 108*a^3*b^2*c^4*d^2*h*l*m*z - 54*a^4*b^2*c^3*f*g*k^2*m*z - 27*a^3*
b^4*c^2*e*h*k^2*m*z - 27*a^3*b^4*c^2*d*j*k*l^2*z + 27*a^3*b^3*c^3*g^2*h*j*l
*z + 27*a^2*b^5*c^2*e*g^2*l*m*z - 27*a^2*b^4*c^3*d^2*h*l*m*z + 270*a^4*b^2*
c^3*f*h*j*l^2*z - 270*a^3*b^2*c^4*e^2*h*j*m*z - 162*a^4*b^2*c^3*e*h*k*l^2*z
 - 162*a^3*b^3*c^3*d*h^2*k*m*z + 162*a^3*b^2*c^4*e^2*h*k*l*z + 108*a^4*b^2*
c^3*d*g*l^2*m*z + 108*a^3*b^2*c^4*e^2*g*k*m*z - 54*a^4*b^2*c^3*e*f*l^2*m*z
- 54*a^3*b^4*c^2*f*h*j*l^2*z + 54*a^3*b^3*c^3*f*h^2*j*l*z - 54*a^3*b^3*c^3*
e*h^2*j*m*z + 54*a^3*b^2*c^4*e^2*f*l*m*z + 54*a^2*b^4*c^3*e^2*h*j*m*z + 27*
a^3*b^4*c^2*e*h*k*l^2*z - 27*a^3*b^4*c^2*d*g*l^2*m*z + 27*a^3*b^3*c^3*g*h^2
*j*k*z + 27*a^2*b^5*c^2*d*h^2*k*m*z - 27*a^2*b^4*c^3*e^2*h*k*l*z - 27*a^2*b
^4*c^3*e^2*g*k*m*z + 432*a^4*b^2*c^3*e*g*j*m^2*z + 432*a^4*b^2*c^3*d*h*j*m^
2*z - 270*a^4*b^2*c^3*d*g*k*m^2*z - 216*a^3*b^4*c^2*e*g*j*m^2*z - 216*a^3*b
^4*c^2*d*h*j*m^2*z + 216*a^3*b^3*c^3*e*g*j^2*m*z + 216*a^3*b^3*c^3*d*h*j^2*
m*z - 162*a^3*b^2*c^4*e*f^2*k*m*z - 162*a^3*b^2*c^4*d*f^2*l*m*z - 108*a^3*b
^2*c^4*f^2*h*j*k*z - 108*a^3*b^2*c^4*f^2*g*j*l*z + 54*a^4*b^2*c^3*e*f*k*m^2
*z + 54*a^4*b^2*c^3*d*f*l*m^2*z + 54*a^3*b^4*c^2*d*g*k*m^2*z - 54*a^3*b^3*c
^3*f*h*j^2*k*z - 54*a^3*b^3*c^3*f*g*j^2*l*z - 27*a^2*b^5*c^2*e*g*j^2*m*z -
27*a^2*b^5*c^2*d*h*j^2*m*z + 27*a^2*b^4*c^3*f^2*h*j*k*z + 27*a^2*b^4*c^3*f^
2*g*j*l*z + 27*a^2*b^4*c^3*e*f^2*k*m*z + 27*a^2*b^4*c^3*d*f^2*l*m*z + 324*a
^2*b^3*c^4*d^2*g*j*m*z - 270*a^3*b^2*c^4*d*g^2*j*m*z - 162*a^3*b^2*c^4*f^2*
g*h*m*z + 162*a^3*b^2*c^4*e*g^2*j*l*z - 162*a^2*b^3*c^4*d^2*e*l*m*z - 135*a
^2*b^3*c^4*d^2*g*k*l*z + 108*a^3*b^2*c^4*d*g^2*k*l*z + 54*a^4*b^2*c^3*f*g*h
*m^2*z + 54*a^3*b^3*c^3*f*g*j*k^2*z - 54*a^3*b^2*c^4*f*g^2*j*k*z + 54*a^2*b
^4*c^3*d*g^2*j*m*z - 54*a^2*b^3*c^4*d^2*f*k*m*z + 27*a^3*b^3*c^3*e*h*j*k^2*
z + 27*a^3*b^3*c^3*d*g*k^2*l*z + 27*a^2*b^4*c^3*f^2*g*h*m*z - 27*a^2*b^4*c^
3*e*g^2*j*l*z - 27*a^2*b^4*c^3*d*g^2*k*l*z + 27*a^2*b^3*c^4*d^2*h*j*l*z + 1
62*a^3*b^2*c^4*d*h^2*j*k*z - 162*a^2*b^3*c^4*d*e^2*k*m*z + 108*a^3*b^2*c^4*
e*g^2*h*m*z + 54*a^3*b^3*c^3*e*f*j*l^2*z + 27*a^3*b^3*c^3*d*g*j*l^2*z - 27*
a^2*b^4*c^3*e*g^2*h*m*z - 27*a^2*b^4*c^3*d*h^2*j*k*z + 27*a^2*b^3*c^4*e^2*g

*j*k*z - 621*a^3*b^3*c^3*d*e*j*m^2*z + 594*a^3*b^2*c^4*d*e*j^2*m*z + 243*a^2*b^5*c^2*d*e*j*m^2*z - 243*a^2*b^4*c^3*d*e*j^2*m*z + 135*a^3*b^3*c^3*e*g*h*l^2*z - 108*a^3*b^2*c^4*e*g*h^2*l*z + 108*a^3*b^2*c^4*d*g*h^2*m*z + 54*a^3*b^2*c^4*e*f*j^2*k*z + 54*a^3*b^2*c^4*e*f*h^2*m*z + 54*a^3*b^2*c^4*d*g*j^2*k*z + 54*a^3*b^2*c^4*d*f*j^2*l*z - 54*a^2*b^3*c^4*e^2*f*h*m*z - 27*a^2*b^5*c^2*e*g*h*l^2*z + 27*a^2*b^4*c^3*e*g*h^2*l*z - 27*a^2*b^4*c^3*d*g*h^2*m*z - 27*a^2*b^3*c^4*e^2*g*h*l*z - 27*a^2*b^3*c^4*e*f^2*j*k*z - 27*a^2*b^3*c^4*d*f^2*j*l*z + 162*a^2*b^2*c^5*d^2*e*j*l*z + 54*a^3*b^2*c^4*f*g*h*j^2*z - 54*a^3*b^2*c^4*d*f*j*k^2*z + 54*a^2*b^3*c^4*e*f^2*h*l*z + 54*a^2*b^2*c^5*d^2*f*j*k*z - 27*a^2*b^3*c^4*f^2*g*h*j*z - 270*a^2*b^2*c^5*d^2*f*g*m*z - 162*a^3*b^2*c^4*d*g*h*k^2*z + 162*a^2*b^2*c^5*d^2*g*h*k*z + 162*a^2*b^2*c^5*d*e^2*j*k*z + 108*a^2*b^2*c^5*d^2*e*h*m*z - 54*a^2*b^3*c^4*d*f*g^2*m*z + 27*a^2*b^4*c^3*d*g*h*k^2*z + 27*a^2*b^3*c^4*e*g^2*h*j*z + 270*a^3*b^2*c^4*d*e*h*l^2*z - 270*a^2*b^2*c^5*d*e^2*h*l*z - 162*a^2*b^4*c^3*d*e*h*l^2*z + 108*a^2*b^3*c^4*d*e*h^2*l*z + 108*a^2*b^2*c^5*d*e^2*g*m*z + 54*a^2*b^2*c^5*e^2*f*h*j*z + 27*a^2*b^3*c^4*d*g*h^2*j*z + 162*a^2*b^2*c^5*d*e*f^2*m*z - 54*a^3*b^2*c^4*d*e*f*m^2*z - 54*a^2*b^2*c^5*d*f^2*g*k*z + 135*a^2*b^3*c^4*d*e*g*k^2*z - 108*a^2*b^2*c^5*d*e*g^2*k*z + 54*a^2*b^2*c^5*d*f*g^2*j*z - 54*a^2*b^2*c^5*d*e*f*j^2*z - 9*a*b^7*c*d*e*l^3*z - 36*a*b*c^7*d^3*e*g*z - 108*a^6*b*c^2*k^2*l^2*m*z + 27*a^5*b^3*c*k^2*l^2*m*z - 18*a^5*b^2*c^2*j*k^3*m*z - 27*a^4*b^3*c^2*j^3*k*l*z - 108*a^5*b*c^3*h^2*k^2*m*z - 108*a^5*b*c^3*g^2*l^2*m*z + 108*a^5*b*c^3*h^2*k*l^2*z + 108*a^5*b*c^3*g^2*k*m^2*z + 90*a^5*b^2*c^2*f*l^3*m*z - 18*a^5*b^2*c^2*h*k*l^3*z + 18*a^4*b^2*c^3*h^3*k*l*z + 18*a^4*b^2*c^3*h^3*j*m*z - 108*a^5*b*c^3*h*j^2*l^2*z + 18*a^4*b^3*c^2*f*k^3*m*z - 18*a^3*b^3*c^3*g^3*j*m*z - 9*a^4*b^3*c^2*g*k^3*l*z + 9*a^3*b^3*c^3*g^3*k*l*z + 252*a^4*b^2*c^3*f*j^3*m*z + 216*a^5*b*c^3*f*j^2*m^2*z + 180*a^3*b^2*c^4*f^3*j*m*z - 108*a^4*b*c^4*e^2*k^2*m*z - 108*a^4*b*c^4*d^2*l^2*m*z + 90*a^5*b^2*c^2*e*k*m^3*z + 90*a^5*b^2*c^2*d*l*m^3*z - 90*a^3*b^2*c^4*f^3*k*l*z + 54*a^3*b^5*c*f*j^2*m^2*z - 54*a^3*b^4*c^2*f*j^3*m*z + 36*a^5*b^2*c^2*f*j*m^3*z + 36*a^4*b^2*c^3*h*j^3*k*z + 36*a^4*b^2*c^3*g*j^3*l*z - 36*a^2*b^4*c^3*f^3*j*m*z - 27*a^2*b^6*c*f^2*j*m^2*z + 18*a^2*b^4*c^3*f^3*k*l*z - 216*a^4*b*c^4*d^2*k*m^2*z + 108*a^5*b*c^3*d*k^2*m^2*z - 108*a^4*b^3*c^2*f*j*l^3*z - 108*a^4*b*c^4*g^2*h^2*m*z + 108*a^2*b^3*c^4*e^3*j*m*z + 90*a^5*b^2*c^2*g*h*m^3*z + 54*a^4*b^3*c^2*e*k*l^3*z - 54*a^2*b^3*c^4*e^3*k*l*z + 234*a^2*b^2*c^5*d^3*j*m*z - 144*a^2*b^2*c^5*d^3*k*l*z + 90*a^4*b^2*c^3*f*j*k^3*z - 72*a^4*b^2*c^3*d*k^3*l*z + 27*a^4*b^3*c^2*g*h*l^3*z - 27*a^3*b^3*c^3*g*h^3*l*z - 18*a^3*b^4*c^2*f*j*k^3*z + 9*a^3*b^4*c^2*d*k^3*l*z + 216*a^4*b*c^4*f^2*h*l^2*z - 216*a^4*b*c^4*e^2*h*m^2*z + 108*a^4*b*c^4*g^2*h*k^2*z - 18*a^4*b^2*c^3*g*h*k^3*z + 18*a^3*b^2*c^4*g^3*h*k*z + 18*a^3*b^2*c^4*f*g^3*m*z + 9*a^3*b^4*c^2*g*h*k^3*z - 9*a^3*b^3*c^3*e*j^3*k*z - 9*a^3*b^3*c^3*d*j^3*l*z - 144*a^4*b^3*c^2*e*g*m^3*z - 144*a^4*b^3*c^2*d*h*m^3*z - 108*a^3*b*c^5*e^2*g^2*m*z + 108*a^3*b*c^5*d^2*j^2*k*z - 108*a^3*b*c^5*d^2*h^2*m*z - 18*a^2*b^3*c^4*f^3*h*k*z - 18*a^2*b^3*c^4*f^3*g*l*z - 9*a^3*b^3*c^3*g*h*j^3*z - 216*a^4*b*c^4*d*g^2*m^2*z + 144*a^4*b^2*c^3*e*g*l^3*z - 126*a^3*b^2*c^4*d*h^3*l*z - 108*a^4*b*c^4*d*h^2*l^2*z - 108*a^3*b*c^5*f^2*g^2*k*z - 108*a^3*b*c^5*e^2*h^2*k*z

- 90*a^2*b^2*c^5*e^3*f*m*z + 72*a^2*b^2*c^5*e^3*g*l*z - 63*a^3*b^4*c^2*e*g
*l^3*z - 36*a^3*b^4*c^2*d*h*l^3*z + 27*a^2*b^4*c^3*d*h^3*l*z + 27*a*b^6*c^2
*d^2*g*m^2*z - 18*a^4*b^2*c^3*d*h*l^3*z - 18*a^3*b^2*c^4*f*h^3*j*z - 18*a^3
*b^2*c^4*e*h^3*k*z + 18*a^2*b^2*c^5*e^3*h*k*z + 108*a^3*b*c^5*e^2*h*j^2*z +
54*a^3*b^3*c^3*d*h*k^3*z + 27*a^3*b^3*c^3*e*g*k^3*z - 27*a^2*b^3*c^4*e*g^3
*k*z + 27*a^2*b^3*c^4*d*g^3*l*z - 27*a*b^4*c^4*d^2*g^2*l*z - 9*a^2*b^5*c^2*
e*g*k^3*z - 9*a^2*b^5*c^2*d*h*k^3*z + 207*a^3*b^4*c^2*d*e*m^3*z - 108*a^2*b
*c^6*d^2*e^2*m*z - 90*a^4*b^2*c^3*d*e*m^3*z - 72*a^3*b^2*c^4*e*g*j^3*z - 72
*a^3*b^2*c^4*d*h*j^3*z + 27*a*b^3*c^5*d^2*e^2*m*z + 18*a^2*b^2*c^5*e*f^3*k*
z + 18*a^2*b^2*c^5*d*f^3*l*z + 9*a^2*b^4*c^3*e*g*j^3*z + 9*a^2*b^4*c^3*d*h*
j^3*z - 216*a^3*b*c^5*d*e^2*l^2*z - 198*a^3*b^3*c^3*d*e*l^3*z + 108*a^3*b*c
^5*d*g^2*j^2*z - 108*a^3*b*c^5*d*f^2*k^2*z + 72*a^2*b^5*c^2*d*e*l^3*z - 27*
a*b^5*c^3*d*e^2*l^2*z + 27*a*b^4*c^4*d^2*g*j^2*z + 18*a^2*b^2*c^5*f^3*g*h*z
+ 144*a^3*b^2*c^4*d*e*k^3*z - 63*a^2*b^4*c^3*d*e*k^3*z + 27*a*b^4*c^4*d^2*
e*k^2*z - 9*a^2*b^3*c^4*e*g*h^3*z - 108*a^2*b*c^6*d^2*g^2*h*z + 81*a^2*b^3*
c^4*d*e*j^3*z + 27*a*b^3*c^5*d^2*g^2*h*z - 27*a*b^2*c^6*d^2*e^2*j*z - 18*a^
2*b^2*c^5*d*g^3*h*z + 108*a^2*b*c^6*d*e^2*h^2*z - 27*a*b^3*c^5*d*e^2*h^2*z
+ 27*a*b^2*c^6*d^2*f^2*g*z - 18*a^2*b^2*c^5*d*e*h^3*z - 216*a^6*c^3*j^2*k*l
*m*z + 216*a^6*c^3*h*j*l^2*m*z + 216*a^6*c^3*f*k*l*m^2*z - 216*a^5*c^4*f^2*
k*l*m*z - 216*a^5*c^4*g^2*j*k*m*z + 216*a^5*c^4*f*j^2*k*l*z + 216*a^5*c^4*f
*h^2*l*m*z + 216*a^5*c^4*e*j^2*k*m*z + 216*a^5*c^4*d*j^2*l*m*z + 216*a^5*c^
4*g*h*j^2*m*z - 216*a^5*c^4*e*j*k^2*l*z - 216*a^5*c^4*d*j*k^2*m*z + 216*a^4
*c^5*d^2*j*k*m*z - 18*a^6*b^2*c*k*l*m^3*z + 216*a^5*c^4*f*g*k^2*m*z - 216*a
^5*c^4*d*j*k*l^2*z - 72*a^6*b*c^2*j*l^3*m*z + 18*a^5*b^3*c*j*l^3*m*z - 216*
a^5*c^4*f*h*j*l^2*z + 216*a^5*c^4*e*h*k*l^2*z + 216*a^5*c^4*e*f*l^2*m*z - 2
16*a^4*c^5*e^2*h*k*l*z + 216*a^4*c^5*e^2*h*j*m*z - 216*a^4*c^5*e^2*f*l*m*z
- 216*a^5*c^4*e*f*k*m^2*z + 216*a^5*c^4*d*g*k*m^2*z - 216*a^5*c^4*d*f*l*m^2
*z + 216*a^4*c^5*e*f^2*k*m*z + 216*a^4*c^5*d*f^2*l*m*z + 108*a^5*b*c^3*j^3*
k*l*z - 216*a^5*c^4*f*g*h*m^2*z + 216*a^4*c^5*f^2*g*h*m*z + 216*a^4*c^5*f*g
^2*j*k*z - 216*a^4*c^5*e*g^2*j*l*z + 216*a^4*c^5*d*g^2*j*m*z - 72*a^6*b*c^2
*h*k*m^3*z - 72*a^6*b*c^2*g*l*m^3*z + 54*a^5*b^3*c*h*k*m^3*z + 54*a^5*b^3*c
*g*l*m^3*z - 216*a^4*c^5*d*h^2*j*k*z - 18*a^4*b^4*c*f*l^3*m*z + 9*a^4*b^4*c
*h*k*l^3*z - 216*a^4*c^5*e*f*j^2*k*z - 216*a^4*c^5*e*f*h^2*m*z - 216*a^4*c^
5*d*g*j^2*k*z - 216*a^4*c^5*d*f*j^2*l*z - 216*a^4*c^5*d*e*j^2*m*z - 72*a^5*
b*c^3*f*k^3*m*z + 72*a^4*b*c^4*g^3*j*m*z + 36*a^5*b*c^3*g*k^3*l*z - 36*a^4*
b*c^4*g^3*k*l*z - 216*a^4*c^5*f*g*h*j^2*z + 216*a^4*c^5*d*f*j*k^2*z - 216*a
^3*c^6*d^2*f*j*k*z - 216*a^3*c^6*d^2*e*j*l*z + 72*a^4*b^4*c*f*j*m^3*z - 63*
a^4*b^4*c*e*k*m^3*z - 63*a^4*b^4*c*d*l*m^3*z + 216*a^4*c^5*d*g*h*k^2*z - 21
6*a^3*c^6*d^2*g*h*k*z + 216*a^3*c^6*d^2*f*g*m*z - 216*a^3*c^6*d*e^2*j*k*z +
144*a^5*b*c^3*f*j*l^3*z - 144*a^3*b*c^5*e^3*j*m*z - 72*a^5*b*c^3*e*k*l^3*z
+ 72*a^3*b*c^5*e^3*k*l*z - 63*a^4*b^4*c*g*h*m^3*z + 18*a^3*b^5*c*f*j*l^3*z
- 18*a*b^5*c^3*e^3*j*m*z - 9*a^3*b^5*c*e*k*l^3*z + 9*a*b^5*c^3*e^3*k*l*z -
216*a^4*c^5*d*e*h*l^2*z - 216*a^3*c^6*e^2*f*h*j*z + 216*a^3*c^6*d*e^2*h*l*
z - 126*a*b^4*c^4*d^3*j*m*z + 108*a^4*b*c^4*g*h^3*l*z + 63*a*b^4*c^4*d^3*k*
l*z + 36*a^5*b*c^3*g*h*l^3*z - 9*a^3*b^5*c*g*h*l^3*z + 216*a^4*c^5*d*e*f*m^

2*z + 216*a^3*c^6*d*f^2*g*k*z - 216*a^3*c^6*d*e*f^2*m*z + 36*a^4*b*c^4*e*j^
3*k*z + 36*a^4*b*c^4*d*j^3*l*z - 216*a^3*c^6*d*f*g^2*j*z + 72*a^3*b^5*c*e*g
*m^3*z + 72*a^3*b^5*c*d*h*m^3*z + 72*a^3*b*c^5*f^3*h*k*z + 72*a^3*b*c^5*f^3
*g*l*z + 36*a^4*b*c^4*g*h*j^3*z + 18*a*b^4*c^4*e^3*f*m*z + 9*a^2*b^6*c*e*g*
l^3*z + 9*a^2*b^6*c*d*h*l^3*z - 9*a*b^4*c^4*e^3*h*k*z - 9*a*b^4*c^4*e^3*g*l
*z + 216*a^3*c^6*d*e*f*j^2*z - 144*a^2*b*c^6*d^3*f*m*z + 108*a^3*b*c^5*e*g^
3*k*z - 108*a^3*b*c^5*d*g^3*l*z + 108*a*b^3*c^5*d^3*f*m*z - 72*a^4*b*c^4*d*
h*k^3*z + 72*a^2*b*c^6*d^3*h*k*z - 54*a*b^3*c^5*d^3*h*k*z + 36*a^4*b*c^4*e*
g*k^3*z - 36*a^2*b*c^6*d^3*g*l*z - 27*a*b^3*c^5*d^3*g*l*z - 81*a^2*b^6*c*d*
e*m^3*z + 216*a^4*b*c^4*d*e*l^3*z + 72*a^2*b*c^6*e^3*f*j*z + 72*a^2*b*c^6*d
*e^3*l*z - 18*a*b^3*c^5*e^3*f*j*z - 18*a*b^3*c^5*d*e^3*l*z - 90*a*b^2*c^6*d
^3*f*j*z + 72*a*b^2*c^6*d^3*e*k*z + 36*a^3*b*c^5*e*g*h^3*z - 36*a^2*b*c^6*e
^3*g*h*z + 9*a*b^6*c^2*d*e*k^3*z + 9*a*b^3*c^5*e^3*g*h*z - 180*a^3*b*c^5*d*
e*j^3*z + 18*a*b^2*c^6*d^3*g*h*z - 9*a*b^5*c^3*d*e*j^3*z + 18*a*b^2*c^6*d*e
^3*h*z + 9*a*b^4*c^4*d*e*h^3*z + 36*a^2*b*c^6*d*e*g^3*z - 9*a*b^3*c^5*d*e*g
^3*z - 18*a*b^2*c^6*d*e*f^3*z + 27*a^5*b^2*c^2*h^2*l*m^2*z - 27*a^5*b^2*c^2
*j*k^2*l^2*z + 27*a^4*b^3*c^2*h^2*k^2*m*z + 27*a^4*b^3*c^2*g^2*l^2*m*z + 27
*a^5*b^2*c^2*g*k^2*m^2*z - 27*a^4*b^3*c^2*h^2*k*l^2*z - 27*a^4*b^3*c^2*g^2*
k*m^2*z - 135*a^4*b^2*c^3*e^2*l*m^2*z + 27*a^5*b^2*c^2*e*l^2*m^2*z + 27*a^4
*b^3*c^2*h*j^2*l^2*z - 27*a^4*b^2*c^3*h^2*j^2*l*z + 27*a^3*b^4*c^2*e^2*l*m^
2*z - 270*a^4*b^3*c^2*f*j^2*m^2*z - 270*a^4*b^2*c^3*f^2*j*m^2*z + 162*a^3*b
^4*c^2*f^2*j*m^2*z - 108*a^3*b^3*c^3*f^2*j^2*m*z - 27*a^4*b^2*c^3*h^2*j*k^2
*z - 27*a^4*b^2*c^3*g^2*j*l^2*z + 27*a^3*b^3*c^3*e^2*k^2*m*z + 27*a^3*b^3*c
^3*d^2*l^2*m*z + 27*a^2*b^5*c^2*f^2*j^2*m*z + 162*a^3*b^3*c^3*d^2*k*m^2*z -
 27*a^4*b^3*c^2*d*k^2*m^2*z - 27*a^4*b^2*c^3*g*j^2*k^2*z + 27*a^3*b^3*c^3*g
^2*h^2*m*z - 27*a^2*b^5*c^2*d^2*k*m^2*z + 162*a^3*b^2*c^4*d^2*k^2*l*z - 108
*a^4*b^2*c^3*g*h^2*l^2*z - 27*a^4*b^2*c^3*e*j^2*l^2*z + 27*a^3*b^4*c^2*g*h^
2*l^2*z + 27*a^3*b^2*c^4*e^2*j^2*l*z - 27*a^2*b^4*c^3*d^2*k^2*l*z - 162*a^3
*b^3*c^3*f^2*h*l^2*z + 162*a^3*b^3*c^3*e^2*h*m^2*z - 135*a^4*b^2*c^3*e*h^2*
m^2*z + 135*a^3*b^2*c^4*f^2*h^2*l*z + 27*a^3*b^4*c^2*e*h^2*m^2*z - 27*a^3*b
^3*c^3*g^2*h*k^2*z - 27*a^3*b^2*c^4*e^2*j*k^2*z - 27*a^3*b^2*c^4*d^2*j*l^2*
z + 27*a^2*b^5*c^2*f^2*h*l^2*z - 27*a^2*b^5*c^2*e^2*h*m^2*z - 27*a^2*b^4*c^
3*f^2*h^2*l*z - 27*a^3*b^2*c^4*g^2*h^2*j*z + 27*a^2*b^3*c^4*e^2*g^2*m*z - 2
7*a^2*b^3*c^4*d^2*j^2*k*z + 27*a^2*b^3*c^4*d^2*h^2*m*z + 351*a^3*b^2*c^4*d^
2*g*m^2*z - 189*a^2*b^4*c^3*d^2*g*m^2*z + 162*a^3*b^3*c^3*d*g^2*m^2*z - 162
*a^3*b^2*c^4*e^2*g*l^2*z + 135*a^3*b^3*c^3*d*h^2*l^2*z + 135*a^3*b^2*c^4*f^
2*g*k^2*z - 27*a^2*b^5*c^2*d*h^2*l^2*z - 27*a^2*b^5*c^2*d*g^2*m^2*z - 27*a^
2*b^4*c^3*f^2*g*k^2*z + 27*a^2*b^4*c^3*e^2*g*l^2*z + 27*a^2*b^3*c^4*f^2*g^2
*k*z + 27*a^2*b^3*c^4*e^2*h^2*k*z + 135*a^3*b^2*c^4*e*f^2*l^2*z - 108*a^3*b
^2*c^4*e*g^2*k^2*z + 108*a^2*b^2*c^5*d^2*g^2*l*z + 27*a^3*b^2*c^4*e*h^2*j^2
*z + 27*a^2*b^4*c^3*e*g^2*k^2*z - 27*a^2*b^4*c^3*e*f^2*l^2*z - 27*a^2*b^3*c
^4*e^2*h*j^2*z - 27*a^2*b^2*c^5*e^2*f^2*l*z - 27*a^2*b^2*c^5*e^2*g^2*j*z -
27*a^2*b^2*c^5*d^2*h^2*j*z + 162*a^2*b^3*c^4*d*e^2*l^2*z - 135*a^2*b^2*c^5*
d^2*g*j^2*z - 27*a^2*b^3*c^4*d*g^2*j^2*z + 27*a^2*b^3*c^4*d*f^2*k^2*z - 162
*a^2*b^2*c^5*d^2*e*k^2*z - 27*a^2*b^2*c^5*e*f^2*h^2*z - 72*a^7*c^2*k*l*m^3*

z + 9*a^5*b^4*k*l*m^3*z + 72*a^6*c^3*j*k^3*m*z - 72*a^6*c^3*h*k*l^3*z - 72*
a^6*c^3*f*l^3*m*z - 72*a^5*c^4*h^3*k*l*z - 72*a^5*c^4*h^3*j*m*z - 9*a^4*b^5
*h*k*m^3*z - 9*a^4*b^5*g*l*m^3*z - 144*a^6*c^3*f*j*m^3*z - 144*a^5*c^4*h*j^
3*k*z - 144*a^5*c^4*g*j^3*l*z - 144*a^5*c^4*f*j^3*m*z - 144*a^4*c^5*f^3*j*m
*z + 72*a^6*c^3*e*k*m^3*z + 72*a^6*c^3*d*l*m^3*z + 72*a^4*c^5*f^3*k*l*z + 7
2*a^6*c^3*g*h*m^3*z + 18*b^6*c^3*d^3*j*m*z - 18*a^3*b^6*f*j*m^3*z - 9*b^6*c
^3*d^3*k*l*z + 9*a^3*b^6*e*k*m^3*z + 9*a^3*b^6*d*l*m^3*z + 144*a^5*c^4*d*k^
3*l*z + 144*a^3*c^6*d^3*k*l*z - 72*a^5*c^4*f*j*k^3*z - 72*a^3*c^6*d^3*j*m*z
 + 9*a^3*b^6*g*h*m^3*z - 72*a^5*c^4*g*h*k^3*z - 72*a^4*c^5*g^3*h*k*z - 72*a
^4*c^5*f*g^3*m*z - 108*a^5*b*c^3*j^4*m*z + 63*a^6*b^2*c*j*m^4*z + 36*a^6*b*
c^2*k*l^4*z - 9*a^5*b^3*c*k*l^4*z - 144*a^5*c^4*e*g*l^3*z - 144*a^3*c^6*e^3
*g*l*z + 72*a^5*c^4*d*h*l^3*z + 72*a^4*c^5*f*h^3*j*z + 72*a^4*c^5*e*h^3*k*z
 + 72*a^4*c^5*d*h^3*l*z + 72*a^3*c^6*e^3*h*k*z + 72*a^3*c^6*e^3*f*m*z - 18*
b^5*c^4*d^3*f*m*z + 9*b^5*c^4*d^3*h*k*z + 9*b^5*c^4*d^3*g*l*z - 9*a^2*b^7*e
*g*m^3*z - 9*a^2*b^7*d*h*m^3*z + 144*a^4*c^5*e*g*j^3*z + 144*a^4*c^5*d*h*j^
3*z - 72*a^5*c^4*d*e*m^3*z - 72*a^3*c^6*e*f^3*k*z - 72*a^3*c^6*d*f^3*l*z +
144*a^6*b*c^2*f*m^4*z - 108*a^5*b^3*c*f*m^4*z - 72*a^3*c^6*f^3*g*h*z + 36*a
^5*b*c^3*h*k^4*z - 36*a^3*b*c^5*f^4*m*z + 18*b^4*c^5*d^3*f*j*z - 9*b^4*c^5*
d^3*e*k*z + 9*a^4*b^4*c*g*l^4*z - 144*a^4*c^5*d*e*k^3*z - 144*a^2*c^7*d^3*e
*k*z + 72*a^2*c^7*d^3*f*j*z - 9*b^4*c^5*d^3*g*h*z + 72*a^3*c^6*d*g^3*h*z +
72*a^2*c^7*d^3*g*h*z - 72*a^5*b*c^3*d*l^4*z - 72*a^4*b*c^4*f*j^4*z + 45*a*b
^2*c^6*d^4*l*z - 36*a^2*b*c^6*e^4*k*z - 9*a^3*b^5*c*d*l^4*z + 9*a*b^3*c^5*e
^4*k*z - 72*a^3*c^6*d*e*h^3*z - 72*a^2*c^7*d*e^3*h*z + 9*b^3*c^6*d^3*e*g*z
+ 72*a^2*c^7*d*e*f^3*z + 36*a^3*b*c^5*d*h^4*z - 9*a*b^2*c^6*e^4*g*z + 36*a*
b*c^7*d^3*f^2*z + 90*a^5*b^2*c^2*j^3*m^2*z + 45*a^5*b^2*c^2*j^2*l^3*z + 9*a
^4*b^3*c^2*j^2*k^3*z - 9*a^4*b^3*c^2*h^3*m^2*z - 45*a^4*b^2*c^3*g^3*m^2*z +
 9*a^3*b^4*c^2*g^3*m^2*z + 198*a^4*b^3*c^2*f^2*m^3*z - 108*a^3*b^3*c^3*f^3*
m^2*z + 18*a^2*b^5*c^2*f^3*m^2*z - 117*a^4*b^2*c^3*f^2*l^3*z + 117*a^3*b^2*
c^4*e^3*m^2*z + 63*a^3*b^4*c^2*f^2*l^3*z - 63*a^2*b^4*c^3*e^3*m^2*z - 171*a
^2*b^3*c^4*d^3*m^2*z - 54*a^3*b^3*c^3*f^2*k^3*z + 9*a^3*b^2*c^4*g^3*j^2*z +
 9*a^2*b^5*c^2*f^2*k^3*z + 18*a^3*b^2*c^4*f^2*j^3*z + 18*a^2*b^3*c^4*f^3*j^
2*z - 9*a^2*b^4*c^3*f^2*j^3*z - 45*a^2*b^2*c^5*e^3*j^2*z + 9*a^2*b^3*c^4*f^
2*h^3*z - 9*a^2*b^2*c^5*f^2*g^3*z + 9*a*b^8*d*e*m^3*z - 36*a*b*c^7*d^4*h*z
- 108*a^6*c^3*h^2*l*m^2*z + 108*a^6*c^3*j*k^2*l^2*z - 108*a^6*c^3*g*k^2*m^2
*z - 108*a^6*c^3*e*l^2*m^2*z + 108*a^5*c^4*h^2*j^2*l*z + 108*a^5*c^4*e^2*l*
m^2*z + 216*a^5*c^4*f^2*j*m^2*z + 108*a^5*c^4*h^2*j*k^2*z + 108*a^5*c^4*g^2
*j*l^2*z + 108*a^5*c^4*g*j^2*k^2*z - 216*a^4*c^5*d^2*k^2*l*z + 108*a^5*c^4*
e*j^2*l^2*z - 108*a^4*c^5*e^2*j^2*l*z - 9*a^6*b^2*c*l^3*m^2*z + 108*a^5*c^4
*e*h^2*m^2*z - 108*a^4*c^5*f^2*h^2*l*z + 108*a^4*c^5*e^2*j*k^2*z + 108*a^4*
c^5*d^2*j*l^2*z - 144*a^6*b*c^2*j^2*m^3*z + 108*a^4*c^5*g^2*h^2*j*z - 27*a^
4*b^4*c*j^3*m^2*z + 27*a^4*b^3*c^2*j^4*m*z + 9*a^5*b^2*c^2*k^4*l*z + 216*a^
4*c^5*e^2*g*l^2*z - 108*a^4*c^5*f^2*g*k^2*z - 108*a^4*c^5*d^2*g*m^2*z - 9*a
^4*b^4*c*j^2*l^3*z - 108*a^4*c^5*e*h^2*j^2*z - 108*a^4*c^5*e*f^2*l^2*z + 10
8*a^3*c^6*e^2*f^2*l*z - 36*a^5*b*c^3*j^2*k^3*z + 36*a^5*b*c^3*h^3*m^2*z + 1
08*a^3*c^6*e^2*g^2*j*z + 108*a^3*c^6*d^2*h^2*j*z - 216*a^5*b*c^3*f^2*m^3*z

+ 144*a^4*b*c^4*f^3*m^2*z + 108*a^3*c^6*d^2*g*j^2*z - 72*a^3*b^5*c*f^2*m^3*
z - 45*a^5*b^2*c^2*g*l^4*z - 9*a^4*b^3*c^2*h*k^4*z - 9*a^3*b^2*c^4*g^4*l*z
+ 9*a^2*b^3*c^4*f^4*m*z + 216*a^3*c^6*d^2*e*k^2*z - 9*a^2*b^6*c*f^2*l^3*z +
9*a*b^6*c^2*e^3*m^2*z + 108*a^3*c^6*e*f^2*h^2*z + 108*a^3*b*c^5*d^3*m^2*z
+ 108*a^2*c^7*d^2*e^2*j*z + 72*a^4*b*c^4*f^2*k^3*z + 72*a*b^5*c^3*d^3*m^2*z
- 72*a^3*b*c^5*f^3*j^2*z + 54*a^4*b^3*c^2*d*l^4*z - 45*a^4*b^2*c^3*e*k^4*z
+ 18*a^3*b^3*c^3*f*j^4*z + 9*a^3*b^4*c^2*e*k^4*z - 9*a^2*b^2*c^5*f^4*j*z -
108*a^2*c^7*d^2*f^2*g*z + 9*a^3*b^2*c^4*g*h^4*z + 9*a*b^4*c^4*e^3*j^2*z -
72*a^2*b*c^6*d^3*j^2*z + 54*a*b^3*c^5*d^3*j^2*z - 36*a^3*b*c^5*f^2*h^3*z -
9*a^2*b^3*c^4*d*h^4*z + 9*a^2*b^2*c^5*e*g^4*z + 9*a*b^2*c^6*e^3*f^2*z + 36*
a^7*c^2*l^3*m^2*z + 72*a^6*c^3*j^3*m^2*z - 36*a^6*c^3*j^2*l^3*z + 9*a^4*b^5
*j^2*m^3*z + 36*a^5*c^4*g^3*m^2*z + 36*a^5*c^4*f^2*l^3*z - 36*a^4*c^5*e^3*m
^2*z - 9*b^7*c^2*d^3*m^2*z + 9*a^2*b^7*f^2*m^3*z - 36*a^4*c^5*g^3*j^2*z + 7
2*a^4*c^5*f^2*j^3*z + 36*a^3*c^6*e^3*j^2*z - 9*b^5*c^4*d^3*j^2*z + 36*a^3*c
^6*f^2*g^3*z - 9*a^4*b^2*c^3*j^5*z - 36*a^2*c^7*e^3*f^2*z - 9*b^3*c^6*d^3*f
^2*z + 36*a^7*c^2*j*m^4*z - 36*a^6*c^3*k^4*l*z - 18*a^5*b^4*j*m^4*z + 36*a^
6*c^3*g*l^4*z + 36*a^4*c^5*g^4*l*z + 18*a^4*b^5*f*m^4*z - 9*b^4*c^5*d^4*l*z
+ 36*a^5*c^4*e*k^4*z + 36*a^3*c^6*f^4*j*z - 36*a^2*c^7*d^4*l*z - 36*a^4*c^
5*g*h^4*z + 9*b^3*c^6*d^4*h*z - 36*a^3*c^6*e*g^4*z + 36*a^2*c^7*e^4*g*z - 9
*b^2*c^7*d^4*e*z - 36*a^7*b*c*m^5*z + 36*a*c^8*d^4*e*z + 9*a^6*b^3*m^5*z +
36*a^5*c^4*j^5*z + 9*a^4*b^3*c*g*h*j*k*l*m - 9*a^3*b^4*c*e*g*j*k*l*m - 9*a^
3*b^4*c*d*h*j*k*l*m - 9*a^3*b^4*c*f*g*h*k*l*m + 36*a^4*b*c^3*d*e*j*k*l*m +
9*a^2*b^5*c*d*e*j*k*l*m + 36*a^4*b*c^3*e*f*h*j*l*m + 36*a^4*b*c^3*e*f*g*k*l
*m + 36*a^4*b*c^3*d*f*h*k*l*m + 9*a^2*b^5*c*e*f*g*k*l*m + 9*a^2*b^5*c*d*f*h
*k*l*m + 36*a^3*b*c^4*d*e*f*j*k*l + 9*a*b^5*c^2*d*e*f*j*k*l + 36*a^3*b*c^4*
d*e*g*h*k*l + 36*a^3*b*c^4*d*e*f*h*k*m + 36*a^3*b*c^4*d*e*f*g*l*m + 9*a*b^5
*c^2*d*e*f*h*k*m + 9*a*b^5*c^2*d*e*f*g*l*m - 9*a*b^4*c^3*d*e*f*h*j*k - 9*a*
b^4*c^3*d*e*f*g*j*l - 9*a*b^4*c^3*d*e*f*g*h*m + 9*a*b^3*c^4*d*e*f*g*h*j - 9
*a*b^6*c*d*e*f*k*l*m + 18*a^4*b^2*c^2*e*g*j*k*l*m + 18*a^4*b^2*c^2*d*h*j*k*
l*m + 18*a^4*b^2*c^2*f*g*h*k*l*m - 36*a^3*b^3*c^2*d*e*j*k*l*m - 36*a^3*b^3*
c^2*e*f*g*k*l*m - 36*a^3*b^3*c^2*d*f*h*k*l*m + 9*a^3*b^3*c^2*f*g*h*j*k*l +
9*a^3*b^3*c^2*e*g*h*j*k*m + 9*a^3*b^3*c^2*d*g*h*j*l*m - 108*a^3*b^2*c^3*d*e
*f*k*l*m + 54*a^2*b^4*c^2*d*e*f*k*l*m - 36*a^3*b^2*c^3*d*f*g*j*k*m + 18*a^3
*b^2*c^3*e*f*g*j*k*l + 18*a^3*b^2*c^3*d*f*h*j*k*l + 18*a^3*b^2*c^3*d*e*h*j*
k*m + 18*a^3*b^2*c^3*d*e*g*j*l*m - 9*a^2*b^4*c^2*e*f*g*j*k*l - 9*a^2*b^4*c^
2*d*f*h*j*k*l - 9*a^2*b^4*c^2*d*e*h*j*k*m - 9*a^2*b^4*c^2*d*e*g*j*l*m + 18*
a^3*b^2*c^3*e*f*g*h*k*m + 18*a^3*b^2*c^3*d*f*g*h*l*m - 9*a^2*b^4*c^2*e*f*g*
h*k*m - 9*a^2*b^4*c^2*d*f*g*h*l*m - 36*a^2*b^3*c^3*d*e*f*j*k*l - 36*a^2*b^3
*c^3*d*e*f*h*k*m - 36*a^2*b^3*c^3*d*e*f*g*l*m + 9*a^2*b^3*c^3*e*f*g*h*j*k +
9*a^2*b^3*c^3*d*f*g*h*j*l + 9*a^2*b^3*c^3*d*e*g*h*j*m + 18*a^2*b^2*c^4*d*e
*f*h*j*k + 18*a^2*b^2*c^4*d*e*f*g*j*l + 18*a^2*b^2*c^4*d*e*f*g*h*m - 9*a^5*
b^2*c*h*j*k^2*l*m - 9*a^5*b^2*c*g*j*k*l^2*m + 27*a^5*b^2*c*f*j*k*l*m^2 - 9*
a^4*b^3*c*f*j^2*k*l*m + 9*a^3*b^4*c*f^2*j*k*l*m - 18*a^5*b*c^2*e*j*k^2*l*m
- 9*a^5*b^2*c*g*h*k*l*m^2 + 9*a^4*b^3*c*e*j*k^2*l*m - 18*a^5*b*c^2*f*h*k^2*
l*m - 18*a^5*b*c^2*d*j*k*l^2*m + 9*a^4*b^3*c*f*h*k^2*l*m + 9*a^4*b^3*c*d*j*

k*l^2*m + 36*a^5*b*c^2*e*h*k*l^2*m - 36*a^4*b*c^3*e^2*h*k*l*m + 18*a^5*b*c^
2*f*h*j*l^2*m - 18*a^5*b*c^2*f*g*k*l^2*m - 18*a^4*b^3*c*e*h*k*l^2*m + 9*a^4
*b^3*c*f*g*k*l^2*m + 9*a^3*b^4*c*e*h^2*k*l*m - 9*a^2*b^5*c*e^2*h*k*l*m - 54
*a^5*b*c^2*e*h*j*l*m^2 - 18*a^5*b*c^2*e*g*k*l*m^2 - 18*a^5*b*c^2*d*h*k*l*m^
2 + 18*a^4*b^3*c*e*h*j*l*m^2 - 9*a^4*b^3*c*f*h*j*k*m^2 - 9*a^4*b^3*c*f*g*j*
l*m^2 + 9*a^4*b^3*c*e*g*k*l*m^2 + 9*a^4*b^3*c*d*h*k*l*m^2 + 18*a^4*b*c^3*f*
g^2*j*k*m - 18*a^4*b*c^3*e*g^2*j*l*m + 18*a^3*b^4*c*d*g*k^2*l*m - 9*a^3*b^4
*c*e*f*k^2*l*m - 9*a^2*b^5*c*d*g^2*k*l*m - 18*a^4*b*c^3*f*g^2*h*l*m - 18*a^
4*b*c^3*d*h^2*j*k*m - 9*a^3*b^4*c*d*f*k*l^2*m - 54*a^4*b*c^3*d*g*j^2*k*m -
18*a^4*b*c^3*f*g*h^2*k*m - 18*a^4*b*c^3*e*g*j^2*k*l - 18*a^4*b*c^3*d*h*j^2*
k*l - 18*a^3*b^4*c*d*g*j*k*m^2 + 9*a^3*b^4*c*e*f*j*k*m^2 + 9*a^3*b^4*c*d*f*
j*l*m^2 - 9*a^3*b^4*c*d*e*k*l*m^2 - 54*a^3*b*c^4*d^2*f*j*k*m + 36*a^4*b*c^3
*d*g*j*k^2*l - 36*a^3*b*c^4*d^2*g*j*k*l - 18*a^4*b*c^3*e*f*j*k^2*l + 18*a^4
*b*c^3*d*f*j*k^2*m - 18*a^3*b*c^4*d^2*e*j*l*m + 9*a^3*b^4*c*f*g*h*j*m^2 - 9
*a*b^5*c^2*d^2*g*j*k*l + 36*a^4*b*c^3*d*g*h*k^2*m - 36*a^3*b*c^4*d^2*g*h*k*
m + 18*a^4*b*c^3*e*g*h*k^2*l - 18*a^4*b*c^3*e*f*h*k^2*m - 18*a^4*b*c^3*d*f*
j*k*l^2 - 18*a^3*b*c^4*d^2*f*h*l*m - 18*a^3*b*c^4*d*e^2*j*k*m - 9*a*b^5*c^2
*d^2*g*h*k*m - 54*a^4*b*c^3*d*g*h*k*l^2 - 54*a^3*b*c^4*e^2*f*h*j*m - 18*a^4
*b*c^3*d*f*g*l^2*m - 18*a^3*b*c^4*e^2*f*g*k*m - 54*a^4*b*c^3*d*f*g*k*m^2 -
36*a^4*b*c^3*e*f*g*j*m^2 - 36*a^4*b*c^3*d*f*h*j*m^2 + 36*a^3*b*c^4*e*f^2*g*
j*m + 36*a^3*b*c^4*d*f^2*h*j*m - 18*a^4*b*c^3*d*e*h*k*m^2 - 18*a^4*b*c^3*d*
e*g*l*m^2 + 18*a^3*b*c^4*e*f^2*h*j*l - 18*a^3*b*c^4*e*f^2*g*k*l - 18*a^3*b*
c^4*d*f^2*h*k*l + 18*a^3*b*c^4*d*f^2*g*k*m - 9*a^2*b^5*c*e*f*g*j*m^2 - 9*a^
2*b^5*c*d*f*h*j*m^2 - 54*a^3*b*c^4*d*f*g^2*j*m - 18*a^3*b*c^4*e*f*g^2*j*l -
18*a*b^4*c^3*d^2*f*g*j*m + 9*a*b^4*c^3*d^2*g*h*j*k + 9*a*b^4*c^3*d^2*f*g*k
*l + 9*a*b^4*c^3*d^2*e*g*k*m - 9*a*b^4*c^3*d^2*e*f*l*m - 18*a^3*b*c^4*e*f*g
^2*h*m - 18*a^3*b*c^4*d*f*h^2*j*k - 9*a*b^4*c^3*d*e^2*f*k*m + 18*a^3*b*c^4*
d*f*g*j^2*k - 18*a^3*b*c^4*d*f*g*h^2*m - 18*a^3*b*c^4*d*e*h*j^2*k - 18*a^3*
b*c^4*d*e*g*j^2*l + 18*a*b^4*c^3*d*e*f^2*j*m - 9*a*b^5*c^2*d*e*f*j^2*m - 9*
a*b^4*c^3*d*e*f^2*k*l - 18*a^2*b*c^5*d^2*e*f*j*l - 9*a*b^3*c^4*d^2*e*g*j*k
+ 9*a*b^3*c^4*d^2*e*f*j*l - 54*a^2*b*c^5*d^2*e*g*h*l - 18*a^2*b*c^5*d^2*e*f
*h*m - 18*a^2*b*c^5*d*e^2*f*j*k + 18*a*b^3*c^4*d^2*e*g*h*l - 9*a*b^3*c^4*d^
2*f*g*h*k + 9*a*b^3*c^4*d^2*e*f*h*m + 9*a*b^3*c^4*d*e^2*f*j*k - 36*a^3*b*c^
4*d*e*f*h*l^2 + 36*a^2*b*c^5*d*e^2*f*h*l + 18*a^2*b*c^5*d*e^2*g*h*k - 18*a^
2*b*c^5*d*e^2*f*g*m - 18*a*b^3*c^4*d*e^2*f*h*l - 9*a*b^5*c^2*d*e*f*h*l^2 +
9*a*b^4*c^3*d*e*f*h^2*l + 9*a*b^3*c^4*d*e^2*f*g*m - 18*a^2*b*c^5*d*e*f^2*h*
k - 18*a^2*b*c^5*d*e*f^2*g*l + 9*a*b^3*c^4*d*e*f^2*h*k + 9*a*b^3*c^4*d*e*f^
2*g*l + 27*a*b^2*c^5*d^2*e*f*g*k + 9*a*b^4*c^3*d*e*f*g*k^2 - 9*a*b^3*c^4*d*
e*f*g^2*k - 9*a*b^2*c^5*d^2*e*f*h*j - 9*a*b^2*c^5*d*e^2*f*g*j - 9*a*b^2*c^5
*d*e*f^2*g*h + 72*a^4*c^4*d*f*g*j*k*m + 72*a^4*c^4*d*e*f*k*l*m + 9*a*b^6*c*
d^2*g*k*l*m + 9*a*b^6*c*d*e*f*j*m^2 - 27*a^4*b^2*c^2*f^2*j*k*l*m - 9*a^4*b^
2*c^2*g^2*h*j*l*m + 36*a^3*b^3*c^2*e^2*h*k*l*m - 18*a^4*b^2*c^2*e*h^2*k*l*m
- 9*a^4*b^2*c^2*g*h^2*j*k*m + 18*a^4*b^2*c^2*f*h*j^2*k*m + 18*a^4*b^2*c^2*
f*g*j^2*l*m - 18*a^4*b^2*c^2*e*h*j^2*l*m - 9*a^4*b^2*c^2*g*h*j^2*k*l - 9*a^
3*b^3*c^2*f^2*h*j*k*m - 9*a^3*b^3*c^2*f^2*g*j*l*m - 63*a^4*b^2*c^2*d*g*k^2*

l*m + 63*a^3*b^2*c^3*d^2*g*k*l*m - 45*a^2*b^4*c^2*d^2*g*k*l*m + 36*a^4*b^2*
c^2*e*f*k^2*l*m + 27*a^3*b^3*c^2*d*g^2*k*l*m - 9*a^4*b^2*c^2*f*h*j*k^2*l -
9*a^4*b^2*c^2*e*h*j*k^2*m + 9*a^3*b^3*c^2*e*g^2*j*l*m - 9*a^3*b^2*c^3*d^2*h
*j*l*m + 36*a^4*b^2*c^2*d*f*k*l^2*m + 27*a^4*b^2*c^2*e*h*j*k*l^2 - 27*a^3*b
^2*c^3*e^2*h*j*k*l - 18*a^3*b^2*c^3*e^2*f*j*l*m - 9*a^4*b^2*c^2*f*g*j*k*l^2
 - 9*a^4*b^2*c^2*d*g*j*l^2*m + 9*a^3*b^3*c^2*f*g^2*h*l*m - 9*a^3*b^3*c^2*e*
h^2*j*k*l + 9*a^3*b^3*c^2*d*h^2*j*k*m - 9*a^3*b^2*c^3*e^2*g*j*k*m + 9*a^2*b
^4*c^2*e^2*h*j*k*l + 72*a^4*b^2*c^2*d*g*j*k*m^2 + 36*a^4*b^2*c^2*d*e*k*l*m^
2 + 27*a^4*b^2*c^2*e*g*h*l^2*m - 27*a^4*b^2*c^2*e*f*j*k*m^2 - 27*a^4*b^2*c^
2*d*f*j*l*m^2 - 27*a^3*b^2*c^3*e^2*g*h*l*m + 27*a^3*b^2*c^3*e*f^2*j*k*m + 2
7*a^3*b^2*c^3*d*f^2*j*l*m + 18*a^3*b^3*c^2*d*g*j^2*k*m + 9*a^3*b^3*c^2*f*g*
h^2*k*m + 9*a^3*b^3*c^2*e*g*j^2*k*l - 9*a^3*b^3*c^2*e*g*h^2*l*m - 9*a^3*b^3
*c^2*e*f*j^2*k*m + 9*a^3*b^3*c^2*d*h*j^2*k*l - 9*a^3*b^3*c^2*d*f*j^2*l*m +
9*a^2*b^4*c^2*e^2*g*h*l*m + 36*a^2*b^3*c^3*d^2*g*j*k*l - 27*a^4*b^2*c^2*f*g
*h*j*m^2 + 27*a^3*b^2*c^3*f^2*g*h*j*m - 18*a^4*b^2*c^2*e*f*h*l*m^2 - 18*a^3
*b^3*c^2*d*g*j*k^2*l - 18*a^3*b^2*c^3*d*g^2*j*k*l + 18*a^2*b^3*c^3*d^2*f*j*
k*m - 9*a^4*b^2*c^2*e*g*h*k*m^2 - 9*a^4*b^2*c^2*d*g*h*l*m^2 - 9*a^3*b^3*c^2
*f*g*h*j^2*m + 9*a^3*b^3*c^2*e*f*j*k^2*l - 9*a^3*b^2*c^3*f^2*g*h*k*l + 9*a^
2*b^4*c^2*d*g^2*j*k*l + 9*a^2*b^3*c^3*d^2*e*j*l*m + 36*a^3*b^2*c^3*e*f*g^2*
l*m + 36*a^2*b^3*c^3*d^2*g*h*k*m - 18*a^3*b^3*c^2*d*g*h*k^2*m - 18*a^3*b^2*
c^3*d*g^2*h*k*m + 9*a^3*b^3*c^2*e*f*h*k^2*m + 9*a^3*b^3*c^2*d*f*j*k*l^2 - 9
*a^3*b^2*c^3*f*g^2*h*j*l - 9*a^3*b^2*c^3*e*g^2*h*j*m - 9*a^2*b^4*c^2*e*f*g^
2*l*m + 9*a^2*b^4*c^2*d*g^2*h*k*m + 9*a^2*b^3*c^3*d^2*f*h*l*m + 9*a^2*b^3*c
^3*d*e^2*j*k*m + 36*a^3*b^2*c^3*d*f*h^2*k*m + 36*a^3*b^2*c^3*d*e*j^2*k*l +
18*a^3*b^3*c^2*d*g*h*k*l^2 + 18*a^3*b^2*c^3*e*g*h^2*j*l + 18*a^3*b^2*c^3*e*
f*h^2*k*l - 18*a^3*b^2*c^3*e*f*h^2*j*m - 18*a^3*b^2*c^3*d*g*h^2*k*l + 18*a^
3*b^2*c^3*d*e*h^2*l*m + 18*a^2*b^3*c^3*e^2*f*h*j*m - 9*a^3*b^3*c^2*e*g*h*j*
l^2 - 9*a^3*b^3*c^2*e*f*h*k*l^2 + 9*a^3*b^3*c^2*d*f*g*l^2*m - 9*a^3*b^3*c^2
*d*e*h*l^2*m - 9*a^3*b^2*c^3*f*g*h^2*j*k - 9*a^3*b^2*c^3*d*g*h^2*j*m - 9*a^
2*b^4*c^2*d*f*h^2*k*m - 9*a^2*b^4*c^2*d*e*j^2*k*l - 9*a^2*b^3*c^3*e^2*g*h*j
*l - 9*a^2*b^3*c^3*e^2*f*h*k*l + 9*a^2*b^3*c^3*e^2*f*g*k*m - 9*a^2*b^3*c^3*
d*e^2*h*l*m + 36*a^3*b^3*c^2*e*f*g*j*m^2 + 36*a^3*b^3*c^2*d*f*h*j*m^2 + 18*
a^3*b^3*c^2*d*f*g*k*m^2 - 18*a^3*b^2*c^3*e*f*g*j^2*m - 18*a^3*b^2*c^3*d*f*h
*j^2*m - 18*a^2*b^3*c^3*e*f^2*g*j*m - 18*a^2*b^3*c^3*d*f^2*h*j*m + 9*a^3*b^
3*c^2*d*e*h*k*m^2 + 9*a^3*b^3*c^2*d*e*g*l*m^2 - 9*a^3*b^2*c^3*e*g*h*j^2*k -
 9*a^3*b^2*c^3*d*g*h*j^2*l + 9*a^2*b^4*c^2*e*f*g*j^2*m + 9*a^2*b^4*c^2*d*f*
h*j^2*m + 9*a^2*b^3*c^3*e*f^2*g*k*l + 9*a^2*b^3*c^3*d*f^2*h*k*l + 72*a^2*b^
2*c^4*d^2*f*g*j*m + 36*a^2*b^2*c^4*d^2*e*f*l*m + 27*a^3*b^2*c^3*d*g*h*j*k^2
 + 27*a^3*b^2*c^3*d*f*g*k^2*l + 27*a^3*b^2*c^3*d*e*g*k^2*m - 27*a^2*b^2*c^4
*d^2*g*h*j*k - 27*a^2*b^2*c^4*d^2*f*g*k*l - 27*a^2*b^2*c^4*d^2*e*g*k*m + 18
*a^2*b^3*c^3*d*f*g^2*j*m - 18*a^2*b^2*c^4*d^2*e*h*k*l - 9*a^3*b^2*c^3*e*f*h
*j*k^2 + 9*a^2*b^3*c^3*e*f*g^2*j*l - 9*a^2*b^3*c^3*d*g^2*h*j*k - 9*a^2*b^3*
c^3*d*f*g^2*k*l - 9*a^2*b^3*c^3*d*e*g^2*k*m - 9*a^2*b^2*c^4*d^2*f*h*j*l - 9
*a^2*b^2*c^4*d^2*e*h*j*m + 36*a^2*b^2*c^4*d*e^2*f*k*m - 27*a^3*b^2*c^3*d*e*
h*j*l^2 + 27*a^2*b^2*c^4*d*e^2*h*j*l - 18*a^3*b^2*c^3*d*e*g*k*l^2 - 9*a^3*b

^2*c^3*d*f*g*j*l^2 + 9*a^2*b^4*c^2*d*e*h*j*l^2 + 9*a^2*b^3*c^3*e*f*g^2*h*m
+ 9*a^2*b^3*c^3*d*f*h^2*j*k - 9*a^2*b^3*c^3*d*e*h^2*j*l - 9*a^2*b^2*c^4*e^2
*f*g*j*k - 9*a^2*b^2*c^4*d*e^2*g*j*m + 63*a^3*b^2*c^3*d*e*f*j*m^2 - 63*a^2*
b^2*c^4*d*e*f^2*j*m - 45*a^2*b^4*c^2*d*e*f*j*m^2 + 36*a^2*b^2*c^4*d*e*f^2*k
*l - 27*a^3*b^2*c^3*e*f*g*h*l^2 + 27*a^2*b^3*c^3*d*e*f*j^2*m + 27*a^2*b^2*c
^4*e^2*f*g*h*l + 9*a^2*b^4*c^2*e*f*g*h*l^2 - 9*a^2*b^3*c^3*e*f*g*h^2*l + 9*
a^2*b^3*c^3*d*f*g*h^2*m + 9*a^2*b^3*c^3*d*e*h*j^2*k + 9*a^2*b^3*c^3*d*e*g*j
^2*l + 18*a^2*b^2*c^4*d*e*g^2*j*k - 9*a^3*b^2*c^3*d*e*g*h*m^2 - 9*a^2*b^3*c
^3*d*e*g*j*k^2 - 9*a^2*b^2*c^4*e*f^2*g*h*k - 9*a^2*b^2*c^4*d*f^2*g*h*l + 18
*a^2*b^2*c^4*d*f*g^2*h*k - 18*a^2*b^2*c^4*d*e*g^2*h*l - 9*a^2*b^3*c^3*d*f*g
*h*k^2 - 9*a^2*b^2*c^4*e*f*g^2*h*j + 36*a^2*b^3*c^3*d*e*f*h*l^2 - 18*a^2*b^
2*c^4*d*e*f*h^2*l - 9*a^2*b^2*c^4*d*f*g*h^2*j - 9*a^2*b^2*c^4*d*e*g*h*j^2 -
27*a^2*b^2*c^4*d*e*f*g*k^2 + 18*a^2*b^2*c^4*d^2*f*h*k^2 - 9*a^2*b^3*c^3*e*
f*g^2*k^2 - 9*a^2*b^2*c^4*e^2*f*h*j^2 - 9*a^2*b^2*c^4*d*f^2*h^2*k + 45*a^2*
b^3*c^3*d*e*f^2*m^2 + 36*a^2*b^2*c^4*d^2*e*g*l^2 + 9*a^2*b^3*c^3*d*e*g^2*l^
2 + 9*a^2*b^2*c^4*e*f^2*g*j^2 + 9*a^2*b^2*c^4*d*f^2*h*j^2 - 9*a^2*b^2*c^4*d
*e^2*h*k^2 - 36*a^2*b^2*c^4*d*e^2*f*l^2 - 9*a^2*b^2*c^4*d*f*g^2*j^2 - 12*a^
6*b*c*h*k*l^3*m + 3*a*b^6*c*e^3*k*l*m + 3*a*b^6*c*d*e*f*l^3 - 12*a*b*c^6*d*
e^3*f*h + 9*a^5*b^2*c*h^2*k*l^2*m + 18*a^5*b*c^2*g^2*k^2*l*m - 9*a^5*b^2*c*
h^2*j*l*m^2 + 9*a^5*b*c^2*h^2*j^2*l*m - 9*a^4*b^3*c*g^2*k^2*l*m - 3*a^4*b^2
*c^2*g^3*k*l*m + 18*a^5*b*c^2*f^2*k*l*m^2 + 15*a^3*b^3*c^2*f^3*k*l*m + 9*a^
5*b^2*c*h*j^2*k*m^2 + 9*a^5*b^2*c*g*j^2*l*m^2 - 9*a^5*b^2*c*f*k^2*l^2*m + 9
*a^5*b*c^2*h^2*j*k^2*m + 9*a^5*b*c^2*g^2*j*l^2*m - 9*a^4*b^3*c*f^2*k*l*m^2
+ 36*a^3*b^2*c^3*e^3*k*l*m - 27*a^5*b*c^2*g^2*j*k*m^2 - 18*a^5*b*c^2*h^2*j*
k*l^2 - 18*a^2*b^4*c^2*e^3*k*l*m - 9*a^5*b^2*c*g*j*k^2*m^2 - 9*a^5*b^2*c*e*
k^2*l*m^2 + 9*a^5*b*c^2*h*j^2*k^2*l + 9*a^5*b*c^2*g*j^2*k^2*m + 9*a^4*b^3*c
*g^2*j*k*m^2 + 9*a^3*b^4*c*e^2*k*l^2*m + 3*a^4*b^2*c^2*h^3*j*k*l - 54*a^4*b
*c^3*d^2*k^2*l*m - 51*a^2*b^3*c^3*d^3*k*l*m - 27*a^4*b*c^3*e^2*j^2*l*m - 18
*a^5*b*c^2*g*h^2*l^2*m - 9*a^5*b^2*c*e*j*l^2*m^2 - 9*a^5*b^2*c*d*k*l^2*m^2
+ 9*a^5*b*c^2*g^2*h*l*m^2 + 9*a^5*b*c^2*g*j^2*k*l^2 + 9*a^5*b*c^2*e*j^2*l^2
*m - 9*a^3*b^4*c*e^2*j*l*m^2 - 9*a^2*b^5*c*d^2*k^2*l*m + 3*a^4*b^2*c^2*g*h^
3*l*m - 3*a^3*b^3*c^2*g^3*j*k*l + 18*a^5*b*c^2*e*j^2*k*m^2 + 18*a^5*b*c^2*d
*j^2*l*m^2 + 18*a^4*b*c^3*f^2*j^2*k*l + 9*a^5*b*c^2*g*h^2*k*m^2 + 9*a^5*b*c
^2*f*h^2*l*m^2 + 9*a^5*b*c^2*f*j*k^2*l^2 - 9*a^4*b^3*c*e*j^2*k*m^2 - 9*a^4*
b^3*c*d*j^2*l*m^2 + 9*a^4*b^2*c^2*f*j^3*k*l + 9*a^4*b^2*c^2*e*j^3*k*m + 9*a
^4*b^2*c^2*d*j^3*l*m + 9*a^4*b*c^3*f^2*h^2*l*m + 9*a^4*b*c^3*e^2*j*k^2*m +
9*a^4*b*c^3*d^2*j*l^2*m - 3*a^3*b^3*c^2*g^3*h*k*m - 3*a^3*b^2*c^3*f^3*j*k*l
+ 3*a^2*b^4*c^2*f^3*j*k*l + 45*a^4*b*c^3*d^2*j*k*m^2 - 27*a^5*b*c^2*d*j*k^
2*m^2 + 18*a^5*b*c^2*g*h*j^2*m^2 + 18*a^4*b*c^3*e^2*j*k*l^2 + 15*a^2*b^3*c^
3*e^3*j*k*l - 12*a^3*b^2*c^3*f^3*h*k*m - 12*a^3*b^2*c^3*f^3*g*l*m + 9*a^5*b
*c^2*g*h*k^2*l^2 - 9*a^4*b^3*c*g*h*j^2*m^2 + 9*a^4*b^3*c*d*j*k^2*m^2 + 9*a^
4*b^2*c^2*g*h*j^3*m + 9*a^4*b*c^3*g^2*h^2*k*l + 9*a^4*b*c^3*g^2*h^2*j*m + 9
*a^2*b^5*c*d^2*j*k*m^2 + 3*a^2*b^4*c^2*f^3*h*k*m + 3*a^2*b^4*c^2*f^3*g*l*m
+ 36*a^2*b^2*c^4*d^3*j*k*l + 18*a^4*b*c^3*e^2*g*l^2*m + 15*a^2*b^3*c^3*e^3*
g*l*m + 12*a^4*b^2*c^2*d*j*k^3*l + 9*a^5*b*c^2*f*g*k^2*m^2 + 9*a^5*b*c^2*e*

h*k^2*m^2 + 9*a^4*b*c^3*g^2*h*j^2*l + 9*a^4*b*c^3*f^2*h*k^2*l + 9*a^4*b*c^3
*f^2*g*k^2*m + 9*a^4*b*c^3*d^2*h*l*m^2 - 9*a^3*b^3*c^2*e*h^3*k*m + 6*a^2*b^
3*c^3*e^3*h*k*m + 45*a^4*b*c^3*e^2*h*j*m^2 + 36*a^2*b^2*c^4*d^3*h*k*m - 33*
a^3*b^2*c^3*d*g^3*l*m - 27*a^4*b*c^3*f^2*h*j*l^2 - 27*a^4*b*c^3*e^2*f*l*m^2
 - 27*a^4*b*c^3*e*h^2*j^2*m - 18*a^4*b*c^3*g^2*h*j*k^2 - 18*a^4*b*c^3*f*g^2
*k^2*l - 18*a^4*b*c^3*e*g^2*k^2*m - 18*a^3*b*c^4*d^2*g^2*l*m + 12*a^4*b^2*c
^2*d*h*k^3*m + 9*a^5*b*c^2*e*f*l^2*m^2 + 9*a^5*b*c^2*d*g*l^2*m^2 + 9*a^4*b*
c^3*f^2*g*k*l^2 + 9*a^4*b*c^3*e^2*g*k*m^2 + 9*a^4*b*c^3*g*h^2*j^2*k + 9*a^4
*b*c^3*f*h^2*j^2*l + 9*a^4*b*c^3*e*f^2*l^2*m - 9*a^3*b^4*c*e*h^2*j*m^2 + 9*
a^3*b*c^4*e^2*f^2*l*m + 9*a^2*b^5*c*e^2*h*j*m^2 + 9*a^2*b^4*c^2*d*g^3*l*m -
 9*a^2*b^2*c^4*d^3*g*l*m - 9*a*b^5*c^2*d^2*g^2*l*m - 6*a^4*b^2*c^2*e*h*k^3*
l - 6*a^3*b^2*c^3*f*g^3*j*m + 3*a^4*b^2*c^2*g*h*j*k^3 + 3*a^4*b^2*c^2*f*g*k
^3*l + 3*a^4*b^2*c^2*e*g*k^3*m + 3*a^3*b^2*c^3*g^3*h*j*k + 3*a^3*b^2*c^3*f*
g^3*k*l + 3*a^3*b^2*c^3*e*g^3*k*m - 27*a^3*b*c^4*d^2*h^2*k*l + 18*a^4*b*c^3
*e*f^2*k*m^2 + 18*a^4*b*c^3*d*f^2*l*m^2 + 9*a^4*b*c^3*f*h^2*j*k^2 + 9*a^4*b
*c^3*f*g^2*j*l^2 + 9*a^4*b*c^3*e*g^2*k*l^2 + 9*a^4*b*c^3*d*h^2*k^2*l + 9*a^
3*b^4*c*e*g*j^2*m^2 + 9*a^3*b^4*c*d*h*j^2*m^2 - 9*a^3*b^3*c^2*e*g*j^3*m - 9
*a^3*b^3*c^2*d*h*j^3*m + 9*a^3*b*c^4*e^2*g^2*k*l + 9*a^3*b*c^4*e^2*g^2*j*m
+ 9*a^3*b*c^4*d^2*h^2*j*m - 3*a^2*b^3*c^3*f^3*h*j*k - 3*a^2*b^3*c^3*f^3*g*j
*l - 3*a^2*b^3*c^3*e*f^3*k*m - 3*a^2*b^3*c^3*d*f^3*l*m + 45*a^4*b*c^3*d*g^2
*j*m^2 + 45*a^3*b*c^4*d^2*g*j^2*m + 24*a^4*b^2*c^2*d*g*k*l^3 + 24*a^2*b^2*c
^4*e^3*f*j*m + 18*a^4*b*c^3*f^2*g*h*m^2 + 18*a^4*b*c^3*d*h^2*j*l^2 + 18*a^3
*b*c^4*e^2*h^2*j*k - 12*a^4*b^2*c^2*e*g*j*l^3 - 12*a^4*b^2*c^2*e*f*k*l^3 -
12*a^4*b^2*c^2*d*e*l^3*m - 12*a^2*b^2*c^4*e^3*g*j*l - 12*a^2*b^2*c^4*e^3*f*
k*l - 12*a^2*b^2*c^4*d*e^3*l*m + 9*a^4*b*c^3*f*g*j^2*k^2 + 9*a^4*b*c^3*e*h*
j^2*k^2 + 9*a^3*b^2*c^3*e*h^3*j*k + 9*a^3*b^2*c^3*d*h^3*j*l + 9*a^3*b*c^4*f
^2*g^2*j*k + 9*a^3*b*c^4*d^2*h*j^2*l + 9*a^2*b^5*c*d*g^2*j*m^2 + 9*a*b^5*c^
2*d^2*g*j^2*m - 3*a^4*b^2*c^2*d*h*j*l^3 - 3*a^2*b^3*c^3*f^3*g*h*m - 3*a^2*b
^2*c^4*e^3*h*j*k + 18*a^4*b*c^3*f*g*h^2*l^2 + 18*a^3*b*c^4*e^2*g*h^2*m + 18
*a^3*b*c^4*d^2*h*j*k^2 + 18*a^3*b*c^4*d^2*f*k^2*l + 18*a^3*b*c^4*d^2*e*k^2*
m + 9*a^4*b*c^3*e*g^2*h*m^2 + 9*a^4*b*c^3*e*f*j^2*l^2 + 9*a^4*b*c^3*d*g*j^2
*l^2 + 9*a^3*b^2*c^3*f*g*h^3*l + 9*a^3*b^2*c^3*e*g*h^3*m + 9*a^3*b*c^4*f^2*
g^2*h*l + 9*a^3*b*c^4*e^2*g*j^2*k + 9*a^3*b*c^4*e^2*f*j^2*l - 9*a^2*b^3*c^3
*d*g^3*j*l + 9*a*b^4*c^3*d^2*g^2*j*l - 3*a^4*b^2*c^2*f*g*h*l^3 - 3*a^3*b^3*
c^2*e*g*j*k^3 - 3*a^3*b^3*c^2*d*h*j*k^3 - 3*a^3*b^3*c^2*d*f*k^3*l - 3*a^3*b
^3*c^2*d*e*k^3*m - 3*a^2*b^2*c^4*e^3*g*h*m - 33*a^3*b^2*c^3*d*e*j^3*m - 27*
a^4*b*c^3*e*f*h^2*m^2 - 27*a^3*b*c^4*d^2*e*k*l^2 - 18*a^4*b*c^3*d*e*j^2*m^2
 - 18*a^3*b*c^4*e*f^2*j^2*k - 18*a^3*b*c^4*d*f^2*j^2*l - 9*a^4*b^2*c^2*d*e*
j*m^3 + 9*a^4*b*c^3*d*g*h^2*m^2 + 9*a^4*b*c^3*d*e*k^2*l^2 + 9*a^3*b*c^4*f^2
*g*h^2*k + 9*a^3*b*c^4*e^2*f*j*k^2 + 9*a^3*b*c^4*d^2*f*j*l^2 + 9*a^3*b*c^4*
e*f^2*h^2*m + 9*a^3*b*c^4*d*e^2*k^2*l - 9*a^2*b^5*c*d*e*j^2*m^2 + 9*a^2*b^4
*c^2*d*e*j^3*m - 9*a^2*b^3*c^3*d*g^3*h*m + 9*a^2*b*c^5*d^2*e^2*k*l + 9*a^2*
b*c^5*d^2*e^2*j*m + 9*a*b^4*c^3*d^2*g^2*h*m - 6*a^3*b^2*c^3*d*g*j^3*k - 3*a
^3*b^3*c^2*f*g*h*k^3 + 3*a^3*b^2*c^3*e*f*j^3*k + 3*a^3*b^2*c^3*d*f*j^3*l +
3*a^2*b^2*c^4*e*f^3*j*k + 3*a^2*b^2*c^4*d*f^3*j*l + 45*a^3*b*c^4*d^2*g*h*l^

2 + 36*a^4*b^2*c^2*e*f*g*m^3 + 36*a^4*b^2*c^2*d*f*h*m^3 - 27*a^3*b*c^4*e^2*
g*h*k^2 - 27*a^3*b*c^4*d*g^2*h^2*l - 18*a^3*b*c^4*f^2*g*h*j^2 + 18*a^3*b*c^
4*d*e^2*j*l^2 + 15*a^3*b^3*c^2*d*e*j*l^3 + 12*a^2*b^2*c^4*e*f^3*g*m + 12*a^
2*b^2*c^4*d*f^3*h*m + 9*a^3*b*c^4*f*g^2*h^2*j + 9*a^3*b*c^4*e*g^2*h^2*k + 9
*a^3*b*c^4*d*f^2*j*k^2 + 9*a^2*b*c^5*d^2*f^2*j*k + 9*a*b^5*c^2*d^2*g*h*l^2
- 9*a*b^4*c^3*d^2*g*h^2*l - 6*a^2*b^2*c^4*e*f^3*h*l + 3*a^3*b^2*c^3*f*g*h*j
^3 + 3*a^2*b^2*c^4*f^3*g*h*j + 45*a^3*b*c^4*d^2*f*g*m^2 - 27*a^2*b*c^5*d^2*
f^2*g*m + 18*a^3*b*c^4*e^2*f*g*l^2 + 15*a^3*b^3*c^2*e*f*g*l^3 - 12*a^3*b^2*
c^3*d*e*j*k^3 + 9*a^3*b*c^4*d^2*e*h*m^2 + 9*a^3*b*c^4*e*g^2*h*j^2 + 9*a^3*b
*c^4*e*f^2*h*k^2 - 9*a^2*b^3*c^3*d*f*h^3*l + 9*a^2*b*c^5*d^2*f^2*h*l + 9*a*
b^5*c^2*d^2*f*g*m^2 + 9*a*b^3*c^4*d^2*f^2*g*m + 6*a^3*b^3*c^2*d*f*h*l^3 + 3
*a^2*b^4*c^2*d*e*j*k^3 + 18*a^3*b*c^4*e*f*g^2*k^2 + 18*a^2*b*c^5*d^2*g^2*h*
j + 18*a^2*b*c^5*d^2*f*g^2*l + 18*a^2*b*c^5*d^2*e*g^2*m - 12*a^3*b^2*c^3*d*
f*h*k^3 + 9*a^3*b*c^4*e*f*h^2*j^2 + 9*a^3*b*c^4*d*f^2*g*l^2 + 9*a^3*b*c^4*d
*e^2*g*m^2 + 9*a^3*b*c^4*d*g*h^2*j^2 + 9*a^2*b^2*c^4*e*f*g^3*k + 9*a^2*b^2*
c^4*d*g^3*h*j + 9*a^2*b^2*c^4*d*f*g^3*l + 9*a^2*b^2*c^4*d*e*g^3*m + 9*a^2*b
*c^5*e^2*f^2*h*j + 9*a^2*b*c^5*e^2*f^2*g*k - 9*a*b^3*c^4*d^2*g^2*h*j - 9*a*
b^3*c^4*d^2*f*g^2*l - 9*a*b^3*c^4*d^2*e*g^2*m - 3*a^3*b^2*c^3*e*f*g*k^3 + 3
*a^2*b^4*c^2*e*f*g*k^3 + 3*a^2*b^4*c^2*d*f*h*k^3 - 54*a^3*b*c^4*d*e*f^2*m^2
 - 51*a^3*b^3*c^2*d*e*f*m^3 - 27*a^3*b*c^4*d*e*g^2*l^2 + 9*a^3*b*c^4*d*e*h^
2*k^2 + 9*a^2*b*c^5*e^2*f*g^2*j + 9*a^2*b*c^5*d^2*f*h^2*j + 9*a^2*b*c^5*d^2
*e*h^2*k + 9*a^2*b*c^5*d*e^2*g^2*l - 9*a*b^5*c^2*d*e*f^2*m^2 - 9*a*b^4*c^3*
d^2*e*g*l^2 - 9*a*b^2*c^5*d^2*e^2*g*l - 9*a*b^2*c^5*d^2*e^2*f*m - 3*a^2*b^3
*c^3*e*f*g*j^3 - 3*a^2*b^3*c^3*d*f*h*j^3 + 36*a^3*b^2*c^3*d*e*f*l^3 - 27*a^
2*b*c^5*d^2*f*g*j^2 - 18*a^2*b^4*c^2*d*e*f*l^3 - 18*a^2*b*c^5*d*e^2*h^2*j +
 9*a^2*b*c^5*d^2*e*h*j^2 + 9*a^2*b*c^5*d*f^2*g^2*j + 9*a*b^4*c^3*d*e^2*f*l^
2 + 9*a*b^3*c^4*d^2*f*g*j^2 - 9*a*b^2*c^5*d^2*f^2*g*j - 9*a*b^2*c^5*d^2*e*f
^2*l + 3*a^2*b^2*c^4*d*e*h^3*j - 18*a^2*b*c^5*e^2*f*g*h^2 + 18*a^2*b*c^5*d^
2*e*f*k^2 + 15*a^2*b^3*c^3*d*e*f*k^3 + 9*a^2*b*c^5*e*f^2*g^2*h + 9*a^2*b*c^
5*d*e^2*g*j^2 - 9*a*b^3*c^4*d^2*e*f*k^2 + 9*a*b^2*c^5*d^2*e*g^2*j - 9*a*b^2
*c^5*d*e^2*f^2*k + 3*a^2*b^2*c^4*e*f*g*h^3 + 18*a^2*b*c^5*d*e*f^2*j^2 + 9*a
^2*b*c^5*d*f^2*g*h^2 - 9*a*b^3*c^4*d*e*f^2*j^2 + 9*a*b^2*c^5*d^2*f*g^2*h -
3*a^2*b^2*c^4*d*e*f*j^3 + 9*a^2*b*c^5*d*e*g^2*h^2 - 9*a*b^2*c^5*d^2*e*g*h^2
 + 9*a*b^2*c^5*d*e^2*f*h^2 - 36*a^6*c^2*f*j*k*l*m^2 + 36*a^5*c^3*f^2*j*k*l*
m - 36*a^5*c^3*f*h^2*j*l*m + 36*a^5*c^3*e*h*j^2*l*m - 18*a^6*b*c*j^2*k*l*m^
2 + 9*a^6*b*c*j*k^2*l^2*m + 3*a^5*b^2*c*j^3*k*l*m - 36*a^5*c^3*f*g*j*k^2*m
- 36*a^5*c^3*e*f*k^2*l*m + 36*a^5*c^3*d*g*k^2*l*m - 36*a^4*c^4*d^2*g*k*l*m
- 36*a^5*c^3*e*h*j*k*l^2 - 36*a^5*c^3*e*f*j*l^2*m - 36*a^5*c^3*d*f*k*l^2*m
+ 36*a^4*c^4*e^2*h*j*k*l + 36*a^4*c^4*e^2*f*j*l*m + 9*a^6*b*c*h*k^2*l*m^2 -
 3*a^4*b^3*c*h^3*k*l*m - 36*a^5*c^3*e*g*h*l^2*m + 36*a^5*c^3*e*f*j*k*m^2 -
36*a^5*c^3*d*g*j*k*m^2 + 36*a^5*c^3*d*f*j*l*m^2 - 36*a^5*c^3*d*e*k*l*m^2 +
36*a^4*c^4*e^2*g*h*l*m - 36*a^4*c^4*e*f^2*j*k*m - 36*a^4*c^4*d*f^2*j*l*m +
9*a^6*b*c*h*j*l^2*m^2 + 9*a^6*b*c*g*k*l^2*m^2 + 9*a^5*b^2*c*g*k^3*l*m + 3*a
^3*b^4*c*g^3*k*l*m + 36*a^5*c^3*f*g*h*j*m^2 + 36*a^5*c^3*e*f*h*l*m^2 - 36*a
^4*c^4*f^2*g*h*j*m - 36*a^4*c^4*e*f^2*h*l*m - 24*a^4*b*c^3*f^3*k*l*m - 12*a

^5*b*c^2*h*j^3*k*m - 12*a^5*b*c^2*g*j^3*l*m - 3*a^2*b^5*c*f^3*k*l*m - 36*a^
4*c^4*e*g^2*h*k*l - 36*a^4*c^4*e*f*g^2*l*m + 12*a^5*b^2*c*e*k*l^3*m - 6*a^5
*b^2*c*f*j*l^3*m + 3*a^5*b^2*c*h*j*k*l^3 + 48*a^3*b*c^4*d^3*k*l*m + 36*a^4*
c^4*e*f*h^2*j*m + 36*a^4*c^4*d*g*h^2*k*l - 36*a^4*c^4*d*f*h^2*k*m - 36*a^4*
c^4*d*e*j^2*k*l + 24*a^5*b*c^2*d*k^3*l*m + 21*a*b^5*c^2*d^3*k*l*m - 12*a^5*
b*c^2*g*j*k^3*l - 9*a^4*b^3*c*d*k^3*l*m + 6*a^5*b*c^2*f*j*k^3*m + 3*a^5*b^2
*c*g*h*l^3*m - 36*a^4*c^4*e*f*h*j^2*l - 12*a^5*b*c^2*g*h*k^3*m - 3*a^5*b^2*
c*e*j*k*m^3 - 3*a^5*b^2*c*d*j*l*m^3 - 36*a^4*c^4*d*g*h*j*k^2 - 36*a^4*c^4*d
*f*g*k^2*l - 36*a^4*c^4*d*e*h*k^2*l - 36*a^4*c^4*d*e*g*k^2*m + 36*a^3*c^5*d
^2*g*h*j*k + 36*a^3*c^5*d^2*f*g*k*l - 36*a^3*c^5*d^2*f*g*j*m + 36*a^3*c^5*d
^2*e*h*k*l + 36*a^3*c^5*d^2*e*g*k*m - 36*a^3*c^5*d^2*e*f*l*m + 24*a^5*b^2*c
*e*h*l*m^3 - 24*a^3*b*c^4*e^3*j*k*l - 12*a^5*b^2*c*f*h*k*m^3 - 12*a^5*b^2*c
*f*g*l*m^3 - 3*a^5*b^2*c*g*h*j*m^3 - 3*a^4*b^3*c*e*j*k*l^3 - 3*a*b^5*c^2*e^
3*j*k*l + 36*a^4*c^4*d*e*h*j*l^2 + 36*a^4*c^4*d*e*g*k*l^2 - 36*a^3*c^5*d*e^
2*h*j*l - 36*a^3*c^5*d*e^2*g*k*l - 36*a^3*c^5*d*e^2*f*k*m + 24*a^4*b*c^3*e*
h^3*k*m - 24*a^3*b*c^4*e^3*g*l*m - 18*a*b^4*c^3*d^3*j*k*l - 12*a^4*b*c^3*g*
h^3*j*l - 12*a^4*b*c^3*f*h^3*k*l - 12*a^4*b*c^3*d*h^3*l*m + 12*a^3*b*c^4*e^
3*h*k*m + 6*a^4*b*c^3*f*h^3*j*m - 3*a^4*b^3*c*g*h*j*l^3 - 3*a^4*b^3*c*f*h*k
*l^3 - 3*a^4*b^3*c*e*g*l^3*m - 3*a^4*b^3*c*d*h*l^3*m - 3*a*b^5*c^2*e^3*h*k*
m - 3*a*b^5*c^2*e^3*g*l*m + 36*a^4*c^4*e*f*g*h*l^2 - 36*a^4*c^4*d*e*f*j*m^2
 - 36*a^3*c^5*e^2*f*g*h*l - 36*a^3*c^5*d*f^2*g*j*k - 36*a^3*c^5*d*e*f^2*k*l
 + 36*a^3*c^5*d*e*f^2*j*m - 18*a*b^4*c^3*d^3*h*k*m - 9*a*b^4*c^3*d^3*g*l*m
+ 30*a^5*b*c^2*d*g*k*m^3 - 30*a^4*b^3*c*d*g*k*m^3 - 24*a^5*b*c^2*e*f*k*m^3
- 24*a^5*b*c^2*d*f*l*m^3 + 24*a^4*b*c^3*e*g*j^3*m + 24*a^4*b*c^3*d*h*j^3*m
+ 15*a^4*b^3*c*e*f*k*m^3 + 15*a^4*b^3*c*d*f*l*m^3 + 12*a^5*b*c^2*e*g*j*m^3
+ 12*a^5*b*c^2*d*h*j*m^3 - 12*a^4*b*c^3*f*h*j^3*k - 12*a^4*b*c^3*f*g*j^3*l
+ 6*a^4*b^3*c*e*g*j*m^3 + 6*a^4*b^3*c*d*h*j*m^3 + 6*a^4*b*c^3*e*h*j^3*l + 3
6*a^3*c^5*d*e*g^2*h*l - 24*a^5*b*c^2*f*g*h*m^3 + 15*a^4*b^3*c*f*g*h*m^3 - 9
*a*b^6*c*d^2*g*j*m^2 - 6*a^3*b^4*c*d*g*k*l^3 - 6*a*b^4*c^3*e^3*f*j*m + 3*a^
3*b^4*c*e*g*j*l^3 + 3*a^3*b^4*c*e*f*k*l^3 + 3*a^3*b^4*c*d*h*j*l^3 + 3*a^3*b
^4*c*d*e*l^3*m + 3*a*b^4*c^3*e^3*h*j*k + 3*a*b^4*c^3*e^3*g*j*l + 3*a*b^4*c^
3*e^3*f*k*l + 3*a*b^4*c^3*d*e^3*l*m - 36*a^3*c^5*d*e*g*h^2*k + 30*a^2*b*c^5
*d^3*f*j*m - 30*a*b^3*c^4*d^3*f*j*m + 24*a^3*b*c^4*d*g^3*j*l - 24*a^2*b*c^5
*d^3*h*j*k - 24*a^2*b*c^5*d^3*f*k*l - 24*a^2*b*c^5*d^3*e*k*m + 15*a*b^3*c^4
*d^3*h*j*k + 15*a*b^3*c^4*d^3*f*k*l + 15*a*b^3*c^4*d^3*e*k*m - 12*a^3*b*c^4
*e*g^3*j*k + 12*a^2*b*c^5*d^3*g*j*l + 6*a*b^3*c^4*d^3*g*j*l + 3*a^3*b^4*c*f
*g*h*l^3 + 3*a*b^4*c^3*e^3*g*h*m + 24*a^3*b*c^4*d*g^3*h*m - 12*a^3*b*c^4*f*
g^3*h*k + 12*a^2*b*c^5*d^3*g*h*m - 9*a^3*b^4*c*d*e*j*m^3 + 6*a^3*b*c^4*e*g^
3*h*l + 6*a*b^3*c^4*d^3*g*h*m + 36*a^3*c^5*d*e*f*g*k^2 - 36*a^2*c^6*d^2*e*f
*g*k - 24*a^4*b*c^3*d*e*j*l^3 - 18*a^3*b^4*c*e*f*g*m^3 - 18*a^3*b^4*c*d*f*h
*m^3 - 3*a^2*b^5*c*d*e*j*l^3 - 3*a*b^3*c^4*d*e^3*j*l - 24*a^4*b*c^3*e*f*g*l
^3 + 24*a^3*b*c^4*d*f*h^3*l + 12*a^4*b*c^3*d*f*h*l^3 - 12*a^3*b*c^4*e*g*h^3
*j - 12*a^3*b*c^4*e*f*h^3*k - 12*a^3*b*c^4*d*e*h^3*m - 12*a*b^2*c^5*d^3*e*j
*k + 6*a^3*b*c^4*d*g*h^3*k - 3*a^2*b^5*c*e*f*g*l^3 - 3*a^2*b^5*c*d*f*h*l^3
- 3*a*b^3*c^4*e^3*g*h*j - 3*a*b^3*c^4*e^3*f*h*k - 3*a*b^3*c^4*e^3*f*g*l - 3

*a*b^3*c^4*d*e^3*h*m + 24*a*b^2*c^5*d^3*e*h*l - 12*a*b^2*c^5*d^3*f*h*k - 3*
a*b^2*c^5*d^3*g*h*j - 3*a*b^2*c^5*d^3*f*g*l - 3*a*b^2*c^5*d^3*e*g*m + 48*a^
4*b*c^3*d*e*f*m^3 + 24*a^2*b*c^5*d*e*f^3*m + 21*a^2*b^5*c*d*e*f*m^3 - 12*a^
2*b*c^5*e*f^3*g*j - 12*a^2*b*c^5*d*f^3*h*j - 9*a*b^3*c^4*d*e*f^3*m + 6*a^2*
b*c^5*d*f^3*g*k + 12*a*b^2*c^5*d*e^3*f*l - 6*a*b^2*c^5*d*e^3*g*k + 3*a*b^2*
c^5*d*e^3*h*j - 24*a^3*b*c^4*d*e*f*k^3 - 12*a^2*b*c^5*d*e*g^3*j - 3*a*b^5*c
^2*d*e*f*k^3 + 3*a*b^2*c^5*e^3*f*g*h - 12*a^2*b*c^5*d*f*g^3*h + 9*a*b^2*c^5
*d*e*f^3*j + 9*a*b*c^6*d^2*e^2*f*j + 3*a*b^4*c^3*d*e*f*j^3 + 9*a*b*c^6*d^2*
e^2*g*h + 9*a*b*c^6*d^2*e*f^2*h - 3*a*b^3*c^4*d*e*f*h^3 - 18*a*b*c^6*d^2*e*
f*g^2 + 9*a*b*c^6*d*e^2*f^2*g + 3*a*b^2*c^5*d*e*f*g^3 - 36*a^4*b^2*c^2*e^2*
k*l^2*m - 9*a^4*b^2*c^2*g^2*j^2*k*m + 45*a^3*b^3*c^2*d^2*k^2*l*m + 36*a^4*b
^2*c^2*e^2*j*l*m^2 + 9*a^4*b^2*c^2*g^2*j*k^2*l + 9*a^3*b^3*c^2*e^2*j^2*l*m
+ 9*a^4*b^2*c^2*g^2*h*k^2*m - 9*a^4*b^2*c^2*f^2*h*l^2*m - 9*a^3*b^3*c^2*f^2
*j^2*k*l - 45*a^3*b^3*c^2*d^2*j*k*m^2 + 36*a^3*b^2*c^3*d^2*j^2*k*m + 18*a^4
*b^2*c^2*f^2*h*k*m^2 + 18*a^4*b^2*c^2*f^2*g*l*m^2 - 9*a^4*b^2*c^2*g^2*h*k*l
^2 - 9*a^4*b^2*c^2*f*h^2*k^2*m - 9*a^4*b^2*c^2*f*g^2*l^2*m - 9*a^4*b^2*c^2*
e*j^2*k^2*l - 9*a^4*b^2*c^2*d*j^2*k^2*m - 9*a^3*b^3*c^2*e^2*j*k*l^2 - 9*a^2
*b^4*c^2*d^2*j^2*k*m - 36*a^3*b^2*c^3*d^2*j*k^2*l - 27*a^3*b^2*c^3*e^2*h^2*
k*m + 9*a^4*b^2*c^2*g*h^2*j*l^2 + 9*a^4*b^2*c^2*f*h^2*k*l^2 - 9*a^4*b^2*c^2
*f*g^2*k*m^2 - 9*a^4*b^2*c^2*e*g^2*l*m^2 - 9*a^4*b^2*c^2*d*j^2*k*l^2 + 9*a^
4*b^2*c^2*d*h^2*l^2*m - 9*a^3*b^3*c^2*e^2*g*l^2*m + 9*a^2*b^4*c^2*e^2*h^2*k
*m + 9*a^2*b^4*c^2*d^2*j*k^2*l - 45*a^3*b^3*c^2*e^2*h*j*m^2 + 36*a^4*b^2*c^
2*e*h^2*j*m^2 + 36*a^3*b^2*c^3*e^2*h*j^2*m - 36*a^3*b^2*c^3*d^2*h*k^2*m + 3
6*a^2*b^3*c^3*d^2*g^2*l*m - 9*a^4*b^2*c^2*f*h*j^2*l^2 - 9*a^4*b^2*c^2*d*h^2
*k*m^2 + 9*a^3*b^3*c^2*f^2*h*j*l^2 + 9*a^3*b^3*c^2*e^2*f*l*m^2 + 9*a^3*b^3*
c^2*e*h^2*j^2*m - 9*a^3*b^2*c^3*f^2*h^2*j*l - 9*a^2*b^4*c^2*e^2*h*j^2*m + 9
*a^2*b^4*c^2*d^2*h*k^2*m + 36*a^3*b^2*c^3*d^2*h*k*l^2 - 27*a^4*b^2*c^2*e*g*
j^2*m^2 - 27*a^4*b^2*c^2*d*h*j^2*m^2 - 9*a^4*b^2*c^2*d*h*k^2*l^2 - 9*a^3*b^
3*c^2*e*f^2*k*m^2 - 9*a^3*b^3*c^2*d*f^2*l*m^2 + 9*a^3*b^2*c^3*f^2*h*j^2*k +
 9*a^3*b^2*c^3*f^2*g*j^2*l - 9*a^3*b^2*c^3*e^2*g*k^2*l - 9*a^3*b^2*c^3*e^2*
f*k^2*m - 9*a^3*b^2*c^3*d^2*f*l^2*m - 9*a^2*b^4*c^2*d^2*h*k*l^2 + 9*a^2*b^3
*c^3*d^2*h^2*k*l - 81*a^3*b^2*c^3*d^2*g*j*m^2 + 54*a^2*b^4*c^2*d^2*g*j*m^2
- 45*a^3*b^3*c^2*d*g^2*j*m^2 - 45*a^2*b^3*c^3*d^2*g*j^2*m + 36*a^3*b^2*c^3*
d^2*f*k*m^2 + 36*a^3*b^2*c^3*d*g^2*j^2*m + 18*a^3*b^2*c^3*e^2*g*j*l^2 + 18*
a^3*b^2*c^3*e^2*f*k*l^2 + 18*a^3*b^2*c^3*d*e^2*l^2*m - 9*a^4*b^2*c^2*d*f*k^
2*m^2 - 9*a^3*b^3*c^2*f^2*g*h*m^2 - 9*a^3*b^3*c^2*d*h^2*j*l^2 - 9*a^3*b^2*c
^3*f^2*g*j*k^2 - 9*a^3*b^2*c^3*d^2*e*l*m^2 - 9*a^3*b^2*c^3*f*g^2*h^2*m - 9*
a^3*b^2*c^3*e*g^2*j^2*l - 9*a^3*b^2*c^3*e*f^2*k^2*l - 9*a^2*b^4*c^2*d^2*f*k
*m^2 - 9*a^2*b^4*c^2*d*g^2*j^2*m - 9*a^2*b^3*c^3*e^2*h^2*j*k - 9*a^2*b^2*c^
4*d^2*f^2*k*m - 27*a^2*b^2*c^4*d^2*g^2*j*l - 9*a^3*b^3*c^2*f*g*h^2*l^2 + 9*
a^3*b^2*c^3*e*g^2*j*k^2 - 9*a^3*b^2*c^3*e*f^2*j*l^2 - 9*a^3*b^2*c^3*d*h^2*j
^2*k - 9*a^3*b^2*c^3*d*f^2*k*l^2 - 9*a^3*b^2*c^3*d*e^2*k*m^2 - 9*a^2*b^3*c^
3*e^2*g*h^2*m - 9*a^2*b^3*c^3*d^2*h*j*k^2 - 9*a^2*b^3*c^3*d^2*f*k^2*l - 9*a
^2*b^3*c^3*d^2*e*k^2*m + 36*a^3*b^3*c^2*d*e*j^2*m^2 + 36*a^3*b^2*c^3*e^2*f*
h*m^2 - 27*a^2*b^2*c^4*d^2*g^2*h*m + 9*a^3*b^3*c^2*e*f*h^2*m^2 + 9*a^3*b^2*

c^3*f*g^2*h*k^2 - 9*a^2*b^4*c^2*e^2*f*h*m^2 + 9*a^2*b^3*c^3*d^2*e*k*l^2 - 9
*a^2*b^2*c^4*e^2*f^2*h*m - 45*a^2*b^3*c^3*d^2*g*h*l^2 - 36*a^3*b^2*c^3*e*f^
2*g*m^2 + 36*a^3*b^2*c^3*d*g^2*h*l^2 - 36*a^3*b^2*c^3*d*f^2*h*m^2 + 36*a^2*
b^2*c^4*d^2*g*h^2*l - 9*a^3*b^2*c^3*e*g*h^2*k^2 + 9*a^2*b^4*c^2*e*f^2*g*m^2
 - 9*a^2*b^4*c^2*d*g^2*h*l^2 + 9*a^2*b^4*c^2*d*f^2*h*m^2 + 9*a^2*b^3*c^3*e^
2*g*h*k^2 + 9*a^2*b^3*c^3*d*g^2*h^2*l - 9*a^2*b^3*c^3*d*e^2*j*l^2 - 9*a^2*b
^2*c^4*e^2*g^2*h*k - 9*a^2*b^2*c^4*e^2*f*g^2*m - 9*a^2*b^2*c^4*d^2*f*j^2*k
- 9*a^2*b^2*c^4*d^2*f*h^2*m - 9*a^2*b^2*c^4*d^2*e*j^2*l - 45*a^2*b^3*c^3*d^
2*f*g*m^2 + 36*a^3*b^2*c^3*d*f*g^2*m^2 - 27*a^3*b^2*c^3*d*f*h^2*l^2 + 18*a^
2*b^2*c^4*d^2*e*j*k^2 + 9*a^2*b^4*c^2*d*f*h^2*l^2 - 9*a^2*b^4*c^2*d*f*g^2*m
^2 - 9*a^2*b^3*c^3*e^2*f*g*l^2 + 9*a^2*b^2*c^4*e^2*g*h^2*j + 9*a^2*b^2*c^4*
e^2*f*h^2*k - 9*a^2*b^2*c^4*e*f^2*g^2*l - 9*a^2*b^2*c^4*d*f^2*g^2*m - 9*a^2
*b^2*c^4*d*e^2*j^2*k + 9*a^2*b^2*c^4*d*e^2*h^2*m + 18*a^4*b^2*c^2*f^2*j^2*m
^2 + 18*a^3*b^2*c^3*e^2*h^2*l^2 - 9*a^2*b^4*c^2*e^2*h^2*l^2 + 18*a^2*b^2*c^
4*d^2*g^2*k^2 + 12*a^6*c^2*j^3*k*l*m + 3*a^6*b^2*j*k*l*m^3 - 12*a^6*c^2*g*k
^3*l*m - 12*a^5*c^3*g^3*k*l*m - 24*a^6*c^2*e*k*l^3*m - 24*a^4*c^4*e^3*k*l*m
 + 12*a^6*c^2*h*j*k*l^3 + 12*a^6*c^2*f*j*l^3*m + 12*a^5*c^3*h^3*j*k*l - 3*a
^5*b^3*h*j*k*m^3 - 3*a^5*b^3*g*j*l*m^3 - 3*a^5*b^3*f*k*l*m^3 + 12*a^6*c^2*g
*h*l^3*m + 12*a^5*c^3*g*h^3*l*m - 12*a^6*c^2*e*j*k*m^3 - 12*a^6*c^2*d*j*l*m
^3 - 12*a^5*c^3*f*j^3*k*l - 12*a^5*c^3*e*j^3*k*m - 12*a^5*c^3*d*j^3*l*m - 1
2*a^4*c^4*f^3*j*k*l + 24*a^6*c^2*f*h*k*m^3 + 24*a^6*c^2*f*g*l*m^3 + 24*a^4*
c^4*f^3*h*k*m + 24*a^4*c^4*f^3*g*l*m - 12*a^6*c^2*g*h*j*m^3 - 12*a^6*c^2*e*
h*l*m^3 - 12*a^5*c^3*g*h*j^3*m + 3*b^6*c^2*d^3*j*k*l + 3*a^4*b^4*e*j*k*m^3
+ 3*a^4*b^4*d*j*l*m^3 - 24*a^5*c^3*d*j*k^3*l - 24*a^3*c^5*d^3*j*k*l - 6*a^4
*b^4*e*h*l*m^3 + 3*b^6*c^2*d^3*h*k*m + 3*b^6*c^2*d^3*g*l*m + 3*a^6*b*c*j^2*
l^3*m + 3*a^4*b^4*g*h*j*m^3 + 3*a^4*b^4*f*h*k*m^3 + 3*a^4*b^4*f*g*l*m^3 - 2
4*a^5*c^3*d*h*k^3*m - 24*a^3*c^5*d^3*h*k*m + 12*a^5*c^3*g*h*j*k^3 + 12*a^5*
c^3*f*g*k^3*l + 12*a^5*c^3*e*h*k^3*l + 12*a^5*c^3*e*g*k^3*m + 12*a^4*c^4*g^
3*h*j*k + 12*a^4*c^4*f*g^3*k*l + 12*a^4*c^4*f*g^3*j*m + 12*a^4*c^4*e*g^3*k*
m + 12*a^4*c^4*d*g^3*l*m + 12*a^3*c^5*d^3*g*l*m + 3*a^6*b*c*j*k^3*m^2 - 9*a
^6*b*c*h^2*l*m^3 - 3*a^5*b*c^2*j^4*k*l + 24*a^5*c^3*e*g*j*l^3 + 24*a^5*c^3*
e*f*k*l^3 + 24*a^5*c^3*d*e*l^3*m + 24*a^3*c^5*e^3*g*j*l + 24*a^3*c^5*e^3*f*
k*l + 24*a^3*c^5*d*e^3*l*m - 12*a^5*c^3*d*h*j*l^3 - 12*a^5*c^3*d*g*k*l^3 -
12*a^4*c^4*e*h^3*j*k - 12*a^4*c^4*d*h^3*j*l - 12*a^3*c^5*e^3*h*j*k - 12*a^3
*c^5*e^3*f*j*m + 9*a^4*b*c^3*g^4*l*m + 6*b^5*c^3*d^3*f*j*m + 6*a^3*b^5*d*g*
k*m^3 - 3*b^5*c^3*d^3*h*j*k - 3*b^5*c^3*d^3*g*j*l - 3*b^5*c^3*d^3*f*k*l - 3
*b^5*c^3*d^3*e*k*m - 3*a^3*b^5*e*g*j*m^3 - 3*a^3*b^5*e*f*k*m^3 - 3*a^3*b^5*
d*h*j*m^3 - 3*a^3*b^5*d*f*l*m^3 - 12*a^5*c^3*f*g*h*l^3 - 12*a^4*c^4*f*g*h^3
*l - 12*a^4*c^4*e*g*h^3*m - 12*a^3*c^5*e^3*g*h*m - 9*a^6*b*c*g*k^2*m^3 - 3*
b^5*c^3*d^3*g*h*m + 3*a^6*b*c*f*l^3*m^2 - 3*a^3*b^5*f*g*h*m^3 + 12*a^5*c^3*
d*e*j*m^3 + 12*a^4*c^4*e*f*j^3*k + 12*a^4*c^4*d*g*j^3*k + 12*a^4*c^4*d*f*j^
3*l + 12*a^4*c^4*d*e*j^3*m + 12*a^3*c^5*e*f^3*j*k + 12*a^3*c^5*d*f^3*j*l -
9*a^6*b*c*e*l^2*m^3 - 24*a^5*c^3*e*f*g*m^3 - 24*a^5*c^3*d*f*h*m^3 - 24*a^3*
c^5*e*f^3*g*m - 24*a^3*c^5*d*f^3*h*m - 15*a^2*b*c^5*d^4*l*m + 15*a*b^3*c^4*
d^4*l*m + 12*a^4*c^4*f*g*h*j^3 + 12*a^3*c^5*f^3*g*h*j + 12*a^3*c^5*e*f^3*h*

l + 9*a^3*b*c^4*f^4*k*l - 9*a^3*b*c^4*f^4*j*m + 3*b^4*c^4*d^3*e*j*k + 3*a^5
*b^2*c*g*j*l^4 + 3*a^5*b^2*c*f*k*l^4 + 3*a^5*b^2*c*d*l^4*m - 3*a^5*b*c^2*h*
j*k^4 - 3*a^5*b*c^2*f*k^4*l - 3*a^5*b*c^2*e*k^4*m - 3*a^4*b*c^3*h^4*j*k + 3
*a^2*b^6*d*e*j*m^3 + 3*a*b^4*c^3*e^4*k*m + 24*a^4*c^4*d*e*j*k^3 + 24*a^2*c^
6*d^3*e*j*k - 6*b^4*c^4*d^3*e*h*l + 3*b^4*c^4*d^3*g*h*j + 3*b^4*c^4*d^3*f*h
*k + 3*b^4*c^4*d^3*f*g*l + 3*b^4*c^4*d^3*e*g*m - 3*a^4*b*c^3*g*h^4*m + 3*a^
2*b^6*e*f*g*m^3 + 3*a^2*b^6*d*f*h*m^3 - 3*a*b^6*c*e^3*j*m^2 + 24*a^4*c^4*d*
f*h*k^3 + 24*a^2*c^6*d^3*f*h*k - 12*a^4*c^4*e*f*g*k^3 - 12*a^3*c^5*e*f*g^3*
k - 12*a^3*c^5*d*g^3*h*j - 12*a^3*c^5*d*f*g^3*l - 12*a^3*c^5*d*e*g^3*m - 12
*a^2*c^6*d^3*g*h*j - 12*a^2*c^6*d^3*f*g*l - 12*a^2*c^6*d^3*e*h*l - 12*a^2*c
^6*d^3*e*g*m - 12*a*b^2*c^5*d^4*j*l + 9*a^5*b*c^2*d*j*l^4 + 9*a^2*b*c^5*e^4
*j*k - 3*a^4*b^3*c*d*j*l^4 - 3*a^4*b*c^3*e*j^4*k - 3*a^4*b*c^3*d*j^4*l - 3*
a*b^3*c^4*e^4*j*k - 24*a^4*c^4*d*e*f*l^3 - 24*a^2*c^6*d*e^3*f*l - 12*a^5*b^
2*c*e*g*m^4 - 12*a^5*b^2*c*d*h*m^4 + 12*a^3*c^5*d*e*h^3*j + 12*a^2*c^6*d*e^
3*h*j + 12*a^2*c^6*d*e^3*g*k - 12*a*b^2*c^5*d^4*h*m + 9*a^5*b*c^2*f*g*l^4 -
 9*a^5*b*c^2*e*h*l^4 - 9*a^2*b*c^5*e^4*h*l + 9*a^2*b*c^5*e^4*g*m + 6*a^4*b^
3*c*e*h*l^4 + 6*a*b^3*c^4*e^4*h*l - 3*b^3*c^5*d^3*e*g*j - 3*b^3*c^5*d^3*e*f
*k - 3*a^4*b^3*c*f*g*l^4 - 3*a^4*b*c^3*g*h*j^4 - 3*a^3*b*c^4*g^4*h*j - 3*a^
3*b*c^4*f*g^4*l - 3*a^3*b*c^4*e*g^4*m - 3*a*b^3*c^4*e^4*g*m + 12*a^3*c^5*e*
f*g*h^3 + 12*a^2*c^6*e^3*f*g*h - 3*b^3*c^5*d^3*f*g*h - 12*a^3*c^5*d*e*f*j^3
 - 12*a^2*c^6*d*e*f^3*j - 3*a*b^6*c*d^2*g*l^3 - 15*a^5*b*c^2*d*e*m^4 + 15*a
^4*b^3*c*d*e*m^4 + 9*a^4*b*c^3*e*f*k^4 - 9*a^4*b*c^3*d*g*k^4 + 3*a^3*b^4*c*
d*f*l^4 - 3*a^3*b*c^4*d*h^4*j - 3*a^2*b*c^5*e*f^4*k - 3*a^2*b*c^5*d*f^4*l +
 3*a*b^2*c^5*e^4*g*j + 3*a*b^2*c^5*e^4*f*k + 3*a*b^2*c^5*d*e^4*m - 9*a*b*c^
6*d^3*e^2*l + 3*b^2*c^6*d^3*e*f*g - 3*a^3*b*c^4*f*g*h^4 - 3*a^2*b*c^5*f^4*g
*h + 12*a^2*c^6*d*e*f*g^3 - 9*a*b*c^6*d^3*f^2*j + 3*a*b*c^6*d^2*e^3*k + 9*a
^3*b*c^4*d*e*j^4 - 3*a^2*b*c^5*e*f*g^4 - 9*a*b*c^6*d^3*e*h^2 + 3*a*b*c^6*d^
2*f^3*g + 3*a*b*c^6*d*e^3*g^2 - 3*a^4*b^2*c^2*h^3*j^2*m + 12*a^4*b^2*c^2*g^
3*j*m^2 - 3*a^4*b^2*c^2*f^2*k^3*m + 3*a^3*b^3*c^2*g^3*j^2*m - 9*a^3*b^4*c*f
^2*j^2*m^2 + 9*a^3*b^3*c^2*f^2*j^3*m - 6*a^3*b^3*c^2*f^3*j*m^2 - 6*a^3*b^2*
c^3*f^3*j^2*m - 3*a^2*b^4*c^2*f^3*j^2*m - 27*a^4*b^2*c^2*d^2*k*m^3 - 27*a^3
*b^2*c^3*e^3*j*m^2 + 18*a^2*b^4*c^2*e^3*j*m^2 - 15*a^2*b^3*c^3*e^3*j^2*m +
12*a^4*b^2*c^2*f^2*j*l^3 + 3*a^3*b^3*c^2*e^2*k^3*l + 42*a^2*b^3*c^3*d^3*j*m
^2 - 27*a^2*b^2*c^4*d^3*j^2*m - 15*a^3*b^3*c^2*d^2*k*l^3 - 3*a^4*b^2*c^2*f*
j^2*k^3 - 3*a^4*b^2*c^2*f*h^3*m^2 + 3*a^3*b^3*c^2*g^3*h*l^2 + 3*a^3*b^3*c^2
*f^2*j*k^3 - 3*a^3*b^2*c^3*g^3*h^2*l - 3*a^3*b^2*c^3*e^2*j^3*l - 27*a^4*b^2
*c^2*e^2*h*m^3 + 12*a^3*b^2*c^3*f^3*h*l^2 + 3*a^3*b^3*c^2*f*g^3*m^2 - 3*a^2
*b^4*c^2*f^3*h*l^2 + 3*a^2*b^3*c^3*f^3*h^2*l + 9*a^3*b^3*c^2*e*h^3*l^2 + 9*
a^2*b^3*c^3*e^2*h^3*l - 6*a^4*b^2*c^2*e*h^2*l^3 - 6*a^3*b^3*c^2*e^2*h*l^3 -
 6*a^2*b^3*c^3*e^3*h*l^2 - 6*a^2*b^2*c^4*e^3*h^2*l + 3*a^2*b^3*c^3*d^2*j^3*
k + 42*a^3*b^3*c^2*d^2*g*m^3 - 27*a^4*b^2*c^2*d*g^2*m^3 - 27*a^2*b^2*c^4*d^
3*h*l^2 - 15*a^2*b^3*c^3*e^3*f*m^2 + 12*a^3*b^2*c^3*e^2*h*k^3 + 3*a^3*b^3*c
^2*e*h^2*k^3 - 3*a^3*b^2*c^3*e*g^3*l^2 - 3*a^2*b^4*c^2*e^2*h*k^3 + 3*a^2*b^
3*c^3*f^3*g*k^2 - 3*a^2*b^2*c^4*f^3*g^2*k - 27*a^3*b^2*c^3*d^2*g*l^3 - 27*a
^2*b^2*c^4*d^3*f*m^2 + 18*a^2*b^4*c^2*d^2*g*l^3 - 15*a^3*b^3*c^2*d*g^2*l^3

+ 12*a^2*b^2*c^4*e^3*g*k^2 - 3*a^3*b^2*c^3*e*h^2*j^3 + 3*a^2*b^3*c^3*e^2*h*
j^3 + 3*a^2*b^3*c^3*e*f^3*l^2 - 3*a^2*b^2*c^4*d^2*h^3*k + 9*a^2*b^3*c^3*d*g
^3*k^2 - 9*a*b^4*c^3*d^2*g^2*k^2 - 6*a^3*b^2*c^3*d*g^2*k^3 - 6*a^2*b^3*c^3*
d^2*g*k^3 - 3*a^2*b^4*c^2*d*g^2*k^3 + 12*a^2*b^2*c^4*d^2*g*j^3 + 3*a^2*b^3*
c^3*d*g^2*j^3 - 3*a^2*b^2*c^4*d*f^3*k^2 - 3*a^2*b^2*c^4*d*g^2*h^3 + 12*a^7*
c*j*k*l*m^3 - 3*b^7*c*d^3*k*l*m - 3*a^6*b*c*k^4*l*m - 3*a^6*b*c*j*k*l^4 - 3
*a^6*b*c*g*l^4*m - 9*a^6*b*c*f*j*m^4 + 9*a^6*b*c*e*k*m^4 + 9*a^6*b*c*d*l*m^
4 + 9*a^6*b*c*g*h*m^4 - 3*a*b^7*d*e*f*m^3 + 9*a*b*c^6*d^4*h*j - 9*a*b*c^6*d
^4*g*k + 9*a*b*c^6*d^4*f*l + 9*a*b*c^6*d^4*e*m + 12*a*c^7*d^3*e*f*g - 3*a*b
*c^6*d*e^4*j - 3*a*b*c^6*e^4*f*g - 3*a*b*c^6*d*e*f^4 + 18*a^6*c^2*h^2*j*l*m
^2 - 18*a^6*c^2*h*j^2*l^2*m + 18*a^6*c^2*f*k^2*l^2*m + 36*a^5*c^3*e^2*k*l^2
*m + 18*a^6*c^2*g*j*k^2*m^2 + 18*a^6*c^2*e*k^2*l*m^2 + 18*a^5*c^3*g^2*j^2*k
*m + 18*a^6*c^2*e*j*l^2*m^2 + 18*a^6*c^2*d*k*l^2*m^2 - 18*a^5*c^3*e^2*j*l*m
^2 - 18*a^6*c^2*f*h*l^2*m^2 + 18*a^5*c^3*f^2*h*l^2*m - 36*a^5*c^3*f^2*h*k*m
^2 - 36*a^5*c^3*f^2*g*l*m^2 + 18*a^5*c^3*g^2*h*k*l^2 - 18*a^5*c^3*g*h^2*k^2
*l + 18*a^5*c^3*f*h^2*k^2*m + 18*a^5*c^3*f*g^2*l^2*m + 18*a^5*c^3*e*j^2*k^2
*l + 18*a^5*c^3*d*j^2*k^2*m - 18*a^4*c^4*d^2*j^2*k*m + 36*a^4*c^4*d^2*j*k^2
*l + 18*a^5*c^3*f*g^2*k*m^2 + 18*a^5*c^3*e*g^2*l*m^2 + 18*a^5*c^3*d*j^2*k*l
^2 - 18*a^4*c^4*f^2*g^2*k*m + 36*a^4*c^4*d^2*h*k^2*m + 18*a^5*c^3*f*h*j^2*l
^2 - 18*a^5*c^3*e*h^2*j*m^2 + 18*a^5*c^3*d*h^2*k*m^2 + 18*a^4*c^4*f^2*h^2*j
*l - 18*a^4*c^4*e^2*h*j^2*m - 18*a^5*c^3*e*g*k^2*l^2 + 18*a^5*c^3*d*h*k^2*l
^2 + 18*a^4*c^4*e^2*g*k^2*l + 18*a^4*c^4*e^2*f*k^2*m - 18*a^4*c^4*d^2*h*k*l
^2 + 18*a^4*c^4*d^2*f*l^2*m - 36*a^4*c^4*e^2*g*j*l^2 - 36*a^4*c^4*e^2*f*k*l
^2 - 36*a^4*c^4*d*e^2*l^2*m + 18*a^5*c^3*d*f*k^2*m^2 + 18*a^4*c^4*f^2*g*j*k
^2 + 18*a^4*c^4*d^2*g*j*m^2 - 18*a^4*c^4*d^2*f*k*m^2 + 18*a^4*c^4*d^2*e*l*m
^2 - 18*a^4*c^4*f*g^2*j^2*k + 18*a^4*c^4*f*g^2*h^2*m + 18*a^4*c^4*e*g^2*j^2
*l + 18*a^4*c^4*e*f^2*k^2*l - 18*a^4*c^4*d*g^2*j^2*m - 18*a^4*c^4*d*f^2*k^2
*m + 18*a^3*c^5*d^2*f^2*k*m + 3*a^4*b^2*c^2*h^4*k*m - 3*a^3*b^3*c^2*g^4*l*m
 + 18*a^4*c^4*e*f^2*j*l^2 + 18*a^4*c^4*d*h^2*j^2*k + 18*a^4*c^4*d*f^2*k*l^2
 + 18*a^4*c^4*d*e^2*k*m^2 - 18*a^3*c^5*e^2*f^2*j*l + 12*a^5*b^2*c*g^2*k*m^3
 - 9*a^5*b*c^2*h^3*j*m^2 - 9*a^5*b*c^2*f^2*l^3*m + 3*a^5*b*c^2*h^2*k^3*l +
3*a^4*b^3*c*h^3*j*m^2 + 3*a^4*b^3*c*f^2*l^3*m - 18*a^4*c^4*e^2*f*h*m^2 + 18
*a^3*c^5*e^2*f^2*h*m + 15*a^5*b*c^2*e^2*l*m^3 - 15*a^4*b^3*c*e^2*l*m^3 - 9*
a^5*b*c^2*g^2*k*l^3 - 9*a^4*b*c^3*g^3*j^2*m - 3*a^5*b^2*c*g*k^2*l^3 + 3*a^5
*b*c^2*h*j^3*l^2 + 3*a^4*b^3*c*g^2*k*l^3 - 3*a^3*b^4*c*g^3*j*m^2 + 36*a^4*c
^4*e*f^2*g*m^2 + 36*a^4*c^4*d*f^2*h*m^2 + 18*a^4*c^4*e*g*h^2*k^2 - 18*a^4*c
^4*d*g^2*h*l^2 - 18*a^4*c^4*d*f*j^2*k^2 + 18*a^3*c^5*e^2*g^2*h*k + 18*a^3*c
^5*e^2*f*g^2*m - 18*a^3*c^5*d^2*g*h^2*l + 18*a^3*c^5*d^2*f*j^2*k + 18*a^3*c
^5*d^2*f*h^2*m + 18*a^3*c^5*d^2*e*j^2*l - 12*a^2*b^2*c^4*e^4*k*m + 9*a^4*b^
3*c*f*j^3*m^2 - 9*a^4*b^2*c^2*f*j^4*m - 6*a^5*b^2*c*f*j^2*m^3 + 6*a^5*b*c^2
*f^2*j*m^3 - 6*a^5*b*c^2*f*j^3*m^2 - 6*a^4*b^3*c*f^2*j*m^3 + 6*a^4*b*c^3*f^
3*j*m^2 - 6*a^4*b*c^3*f^2*j^3*m + 6*a^2*b^3*c^3*f^4*j*m + 3*a^3*b^2*c^3*g^4
*j*l + 3*a^2*b^5*c*f^3*j*m^2 - 3*a^2*b^3*c^3*f^4*k*l - 36*a^3*c^5*d^2*e*j*k
^2 - 18*a^4*c^4*d*f*g^2*m^2 + 18*a^3*c^5*e*f^2*g^2*l + 18*a^3*c^5*d*f^2*g^2
*m + 18*a^3*c^5*d*e^2*j^2*k + 18*a^3*b^4*c*d^2*k*m^3 + 15*a^3*b*c^4*e^3*j^2

*m + 12*a^5*b^2*c*d*k^2*m^3 - 9*a^5*b*c^2*f*j^2*l^3 - 9*a^4*b*c^3*e^2*k^3*l
 + 3*a^5*b*c^2*e*k^3*l^2 + 3*a^4*b^3*c*f*j^2*l^3 + 3*a^4*b*c^3*g^2*j^3*k -
3*a^3*b^4*c*f^2*j*l^3 + 3*a^3*b^2*c^3*g^4*h*m + 3*a*b^5*c^2*e^3*j^2*m - 36*
a^3*c^5*d^2*f*h*k^2 - 21*a^3*b*c^4*d^3*j*m^2 - 21*a*b^5*c^2*d^3*j*m^2 + 18*
a^3*c^5*e^2*f*h*j^2 - 18*a^3*c^5*e*f^2*h^2*j + 18*a^3*c^5*d*f^2*h^2*k + 18*
a*b^4*c^3*d^3*j^2*m + 15*a^4*b*c^3*d^2*k*l^3 - 9*a^5*b*c^2*d*k^2*l^3 - 9*a^
4*b*c^3*g^3*h*l^2 - 9*a^4*b*c^3*f^2*j*k^3 + 3*a^4*b^3*c*d*k^2*l^3 + 3*a^2*b
^5*c*d^2*k*l^3 - 18*a^3*c^5*d^2*e*g*l^2 + 18*a^3*c^5*d*e^2*h*k^2 + 18*a^3*b
^4*c*e^2*h*m^3 - 18*a^2*c^6*d^2*e^2*h*k + 18*a^2*c^6*d^2*e^2*g*l + 18*a^2*c
^6*d^2*e^2*f*m + 15*a^5*b*c^2*e*h^2*m^3 - 15*a^4*b^3*c*e*h^2*m^3 - 9*a^4*b*
c^3*f*g^3*m^2 - 9*a^3*b*c^4*f^3*h^2*l + 3*a^4*b^2*c^2*e*j*k^4 + 3*a^4*b*c^3
*g*h^3*k^2 + 3*a^3*b*c^4*f^2*g^3*m + 36*a^3*c^5*d*e^2*f*l^2 + 18*a^3*c^5*d*
f*g^2*j^2 + 18*a^2*c^6*d^2*f^2*g*j + 18*a^2*c^6*d^2*e*f^2*l - 9*a^3*b^2*c^3
*e*h^4*l - 9*a^3*b*c^4*d^2*j^3*k + 6*a^4*b*c^3*e^2*h*l^3 - 6*a^4*b*c^3*e*h^
3*l^2 + 6*a^3*b*c^4*e^3*h*l^2 - 6*a^3*b*c^4*e^2*h^3*l + 3*a^4*b^2*c^2*f*h*k
^4 + 3*a^4*b*c^3*d*j^3*k^2 - 3*a^3*b^4*c*e*h^2*l^3 + 3*a^2*b^5*c*e^2*h*l^3
+ 3*a^2*b^2*c^4*f^4*h*k + 3*a^2*b^2*c^4*f^4*g*l + 3*a*b^5*c^2*e^3*h*l^2 - 3
*a*b^4*c^3*e^3*h^2*l - 21*a^4*b*c^3*d^2*g*m^3 - 21*a^2*b^5*c*d^2*g*m^3 + 18
*a^3*b^4*c*d*g^2*m^3 + 18*a^2*c^6*d*e^2*f^2*k + 18*a*b^4*c^3*d^3*h*l^2 + 15
*a^3*b*c^4*e^3*f*m^2 + 15*a^2*b*c^5*d^3*h^2*l - 15*a*b^3*c^4*d^3*h^2*l - 9*
a^4*b*c^3*e*h^2*k^3 - 9*a^3*b*c^4*f^3*g*k^2 - 9*a^2*b*c^5*e^3*f^2*m + 3*a^3
*b*c^4*f^2*h^3*j + 3*a*b^5*c^2*e^3*f*m^2 + 3*a*b^3*c^4*e^3*f^2*m + 18*a*b^4
*c^3*d^3*f*m^2 + 15*a^4*b*c^3*d*g^2*l^3 + 12*a*b^2*c^5*d^3*f^2*m - 9*a^3*b*
c^4*e^2*h*j^3 - 9*a^3*b*c^4*e*f^3*l^2 - 9*a^2*b*c^5*e^3*g^2*k + 3*a^3*b*c^4
*f*g^3*j^2 + 3*a^2*b^5*c*d*g^2*l^3 + 3*a^2*b*c^5*e^2*f^3*l - 3*a*b^4*c^3*e^
3*g*k^2 + 3*a*b^3*c^4*e^3*g^2*k + 18*a^2*c^6*d^2*e*g*h^2 - 18*a^2*c^6*d*e^2
*g^2*h - 12*a^4*b^2*c^2*d*f*l^4 - 9*a^2*b^2*c^4*d*g^4*k + 9*a*b^3*c^4*d^2*g
^3*k + 6*a^3*b^3*c^2*d*g*k^4 + 6*a^3*b*c^4*d^2*g*k^3 - 6*a^3*b*c^4*d*g^3*k^
2 + 6*a^2*b*c^5*d^3*g*k^2 - 6*a^2*b*c^5*d^2*g^3*k - 6*a*b^3*c^4*d^3*g*k^2 -
 6*a*b^2*c^5*d^3*g^2*k - 3*a^3*b^3*c^2*e*f*k^4 + 3*a^3*b^2*c^3*e*g*j^4 + 3*
a^3*b^2*c^3*d*h*j^4 + 3*a*b^5*c^2*d^2*g*k^3 + 15*a^2*b*c^5*d^3*e*l^2 - 15*a
*b^3*c^4*d^3*e*l^2 - 9*a^3*b*c^4*d*g^2*j^3 - 9*a^2*b*c^5*e^3*f*j^2 - 3*a*b^
4*c^3*d^2*g*j^3 + 3*a*b^3*c^4*e^3*f*j^2 - 3*a*b^2*c^5*e^3*f^2*j + 12*a*b^2*
c^5*d^3*f*j^2 - 9*a^2*b*c^5*d*e^3*k^2 + 3*a^2*b*c^5*e^2*g^3*h + 3*a*b^3*c^4
*d*e^3*k^2 - 9*a^2*b*c^5*d^2*g*h^3 - 3*a^2*b^3*c^3*d*e*j^4 + 3*a^2*b*c^5*e*
f^3*h^2 + 3*a*b^3*c^4*d^2*g*h^3 + 3*a^2*b^2*c^4*d*f*h^4 - 9*a^7*c*k^2*l^2*m
^2 - 6*a^6*c^2*j^2*k^3*m - 3*a^6*b^2*h*l^2*m^3 + 3*a^5*b^3*h^2*l*m^3 - 6*a^
6*c^2*g^2*k*m^3 - 6*a^6*c^2*h*k^3*l^2 + 6*a^5*c^3*h^3*j^2*m + 6*a^6*c^2*g*k
^2*l^3 - 6*a^6*c^2*f*k^3*m^2 - 6*a^5*c^3*h^2*j^3*l - 6*a^5*c^3*g^3*j*m^2 +
6*a^5*c^3*f^2*k^3*m + 3*a^5*b^3*g*k^2*m^3 - 3*a^4*b^4*g^2*k*m^3 + 12*a^6*c^
2*f*j^2*m^3 + 12*a^4*c^4*f^3*j^2*m + 3*a^5*b^3*e*l^2*m^3 + 3*a^3*b^5*e^2*l*
m^3 - 6*a^6*c^2*d*k^2*m^3 - 6*a^5*c^3*f^2*j*l^3 + 6*a^5*c^3*d^2*k*m^3 - 6*a
^5*c^3*g*j^3*k^2 + 6*a^4*c^4*e^3*j*m^2 - 3*b^6*c^2*d^3*j^2*m - 3*a^4*b^4*f*
j^2*m^3 + 3*a^3*b^5*f^2*j*m^3 + 6*a^5*c^3*f*j^2*k^3 + 6*a^5*c^3*f*h^3*m^2 -
 6*a^5*c^3*e*j^3*l^2 + 6*a^4*c^4*g^3*h^2*l - 6*a^4*c^4*f^2*h^3*m + 6*a^4*c^

4*e^2*j^3*l + 6*a^3*c^5*d^3*j^2*m - 3*a^4*b^4*d*k^2*m^3 - 3*a^2*b^6*d^2*k*m
^3 + 6*a^5*c^3*e^2*h*m^3 - 6*a^4*c^4*g^2*h^3*k - 6*a^4*c^4*f^3*h*l^2 + 12*a
^5*c^3*e*h^2*l^3 + 12*a^3*c^5*e^3*h^2*l - 3*b^6*c^2*d^3*h*l^2 + 3*b^5*c^3*d
^3*h^2*l - 3*a^5*b^2*c*j^4*m^2 + 3*a^3*b^5*e*h^2*m^3 - 3*a^2*b^6*e^2*h*m^3
+ 6*a^5*c^3*d*g^2*m^3 - 6*a^4*c^4*e^2*h*k^3 - 6*a^4*c^4*f*h^3*j^2 + 6*a^4*c
^4*e*g^3*l^2 + 6*a^3*c^5*f^3*g^2*k - 6*a^3*c^5*e^2*g^3*l + 6*a^3*c^5*d^3*h*
l^2 - 3*b^6*c^2*d^3*f*m^2 - 3*b^4*c^4*d^3*f^2*m + 6*a^4*c^4*d^2*g*l^3 + 6*a
^4*c^4*e*h^2*j^3 - 6*a^4*c^4*d*h^3*k^2 - 6*a^3*c^5*f^2*g^3*j - 6*a^3*c^5*e^
3*g*k^2 + 6*a^3*c^5*d^3*f*m^2 + 6*a^3*c^5*d^2*h^3*k - 6*a^2*c^6*d^3*f^2*m +
 4*a^5*b^2*c*h^3*m^3 + 3*b^5*c^3*d^3*g*k^2 - 3*b^4*c^4*d^3*g^2*k - 3*a^2*b^
6*d*g^2*m^3 + a^5*b*c^2*j^3*k^3 + 12*a^4*c^4*d*g^2*k^3 + 12*a^2*c^6*d^3*g^2
*k + 6*a^5*b*c^2*h^3*l^3 + 5*a^5*b*c^2*g^3*m^3 - 5*a^4*b^3*c*g^3*m^3 + 3*b^
5*c^3*d^3*e*l^2 + 3*b^3*c^5*d^3*e^2*l - 3*a^5*b^2*c*h^2*l^4 + a^4*b^3*c*h^3
*l^3 + 12*a^5*b^2*c*f^2*m^4 - 6*a^3*c^5*d^2*g*j^3 + 6*a^3*c^5*d*f^3*k^2 + 6
*a^3*b^4*c*f^3*m^3 + 6*a^2*c^6*e^3*f^2*j - 6*a^2*c^6*d^2*f^3*k - 3*b^4*c^4*
d^3*f*j^2 + 3*b^3*c^5*d^3*f^2*j - 3*a^2*b^2*c^4*f^5*m - 7*a^4*b*c^3*e^3*m^3
 - 7*a^2*b^5*c*e^3*m^3 + 6*a^4*b*c^3*g^3*k^3 - 6*a^3*c^5*e*g^3*h^2 - 6*a^2*
c^6*d^3*f*j^2 + 5*a^4*b*c^3*f^3*l^3 + a^4*b*c^3*h^3*j^3 + a^2*b^5*c*f^3*l^3
 + 6*a^3*c^5*d*g^2*h^3 - 6*a^2*c^6*e^2*f^3*h - 3*a^3*b^4*c*e^2*l^4 - 3*a*b^
4*c^3*e^4*l^2 - 7*a^3*b*c^4*d^3*l^3 - 7*a*b^5*c^2*d^3*l^3 + 6*a^3*b*c^4*f^3
*j^3 + 5*a^3*b*c^4*e^3*k^3 + 3*b^3*c^5*d^3*e*h^2 - 3*b^2*c^6*d^3*e^2*h + a*
b^5*c^2*e^3*k^3 + 12*a*b^2*c^5*d^4*k^2 - 6*a^2*c^6*d*f^3*g^2 + 6*a*b^4*c^3*
d^3*k^3 - 3*a^4*b^2*c^2*d*k^5 + a^3*b*c^4*g^3*h^3 + 5*a^2*b*c^5*d^3*j^3 - 5
*a*b^3*c^4*d^3*j^3 - 9*a*c^7*d^2*e^2*f^2 + 6*a^2*b*c^5*e^3*h^3 - 3*a*b^2*c^
5*e^4*h^2 + a^2*b*c^5*f^3*g^3 + a*b^3*c^4*e^3*h^3 + 4*a*b^2*c^5*d^3*h^3 - 3
*a*b^2*c^5*d^2*g^4 - 6*a^7*c*j*l^3*m^2 + 6*a^7*c*h*l^2*m^3 + 6*a^6*c^2*j*k^
4*l + 6*a^6*c^2*h*k^4*m - 6*a^5*c^3*h^4*k*m + 3*a^6*b^2*h*k*m^4 + 3*a^6*b^2
*g*l*m^4 - 3*b^5*c^3*d^4*l*m - 6*a^6*c^2*g*j*l^4 - 6*a^6*c^2*f*k*l^4 - 6*a^
6*c^2*d*l^4*m + 6*a^5*c^3*h*j^4*k + 6*a^5*c^3*g*j^4*l + 6*a^5*c^3*f*j^4*m -
 6*a^4*c^4*g^4*j*l + 6*a^3*c^5*e^4*k*m + 6*a^5*b^3*f*j*m^4 - 6*a^4*c^4*g^4*
h*m + 3*b^7*c*d^3*j*m^2 - 3*a^5*b^3*e*k*m^4 - 3*a^5*b^3*d*l*m^4 + 3*b^4*c^4
*d^4*j*l - 3*a^5*b^3*g*h*m^4 - 6*a^5*c^3*e*j*k^4 + 6*a^2*c^6*d^4*j*l + 3*b^
4*c^4*d^4*h*m + 6*a^6*c^2*e*g*m^4 + 6*a^6*c^2*d*h*m^4 + 6*a^6*b*c*j^3*m^3 -
 6*a^5*c^3*f*h*k^4 + 6*a^4*c^4*g*h^4*j + 6*a^4*c^4*f*h^4*k + 6*a^4*c^4*e*h^
4*l + 6*a^4*c^4*d*h^4*m - 6*a^3*c^5*f^4*h*k - 6*a^3*c^5*f^4*g*l + 6*a^2*c^6
*d^4*h*m + 3*a^5*b*c^2*j^5*m + a^6*b*c*k^3*l^3 + 3*a^4*b^4*e*g*m^4 + 3*a^4*
b^4*d*h*m^4 + 6*b^3*c^5*d^4*g*k - 3*b^3*c^5*d^4*h*j - 3*b^3*c^5*d^4*f*l - 3
*b^3*c^5*d^4*e*m + 3*a*b^7*d^2*g*m^3 + 6*a^5*c^3*d*f*l^4 - 6*a^4*c^4*e*g*j^
4 - 6*a^4*c^4*d*h*j^4 + 6*a^3*c^5*e*g^4*j + 6*a^3*c^5*d*g^4*k - 6*a^2*c^6*e
^4*g*j - 6*a^2*c^6*e^4*f*k - 6*a^2*c^6*d*e^4*m + 3*a^4*b*c^3*h^5*l + 6*a^3*
c^5*f*g^4*h - 3*a^3*b^5*d*e*m^4 + 3*b^2*c^6*d^4*e*j + 3*a^5*b*c^2*g*k^5 + 3
*a^3*b*c^4*g^5*k + 8*a*b^6*c*d^3*m^3 + 3*b^2*c^6*d^4*f*h - 3*a^5*b^2*c*e*l^
5 - 3*a*b^2*c^5*e^5*l - 6*a^3*c^5*d*f*h^4 + 6*a^2*c^6*e*f^4*g + 6*a^2*c^6*d
*f^4*h + 3*a^4*b*c^3*f*j^5 + 3*a^2*b*c^5*f^5*j + 6*a*c^7*d^3*e^2*h - 6*a*c^
7*d^2*e^3*g + 3*a^3*b*c^4*e*h^5 + 6*a*b*c^6*d^3*g^3 + 3*a^2*b*c^5*d*g^5 + a

```
*b*c^6*e^3*f^3 - 9*a^6*c^2*j^2*k^2*l^2 - 9*a^6*c^2*h^2*k^2*m^2 - 9*a^6*c^2*
g^2*l^2*m^2 - 18*a^5*c^3*f^2*j^2*m^2 - 9*a^5*c^3*h^2*j^2*k^2 - 9*a^5*c^3*g^
2*j^2*l^2 - 9*a^5*c^3*f^2*k^2*l^2 - 9*a^5*c^3*e^2*k^2*m^2 - 9*a^5*c^3*d^2*l
^2*m^2 - 9*a^5*c^3*g^2*h^2*m^2 - 9*a^4*c^4*e^2*j^2*k^2 - 9*a^4*c^4*d^2*j^2*
l^2 - 18*a^4*c^4*e^2*h^2*l^2 - 9*a^4*c^4*g^2*h^2*j^2 - 9*a^4*c^4*f^2*h^2*k^
2 - 9*a^4*c^4*f^2*g^2*l^2 - 9*a^4*c^4*e^2*g^2*m^2 - 9*a^4*c^4*d^2*h^2*m^2 -
 18*a^3*c^5*d^2*g^2*k^2 - 9*a^3*c^5*e^2*g^2*j^2 - 9*a^3*c^5*e^2*f^2*k^2 - 9
*a^3*c^5*d^2*h^2*j^2 - 9*a^3*c^5*d^2*f^2*l^2 - 9*a^3*c^5*d^2*e^2*m^2 - 3*a^
4*b^2*c^2*h^4*l^2 - 18*a^4*b^2*c^2*f^3*m^3 + 12*a^3*b^2*c^3*f^4*m^2 - 9*a^3
*c^5*f^2*g^2*h^2 + 4*a^4*b^2*c^2*g^3*l^3 - 3*a^2*b^4*c^2*f^4*m^2 + 14*a^3*b
^3*c^2*e^3*m^3 - 5*a^3*b^3*c^2*f^3*l^3 - 3*a^4*b^2*c^2*g^2*k^4 - 3*a^3*b^2*
c^3*g^4*k^2 + a^3*b^3*c^2*g^3*k^3 - 20*a^2*b^4*c^2*d^3*m^3 - 18*a^3*b^2*c^3
*e^3*l^3 + 16*a^3*b^2*c^3*d^3*m^3 + 12*a^4*b^2*c^2*e^2*l^4 + 12*a^2*b^2*c^4
*e^4*l^2 - 9*a^2*c^6*d^2*e^2*j^2 + 6*a^2*b^4*c^2*e^3*l^3 + 4*a^3*b^2*c^3*f^
3*k^3 + 14*a^2*b^3*c^3*d^3*l^3 - 9*a^2*c^6*e^2*f^2*g^2 - 9*a^2*c^6*d^2*f^2*
h^2 - 5*a^2*b^3*c^3*e^3*k^3 - 3*a^3*b^2*c^3*f^2*j^4 - 3*a^2*b^2*c^4*f^4*j^2
 + a^2*b^3*c^3*f^3*j^3 - 18*a^2*b^2*c^4*d^3*k^3 + 12*a^3*b^2*c^3*d^2*k^4 +
4*a^2*b^2*c^4*e^3*j^3 - 3*a^2*b^4*c^2*d^2*k^4 - 3*a^2*b^2*c^4*e^2*h^4 + 6*a
^7*c*k*l^4*m - 3*a^7*b*k*l*m^4 - 6*a^7*c*h*k*m^4 - 6*a^7*c*g*l*m^4 + 3*a^6*
b*c*h*l^5 - 6*a*c^7*d^4*e*j - 6*a*c^7*d^4*f*h - 3*b*c^7*d^4*e*f + 6*a*c^7*d
*e^4*f + 3*a*b*c^6*e^5*h - a^5*b^2*c*j^3*l^3 - a^3*b^4*c*g^3*l^3 - a*b^4*c^
3*e^3*j^3 - a*b^2*c^5*e^3*g^3 + 3*a^7*b*j*m^5 + 6*a^7*c*f*m^5 + 6*a*c^7*d^5
*k + 3*b*c^7*d^5*g - 3*a^6*c^2*j^4*m^2 - 3*a^6*b^2*j^2*m^4 + 2*a^6*c^2*j^3*
l^3 + a^5*b^3*j^3*m^3 - 2*a^6*c^2*h^3*m^3 - 3*a^6*c^2*h^2*l^4 - 3*a^5*c^3*h
^4*l^2 - a*b^6*c*e^3*l^3 + 20*a^5*c^3*f^3*m^3 - 15*a^6*c^2*f^2*m^4 - 15*a^4
*c^4*f^4*m^2 + 2*a^5*c^3*h^3*k^3 - 2*a^5*c^3*g^3*l^3 + a^3*b^5*g^3*m^3 - 3*
a^5*c^3*g^2*k^4 - 3*a^4*c^4*g^4*k^2 - 3*a^4*b^4*f^2*m^4 + 20*a^4*c^4*e^3*l^
3 - 15*a^5*c^3*e^2*l^4 - 15*a^3*c^5*e^4*l^2 + 2*a^4*c^4*g^3*j^3 - 2*a^4*c^4
*f^3*k^3 - 2*a^4*c^4*d^3*m^3 - 3*b^4*c^4*d^4*k^2 - 3*a^4*c^4*f^2*j^4 - 3*a^
3*c^5*f^4*j^2 + 20*a^3*c^5*d^3*k^3 - 15*a^4*c^4*d^2*k^4 - 15*a^2*c^6*d^4*k^
2 - 2*a^3*c^5*e^3*j^3 + b^5*c^3*d^3*j^3 + 2*a^3*c^5*f^3*h^3 - 3*a^3*c^5*e^2
*h^4 - 3*a^2*c^6*e^4*h^2 - 3*b^2*c^6*d^4*g^2 + 2*a^2*c^6*e^3*g^3 - 2*a^2*c^
6*d^3*h^3 + b^3*c^5*d^3*g^3 - 3*a^2*c^6*d^2*g^4 - a^4*b^2*c^2*h^3*k^3 - a^3
*b^2*c^3*g^3*j^3 - a^2*b^4*c^2*f^3*k^3 - a^2*b^2*c^4*f^3*h^3 + 2*a^7*c*k^3*
m^3 + a^7*b*l^3*m^3 - 3*a^7*c*j^2*m^4 + 6*a^3*c^5*f^5*m - 3*a^6*b^2*f*m^5 +
 6*a^6*c^2*e*l^5 + 6*a^2*c^6*e^5*l + b^7*c*d^3*l^3 + a*b^7*e^3*m^3 - 3*b^2*
c^6*d^5*k + 6*a^5*c^3*d*k^5 - 3*a*c^7*d^4*g^2 + 2*a*c^7*d^3*f^3 + b*c^7*d^3
*e^3 - a^6*b^2*k^3*m^3 - a^4*b^4*h^3*m^3 - a^2*b^6*f^3*m^3 - b^6*c^2*d^3*k^
3 - b^4*c^4*d^3*h^3 - b^2*c^6*d^3*f^3 - b^8*d^3*m^3 - a^6*c^2*k^6 - a^5*c^3
*j^6 - a^4*c^4*h^6 - a^3*c^5*g^6 - a^2*c^6*f^6 - a^7*c*l^6 - a*c^7*e^6 - a^
8*m^6 - c^8*d^6, z, k1), k1, 1, 6) + (k*x)/c + (l*x^2)/(2*c) + (m*x^3)/(3*c
)
```

3.2 $\int \frac{1}{a+bx^n+cx^{2n}}\,dx$

Optimal result

Integrand size = 16, antiderivative size = 124

$$\int \frac{1}{a+bx^n+cx^{2n}}\,dx = -\frac{2cx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}$$

```
[Out] -2*c*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4*a
*c-b*(-4*a*c+b^2)^(1/2))-2*c*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a
*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 124, 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 = {1361, 251}

$$\int \frac{1}{a+bx^n+cx^{2n}}\,dx = -\frac{2cx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2} - \frac{2cx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}$$

```
[In] Int[(a + b*x^n + c*x^(2*n))^(-1),x]
```

```
[Out] (-2*c*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 -
 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (2*c*x*Hypergeometric2F1[1
```

```
, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c + b
*Sqrt[b^2 - 4*a*c])
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 1361

```
Int[((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_))^(-1), x_Symbol] :> With[{q
= Rt[b^2 - 4*a*c, 2]}, Dist[c/q, Int[1/(b/2 - q/2 + c*x^n), x], x] - Dist[c
/q, Int[1/(b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c}, x] && EqQ[n2, 2*
n] && NeQ[b^2 - 4*a*c, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \frac{c\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{c\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{\sqrt{b^2-4ac}}\\ &= -\frac{2cx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}\end{aligned}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 261 vs. $2(124) = 248$.

Time = 0.30 (sec) , antiderivative size = 261, normalized size of antiderivative = 2.10

$$\begin{aligned}&\int \frac{1}{a+bx^n+cx^{2n}}\,dx\\ &= -2cx\left(\frac{1-\left(\frac{x^n}{-\frac{-b+\sqrt{b^2-4ac}}{2c}+x^n}\right)^{-1/n}\text{Hypergeometric2F1}\left(-\frac{1}{n},-\frac{1}{n},\frac{-1+n}{n},\frac{b-\sqrt{b^2-4ac}}{b-\sqrt{b^2-4ac}+2cx^n}\right)}{b^2-4ac-b\sqrt{b^2-4ac}}\right.\\ &\left.+\frac{1-2^{-1/n}\left(\frac{cx^n}{b+\sqrt{b^2-4ac}+2cx^n}\right)^{-1/n}\text{Hypergeometric2F1}\left(-\frac{1}{n},-\frac{1}{n},\frac{-1+n}{n},\frac{b+\sqrt{b^2-4ac}}{b+\sqrt{b^2-4ac}+2cx^n}\right)}{\sqrt{b^2-4ac}\left(b+\sqrt{b^2-4ac}\right)}\right)\end{aligned}$$

```
[In] Integrate[(a + b*x^n + c*x^(2*n))^(-1),x]
```

```
[Out] -2*c*x*((1 - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b - Sqrt[b^2
- 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*
a*c])/c + x^n))^n^(-1))/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) + (1 - Hypergeo
metric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b
^2 - 4*a*c] + 2*c*x^n)]/(2^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n
))^n^(-1)))/(Sqrt[b^2 - 4*a*c]*(b + Sqrt[b^2 - 4*a*c])))
```

Maple [F]

$$\int \frac{1}{a+b\,x^{n}+c\,x^{2n}}dx$$

```
[In] int(1/(a+b*x^n+c*x^(2*n)),x)
[Out] int(1/(a+b*x^n+c*x^(2*n)),x)
```

Fricas [F]

$$\int \frac{1}{a+bx^n+cx^{2n}}\,dx=\int \frac{1}{cx^{2\,n}+bx^n+a}\,dx$$

```
[In] integrate(1/(a+b*x^n+c*x^(2*n)),x, algorithm="fricas")
[Out] integral(1/(c*x^(2*n) + b*x^n + a), x)
```

Sympy [F]

$$\int \frac{1}{a+bx^n+cx^{2n}}\,dx=\int \frac{1}{a+bx^n+cx^{2n}}\,dx$$

```
[In] integrate(1/(a+b*x**n+c*x**(2*n)),x)
[Out] Integral(1/(a + b*x**n + c*x**(2*n)), x)
```

Maxima [F]

$$\int \frac{1}{a+bx^n+cx^{2n}}\,dx=\int \frac{1}{cx^{2\,n}+bx^n+a}\,dx$$

```
[In] integrate(1/(a+b*x^n+c*x^(2*n)),x, algorithm="maxima")
[Out] integrate(1/(c*x^(2*n) + b*x^n + a), x)
```

Giac [F]

$$\int \frac{1}{a + bx^n + cx^{2n}}\,dx = \int \frac{1}{cx^{2n} + bx^n + a}\,dx$$

```
[In] integrate(1/(a+b*x^n+c*x^(2*n)),x, algorithm="giac")
[Out] integrate(1/(c*x^(2*n) + b*x^n + a), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{1}{a + bx^n + cx^{2n}}\,dx = \int \frac{1}{a + b\,x^n + c\,x^{2\,n}}\,dx$$

```
[In] int(1/(a + b*x^n + c*x^(2*n)),x)
[Out] int(1/(a + b*x^n + c*x^(2*n)), x)
```

3.3 $\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx$

Optimal result

Integrand size = 22, antiderivative size = 263

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = -\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} - \frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}$$

```
[Out] -2*c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4
*a*c-b*(-4*a*c+b^2)^(1/2))-c*e*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b
-(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2*c*d*x*hypergeom([1
, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(
1/2))-c*e*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))
/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 263, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.182, Rules used = {1807, 1907, 251, 371}

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = -\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2} - \frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2} - \frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2} - \frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}$$

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

```
[Out] (-2*c*d*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2
 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (2*c*d*x*Hypergeometric2
F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c
 + b*Sqrt[b^2 - 4*a*c]) - (c*e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2
*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (c*
e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c
])])/(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 1807

```
Int[(Pq_)/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[Pq/(b - q + 2*c*x^n), x], x] -
Dist[2*(c/q), Int[Pq/(b + q + 2*c*x^n), x], x]] /; FreeQ[{a, b, c, n}, x] &
```

```
& EqQ[n2, 2*n] && PolyQ[Pq, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1907

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_))^(p_.), x_Symbol] :> Int[ExpandIntegrand[
Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n, p}, x] && (PolyQ[Pq, x] || Poly
Q[Pq, x^n])
```

Rubi steps

$$\text{integral} = \frac{(2c)\int \frac{d+ex}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2c)\int \frac{d+ex}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}}$$

$$= \frac{(2c)\int \left(-\frac{d}{-b+\sqrt{b^2-4ac}-2cx^n} - \frac{ex}{-b+\sqrt{b^2-4ac}-2cx^n}\right)dx}{\sqrt{b^2-4ac}} - \frac{(2c)\int \left(\frac{d}{b+\sqrt{b^2-4ac}+2cx^n} + \frac{ex}{b+\sqrt{b^2-4ac}+2cx^n}\right)dx}{\sqrt{b^2-4ac}}$$

$$= -\frac{(2cd)\int \frac{1}{-b+\sqrt{b^2-4ac}-2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2cd)\int \frac{1}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2ce)\int \frac{x}{-b+\sqrt{b^2-4ac}-2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2ce)\int \frac{x}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}}$$

$$= -\frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}$$

Mathematica [A] (verified)

Time = 1.11 (sec) , antiderivative size = 525, normalized size of antiderivative = 2.00

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx$$

$$= cx\left(-ex\left(\frac{1-\left(\frac{x^n}{-\frac{-b+\sqrt{b^2-4ac}}{2c}+x^n}\right)^{-2/n}\text{Hypergeometric2F1}\left(-\frac{2}{n},-\frac{2}{n},\frac{-2+n}{n},\frac{b-\sqrt{b^2-4ac}}{b-\sqrt{b^2-4ac}+2cx^n}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} + \frac{1-4^{-1/n}\left(\frac{cx^n}{b+\sqrt{b^2-4ac}+2cx^n}\right)^{-2/n}\text{Hypergeometric2F1}\left(-\frac{2}{n},-\frac{2}{n},\frac{-2+n}{n},\frac{b+\sqrt{b^2-4ac}}{b+\sqrt{b^2-4ac}+2cx^n}\right)}{\sqrt{b^2-4ac}\left(b+\sqrt{b^2-4ac}\right)}\right) - 2d\left(\frac{1-\left(\cdots\right.}{}\right.\right.$$

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

```
[Out] c*x*(-(e*x*((1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, (b - Sqrt[b^2 -
4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*
c])/c + x^n))^(2/n))/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) + (1 - Hypergeomet
ric2F1[-2/n, -2/n, (-2 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*
c] + 2*c*x^n)]/(4^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(2/n))
)/(Sqrt[b^2 - 4*a*c]*(b + Sqrt[b^2 - 4*a*c])))) - 2*d*((1 - Hypergeometric2
F1[-n^(-1), -n^(-1), (-1 + n)/n, (b - Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*
a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))/(b^2
 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) + (1 - Hypergeometric2F1[-n^(-1), -n^(-1),
(-1 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(2^n
^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^n^(-1)))/(Sqrt[b^2 - 4*a*
c]*(b + Sqrt[b^2 - 4*a*c]))))
```

Maple [F]

$$\int \frac{ex+d}{a+b\,x^{n}+c\,x^{2n}}dx$$

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

```
[Out] int((e*x+d)/(a+b*x^n+c*x^(2*n)),x)
```

Fricas [F]

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = \int \frac{ex+d}{cx^{2\,n}+bx^n+a}\,dx$$

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

```
[Out] integral((e*x + d)/(c*x^(2*n) + b*x^n + a), x)
```

Sympy [F]

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = \int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx$$

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

```
[Out] Integral((d + e*x)/(a + b*x**n + c*x**(2*n)), x)
```

Maxima [F]

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = \int \frac{ex+d}{cx^{2\,n}+bx^n+a}\,dx$$

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

Giac [F]

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = \int \frac{ex+d}{cx^{2\,n}+bx^n+a}\,dx$$

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

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex}{a+bx^n+cx^{2n}}\,dx = \int \frac{d+e\,x}{a+b\,x^n+c\,x^{2\,n}}\,dx$$

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

3.4 $\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}} dx$

Optimal result

Integrand size = 27, antiderivative size = 404

$$\begin{aligned}\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = & -\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} \\ & -\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} \\ & -\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} \\ & -\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} \\ & -\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac-b\sqrt{b^2-4ac}\right)} \\ & -\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac+b\sqrt{b^2-4ac}\right)}\end{aligned}$$

```
[Out] -2*c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4
*a*c-b*(-4*a*c+b^2)^(1/2))-c*e*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b
-(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2/3*c*f*x^3*hypergeo
m([1, 3/n],[(3+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+
b^2)^(1/2))-2*c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/
2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-c*e*x^2*hypergeom([1, 2/n],[(2+n)/n],
-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-2/3*c*f*x
^3*hypergeom([1, 3/n],[(3+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c
+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.19 (sec) , antiderivative size = 404, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.148, Rules used = {1807, 1907, 251, 371}

$$\begin{aligned}\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = &-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}\end{aligned}$$

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

```
[Out] (-2*c*d*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2
 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (2*c*d*x*Hypergeometric2
F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c
 + b*Sqrt[b^2 - 4*a*c]) - (c*e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2
*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (c*
e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c
])])/(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c]) - (2*c*f*x^3*Hypergeometric2F1[1,
3/n, (3 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(3*(b^2 - 4*a*c - b*Sq
rt[b^2 - 4*a*c])) - (2*c*f*x^3*Hypergeometric2F1[1, 3/n, (3 + n)/n, (-2*c*x
^n)/(b + Sqrt[b^2 - 4*a*c])])/(3*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c]))
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 1807

```
Int[(Pq_)/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[Pq/(b - q + 2*c*x^n), x], x] -
Dist[2*(c/q), Int[Pq/(b + q + 2*c*x^n), x], x]] /; FreeQ[{a, b, c, n}, x] &
& EqQ[n2, 2*n] && PolyQ[Pq, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1907

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_))^(p_.), x_Symbol] :> Int[ExpandIntegrand[
Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n, p}, x] && (PolyQ[Pq, x] || Poly
Q[Pq, x^n])
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{(2c)\int \frac{d+ex+fx^2}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2c)\int \frac{d+ex+fx^2}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} \\
&= \frac{(2c)\int \left(-\frac{d}{-b+\sqrt{b^2-4ac}-2cx^n} - \frac{ex}{-b+\sqrt{b^2-4ac}-2cx^n} - \frac{fx^2}{-b+\sqrt{b^2-4ac}-2cx^n}\right)dx}{\sqrt{b^2-4ac}} \\
&\quad - \frac{(2c)\int \left(\frac{d}{b+\sqrt{b^2-4ac}+2cx^n} + \frac{ex}{b+\sqrt{b^2-4ac}+2cx^n} + \frac{fx^2}{b+\sqrt{b^2-4ac}+2cx^n}\right)dx}{\sqrt{b^2-4ac}} \\
&= -\frac{(2cd)\int \frac{1}{-b+\sqrt{b^2-4ac}-2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2cd)\int \frac{1}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2ce)\int \frac{x}{-b+\sqrt{b^2-4ac}-2cx^n}\,dx}{\sqrt{b^2-4ac}} \\
&\quad - \frac{(2ce)\int \frac{x}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2cf)\int \frac{x^2}{-b+\sqrt{b^2-4ac}-2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2cf)\int \frac{x^2}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} \\
&= -\frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} \\
&\quad - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} \\
&\quad - \frac{2cfx^3\,{}_2F_1\left(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac-b\sqrt{b^2-4ac}\right)} - \frac{2cfx^3\,{}_2F_1\left(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac+b\sqrt{b^2-4ac}\right)}
\end{aligned}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 834 vs. 2(404) = 808.

Time = 0.92 (sec) , antiderivative size = 834, normalized size of antiderivative = 2.06

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx$$
$$= \frac{x\left(2fx^2\left(\left(-b^2+4ac-b\sqrt{b^2-4ac}\right)\left(1-\left(\frac{x^n}{-\frac{-b+\sqrt{b^2-4ac}}{2c}+x^n}\right)^{-3/n}\text{Hypergeometric2F1}\left(-\frac{3}{n},-\frac{3}{n},\frac{-3+n}{n},\frac{}{b-}\right.\right.\right.\right.}{}$$

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

```
[Out] (x*(2*f*x^2*((-b^2 + 4*a*c - b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-3
/n, -3/n, (-3 + n)/n, (b - Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*
x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(3/n)) + (-b^2 + 4*a*c
+ b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-3/n, -3/n, (-3 + n)/n, (b +
Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(8^n^(-1)*((c*x^n)/(b
 + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(3/n)))) + 3*e*x*((-b^2 + 4*a*c - b*Sqrt[b
^2 - 4*a*c])*(1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, (b - Sqrt[b^2 -
 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a
*c])/c + x^n))^(2/n)) + (-b^2 + 4*a*c + b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeom
etric2F1[-2/n, -2/n, (-2 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*
a*c] + 2*c*x^n)]/(4^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(2/n
)))) + 6*d*((-b^2 + 4*a*c - b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-n^
(-1), -n^(-1), (-1 + n)/n, (b - Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] +
 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1)) - (Sqrt[b^
2 - 4*a*c]*(-b + Sqrt[b^2 - 4*a*c])*(2^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*
c] + 2*c*x^n))^n^(-1) - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b
+ Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)]))/(2^n^(-1)*((c*x^n
)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^n^(-1)))))/(12*a*(-b^2 + 4*a*c))
```

Maple [F]

$$\int \frac{f\,x^2+ex+d}{a+b\,x^n+c\,x^{2n}}dx$$

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

```
[Out] int((f*x^2+e*x+d)/(a+b*x^n+c*x^(2*n)),x)
```

Fricas [F]

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = \int \frac{fx^2+ex+d}{cx^{2n}+bx^n+a}\,dx$$

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

Sympy [F]

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = \int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx$$

```
[In] integrate((f*x**2+e*x+d)/(a+b*x**n+c*x**(2*n)),x)
[Out] Integral((d + e*x + f*x**2)/(a + b*x**n + c*x**(2*n)), x)
```

Maxima [F]

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = \int \frac{fx^2+ex+d}{cx^{2n}+bx^n+a}\,dx$$

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

Giac [F]

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = \int \frac{fx^2+ex+d}{cx^{2n}+bx^n+a}\,dx$$

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

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{a+bx^n+cx^{2n}}\,dx = \int \frac{f\,x^2+e\,x+d}{a+b\,x^n+c\,x^{2\,n}}\,dx$$

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

3.5 $\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 545

$$\begin{aligned}\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx = &-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}}\\
&-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac-b\sqrt{b^2-4ac}\right)}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac+b\sqrt{b^2-4ac}\right)}\\
&-\frac{cgx^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{4+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{2\left(b^2-4ac-b\sqrt{b^2-4ac}\right)}\\
&-\frac{cgx^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{4+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{2\left(b^2-4ac+b\sqrt{b^2-4ac}\right)}\end{aligned}$$

```
[Out] -2*c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4
*a*c-b*(-4*a*c+b^2)^(1/2))-c*e*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b
-(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2/3*c*f*x^3*hypergeo
```

```
m([1, 3/n],[(3+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+
b^2)^(1/2))-1/2*c*g*x^4*hypergeom([1, 4/n],[(4+n)/n],-2*c*x^n/(b-(-4*a*c+b^
2)^(1/2)))/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2*c*d*x*hypergeom([1, 1/n],[1+1
/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-c*e*x
^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c
+b*(-4*a*c+b^2)^(1/2))-2/3*c*f*x^3*hypergeom([1, 3/n],[(3+n)/n],-2*c*x^n/(b
+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-1/2*c*g*x^4*hypergeo
m([1, 4/n],[(4+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/(b^2-4*a*c+b*(-4*a*c+
b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.24 (sec) , antiderivative size = 545, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1807, 1907, 251, 371}

$$
\begin{aligned}
\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx = &-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{2cdx\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{-b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{cex^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b\sqrt{b^2-4ac}-4ac+b^2}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}\\
&-\frac{2cfx^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}\\
&-\frac{cgx^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{n+4}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{2\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}\\
&-\frac{cgx^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{n+4}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{2\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}
\end{aligned}
$$

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

```
[Out] (-2*c*d*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2
 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (2*c*d*x*Hypergeometric2
F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c
```

```
 + b*Sqrt[b^2 - 4*a*c]) - (c*e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2
*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c]) - (c*
e*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c
])])/(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c]) - (2*c*f*x^3*Hypergeometric2F1[1,
3/n, (3 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(3*(b^2 - 4*a*c - b*Sq
rt[b^2 - 4*a*c])) - (2*c*f*x^3*Hypergeometric2F1[1, 3/n, (3 + n)/n, (-2*c*x
^n)/(b + Sqrt[b^2 - 4*a*c])])/(3*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])) - (c*
g*x^4*Hypergeometric2F1[1, 4/n, (4 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c
])])/(2*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])) - (c*g*x^4*Hypergeometric2F1[1
, 4/n, (4 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(2*(b^2 - 4*a*c + b*
Sqrt[b^2 - 4*a*c]))
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 1807

```
Int[(Pq_)/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.)), x_Symbol] :> With[
{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[Pq/(b - q + 2*c*x^n), x], x] -
Dist[2*(c/q), Int[Pq/(b + q + 2*c*x^n), x], x]] /; FreeQ[{a, b, c, n}, x] &
& EqQ[n2, 2*n] && PolyQ[Pq, x] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1907

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_))^(p_.), x_Symbol] :> Int[ExpandIntegrand[
Pq*(a + b*x^n)^p, x], x] /; FreeQ[{a, b, n, p}, x] && (PolyQ[Pq, x] || Poly
Q[Pq, x^n])
```

Rubi steps

$$\text{integral} = \frac{(2c)\int \frac{d+ex+fx^2+gx^3}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}} - \frac{(2c)\int \frac{d+ex+fx^2+gx^3}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{\sqrt{b^2-4ac}}$$

$$= \frac{(2c)\int\left(-\frac{d}{-b+\sqrt{b^2-4ac}-2cx^n}-\frac{ex}{-b+\sqrt{b^2-4ac}-2cx^n}-\frac{fx^2}{-b+\sqrt{b^2-4ac}-2cx^n}-\frac{gx^3}{-b+\sqrt{b^2-4ac}-2cx^n}\right)dx}{\sqrt{b^2-4ac}} - \frac{(2c)\int\left(\frac{d}{b+\sqrt{b^2-4ac}+2cx^n}+\frac{ex}{b+\sqrt{b^2-4ac}+2cx^n}+\frac{fx^2}{b+\sqrt{b^2-4ac}+2cx^n}+\frac{gx^3}{b+\sqrt{b^2-4ac}+2cx^n}\right)dx}{\sqrt{b^2-4ac}}$$

$$= -\frac{(2cd)\int\frac{1}{-b+\sqrt{b^2-4ac}-2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2cd)\int\frac{1}{b+\sqrt{b^2-4ac}+2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2ce)\int\frac{x}{-b+\sqrt{b^2-4ac}-2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2ce)\int\frac{x}{b+\sqrt{b^2-4ac}+2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2cf)\int\frac{x^2}{-b+\sqrt{b^2-4ac}-2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2cf)\int\frac{x^2}{b+\sqrt{b^2-4ac}+2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2cg)\int\frac{x^3}{-b+\sqrt{b^2-4ac}-2cx^n}dx}{\sqrt{b^2-4ac}} - \frac{(2cg)\int\frac{x^3}{b+\sqrt{b^2-4ac}+2cx^n}dx}{\sqrt{b^2-4ac}}$$

$$= -\frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{2cdx\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{b^2-4ac-b\sqrt{b^2-4ac}} - \frac{cex^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{b^2-4ac+b\sqrt{b^2-4ac}} - \frac{2cfx^3\,{}_2F_1\left(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac-b\sqrt{b^2-4ac}\right)} - \frac{2cfx^3\,{}_2F_1\left(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3\left(b^2-4ac+b\sqrt{b^2-4ac}\right)} - \frac{cgx^4\,{}_2F_1\left(1,\frac{4}{n};\frac{4+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{2\left(b^2-4ac-b\sqrt{b^2-4ac}\right)} - \frac{cgx^4\,{}_2F_1\left(1,\frac{4}{n};\frac{4+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{2\left(b^2-4ac+b\sqrt{b^2-4ac}\right)}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 1093 vs. 2(545) = 1090.

Time = 1.28 (sec) , antiderivative size = 1093, normalized size of antiderivative = 2.01

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx$$

$$= \frac{x\left(3gx^3\left(\left(-b^2+4ac-b\sqrt{b^2-4ac}\right)\left(1-\left(\frac{x^n}{-\frac{-b+\sqrt{b^2-4ac}}{2c}+x^n}\right)^{-4/n}\text{Hypergeometric2F1}\left(-\frac{4}{n},-\frac{4}{n},\frac{-4+n}{n},\frac{}{b-}\right.\right.\right.\right.}{}$$

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

```
[Out] (x*(3*g*x^3*((-b^2 + 4*a*c - b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-4
/n, -4/n, (-4 + n)/n, (b - Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*
```

```
x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(4/n)) + (-b^2 + 4*a*c
+ b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-4/n, -4/n, (-4 + n)/n, (b +
Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(2^(4/n)*((c*x^n)/(b
+ Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(4/n)))) + 4*f*x^2*((-b^2 + 4*a*c - b*Sqrt[
b^2 - 4*a*c])*(1 - Hypergeometric2F1[-3/n, -3/n, (-3 + n)/n, (b - Sqrt[b^2
- 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*
a*c])/c + x^n))^(3/n)) + (-b^2 + 4*a*c + b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeo
metric2F1[-3/n, -3/n, (-3 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4
*a*c] + 2*c*x^n)]/(8^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(3/
n)))) + 6*e*x*((-b^2 + 4*a*c - b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[
-2/n, -2/n, (-2 + n)/n, (b - Sqrt[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*
c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n)) + (-b^2 + 4*a*
c + b*Sqrt[b^2 - 4*a*c])*(1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, (b
+ Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(4^n^(-1)*((c*x^n)/
(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^(2/n)))) + 12*d*((-b^2 + 4*a*c - b*Sqrt[
b^2 - 4*a*c])*(1 - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b - Sqr
t[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)]/(x^n/(-1/2*(-b + Sqrt[b^
2 - 4*a*c])/c + x^n))^n^(-1)) - (Sqrt[b^2 - 4*a*c]*(-b + Sqrt[b^2 - 4*a*c])
*(2^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^n^(-1) - Hypergeomet
ric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b + Sqrt[b^2 - 4*a*c])/(b + Sqrt[b^2
- 4*a*c] + 2*c*x^n)]))/(2^n^(-1)*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)
)^n^(-1)))))/(24*a*(-b^2 + 4*a*c))
```

Maple [F]

$$\int \frac{g\,x^{3}+f\,x^{2}+ex+d}{a+b\,x^{n}+c\,x^{2n}}dx$$

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

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

Fricas [F]

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx=\int \frac{gx^3+fx^2+ex+d}{cx^{2\,n}+bx^n+a}\,dx$$

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

[Out] `integral((g*x^3 + f*x^2 + e*x + d)/(c*x^(2*n) + b*x^n + a), x)`

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx=\text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(a+b*x**n+c*x**(2*n)),x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx=\int \frac{gx^3+fx^2+ex+d}{cx^{2n}+bx^n+a}\,dx$$

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

Giac [F]

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx=\int \frac{gx^3+fx^2+ex+d}{cx^{2n}+bx^n+a}\,dx$$

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

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{a+bx^n+cx^{2n}}\,dx=\int \frac{g\,x^3+f\,x^2+e\,x+d}{a+b\,x^n+c\,x^{2\,n}}\,dx$$

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

3.6 $\int \frac{1}{(a+bx^n+cx^{2n})^2} dx$

Optimal result

Integrand size = 16, antiderivative size = 283

$$\int \frac{1}{(a+bx^n+cx^{2n})^2} dx = \frac{x(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} - \frac{c\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x \,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} - \frac{c\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x \,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n}$$

```
[Out] x*(b^2-2*a*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))-c*x*hypergeom([1
, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)-b*
(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))
-c*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))*(4*a*c*(1-
2*n)-b^2*(1-n)+b*(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-
4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.26 (sec) , antiderivative size = 283, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.188, Rules used

= {1359, 1436, 251}

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx =$$
$$-\frac{cx\left(-b(1-n)\sqrt{b^2-4ac}+4ac(1-2n)-\left(b^2(1-n)\right)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$-\frac{cx\left(b(1-n)\sqrt{b^2-4ac}+4ac(1-2n)-\left(b^2(1-n)\right)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$+\frac{x(-2ac+b^2+bcx^n)}{an\left(b^2-4ac\right)\left(a+bx^n+cx^{2n}\right)}$$

```
[In] Int[(a + b*x^n + c*x^(2*n))^(-2),x]
```

```
[Out] (x*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))) - (
c*(4*a*c*(1 - 2*n) - b^2*(1 - n) - b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hypergeom
etric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^
2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*(4*a*c*(1 - 2*n) - b
^2*(1 - n) + b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hypergeometric2F1[1, n^(-1), 1
+ n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*
c + b*Sqrt[b^2 - 4*a*c])*n)
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 1359

```
Int[((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(-
x)*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x^(2*n))^(p + 1)/(a*n*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c +
n*(p + 1)*(b^2 - 4*a*c) + b*c*(n*(2*p + 3) + 1)*x^n)*(a + b*x^n + c*x^(2*n)
)^(p + 1), x], x] /; FreeQ[{a, b, c, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*
a*c, 0] && ILtQ[p, -1]
```

Rule 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
```

```
*c] ||  !IGtQ[n/2, 0])
```

Rubi steps

$$\text{integral} = \frac{x(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} - \frac{\int \frac{b^2-2ac-(b^2-4ac)n+bc(1-n)x^n}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n}$$

$$= \frac{x(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{\left(c\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{2a\left(b^2-4ac\right)^{3/2}n} - \frac{\left(c\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{2a\left(b^2-4ac\right)^{3/2}n}$$

$$= \frac{x(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{c\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} - \frac{c\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n}$$

Mathematica [A] (verified)

Time = 2.68 (sec) , antiderivative size = 456, normalized size of antiderivative = 1.61

$$\int \frac{1}{\left(a+bx^n+cx^{2n}\right)^2}\,dx = -\frac{x\left(\frac{4a^2cn-b^2(-1+n)x^n(b+cx^n)+a\left(-b^2n+bc(-3+4n)x^n+2c^2(-1+2n)x^{2n}\right)}{a+x^n(b+cx^n)} + \frac{2^{-1/n}ac\left(4ac\sqrt{b^2-4ac}(1-2n)+b^3(-1+n)-4abc(-1+n)+b\right.}{}\right.}{}$$

```
[In] Integrate[(a + b*x^n + c*x^(2*n))^(-2),x]
```

```
[Out] -((x*((4*a^2*c*n - b^2*(-1 + n)*x^n*(b + c*x^n) + a*(-(b^2*n) + b*c*(-3 + 4
*n)*x^n + 2*c^2*(-1 + 2*n)*x^(2*n))))/(a + x^n*(b + c*x^n)) + (a*c*(4*a*c*Sq
rt[b^2 - 4*a*c]*(1 - 2*n) + b^3*(-1 + n) - 4*a*b*c*(-1 + n) + b^2*Sqrt[b^2
- 4*a*c]*(-1 + n))*Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b - Sqr
t[b^2 - 4*a*c])/(b - Sqrt[b^2 - 4*a*c] + 2*c*x^n)])/(2^n^(-1)*Sqrt[b^2 - 4*
a*c]*(-b^2 + 4*a*c + b*Sqrt[b^2 - 4*a*c])*((c*x^n)/(b - Sqrt[b^2 - 4*a*c] +
 2*c*x^n))^n^(-1)) + (a*c*(-(b^2*(-1 + n)) + b*Sqrt[b^2 - 4*a*c]*(-1 + n) +
```

```
 4*a*c*(-1 + 2*n))*Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, (b + Sqr
t[b^2 - 4*a*c])/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n)])/(2^n^(-1)*Sqrt[b^2 - 4*
a*c]*(b + Sqrt[b^2 - 4*a*c])*((c*x^n)/(b + Sqrt[b^2 - 4*a*c] + 2*c*x^n))^n^
(-1))))/(a^2*(b^2 - 4*a*c)*n))
```

Maple [F]

$$\int \frac{1}{(a+b\,x^n+c\,x^{2n})^2}dx$$

```
[In] int(1/(a+b*x^n+c*x^(2*n))^2,x)
[Out] int(1/(a+b*x^n+c*x^(2*n))^2,x)
```

Fricas [F]

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{1}{(cx^{2\,n}+bx^n+a)^2}\,dx$$

```
[In] integrate(1/(a+b*x^n+c*x^(2*n))^2,x, algorithm="fricas")
[Out] integral(1/(c^2*x^(4*n) + b^2*x^(2*n) + 2*a*b*x^n + a^2 + 2*(b*c*x^n + a*c)
*x^(2*n)), x)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx = \text{Timed out}$$

```
[In] integrate(1/(a+b*x**n+c*x**(2*n))**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{1}{(cx^{2\,n}+bx^n+a)^2}\,dx$$

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

Giac [F]

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{1}{(cx^{2n}+bx^n+a)^2}\,dx$$

```
[In] integrate(1/(a+b*x^n+c*x^(2*n))^2,x, algorithm="giac")
[Out] integrate((c*x^(2*n) + b*x^n + a)^(-2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{1}{(a+b\,x^n+c\,x^{2\,n})^2}\,dx$$

```
[In] int(1/(a + b*x^n + c*x^(2*n))^2,x)
[Out] int(1/(a + b*x^n + c*x^(2*n))^2, x)
```

3.7 $\int \frac{d+ex}{\left(a+bx^n+cx^{2n}\right)^2}\,dx$

Optimal result

Integrand size = 22, antiderivative size = 738

$$
\begin{aligned}
\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx &= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}+\frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} \\
&- \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(2+n)} \\
&+ \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(2+n)}
\end{aligned}
$$

```
[Out] d*x*(b^2-2*a*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+e*x^2*(b^2-2*a
*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))-2*b*c^2*e*(2-n)*x^(2+n)*hy
pergeom([1, (2+n)/n],[2+2/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2
)^(3/2)/n/(2+n)/(b-(-4*a*c+b^2)^(1/2))+2*b*c^2*e*(2-n)*x^(2+n)*hypergeom([1
, (2+n)/n],[2+2/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/
(2+n)/(b+(-4*a*c+b^2)^(1/2))-c*e*(4*a*c*(1-n)-b^2*(2-n))*x^2*hypergeom([1,
```

```
2/n],[(2+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c
-b*(-4*a*c+b^2)^(1/2))-c*e*(4*a*c*(1-n)-b^2*(2-n))*x^2*hypergeom([1, 2/n],[
(2+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-4
*a*c+b^2)^(1/2))-c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^
(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)-b*(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)
/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x
^n/(b+(-4*a*c+b^2)^(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)+b*(1-n)*(-4*a*c+b^2)^(1
/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.87 (sec) , antiderivative size = 738, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.364$, Rules used = {1810, 1359, 1436, 251, 1398, 1574, 1397, 371}

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx$$
$$= -\frac{2bc^2e(2-n)x^{n+2}\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{an(n+2)\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)}$$
$$+\frac{2bc^2e(2-n)x^{n+2}\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{an(n+2)\left(b^2-4ac\right)^{3/2}\left(\sqrt{b^2-4ac}+b\right)}$$
$$-\frac{cdx\left(-b(1-n)\sqrt{b^2-4ac}+4ac(1-2n)-\left(b^2(1-n)\right)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$-\frac{cdx\left(b(1-n)\sqrt{b^2-4ac}+4ac(1-2n)-\left(b^2(1-n)\right)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$+\frac{dx(-2ac+b^2+bcx^n)}{an\left(b^2-4ac\right)\left(a+bx^n+cx^{2n}\right)}$$
$$-\frac{cex^2\left(4ac(1-n)-b^2(2-n)\right)\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(-b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$-\frac{cex^2\left(4ac(1-n)-b^2(2-n)\right)\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{an\left(b^2-4ac\right)\left(b\sqrt{b^2-4ac}-4ac+b^2\right)}$$
$$+\frac{ex^2(-2ac+b^2+bcx^n)}{an\left(b^2-4ac\right)\left(a+bx^n+cx^{2n}\right)}$$

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

```
[Out] (d*x*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))) +
 (e*x^2*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))
```

```
) - (c*d*(4*a*c*(1 - 2*n) - b^2*(1 - n) - b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hy
pergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])
/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*d*(4*a*c*(1 -
 2*n) - b^2*(1 - n) + b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hypergeometric2F1[1, n
^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*(b
^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a*c*(1 - n) - b^2*(2 - n))*x
^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])]
)/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a*c*(1
- n) - b^2*(2 - n))*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b
+ Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])
*n) - (2*b*c^2*e*(2 - n)*x^(2 + n)*Hypergeometric2F1[1, (2 + n)/n, 2*(1 + n
^(-1)), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b - Sq
rt[b^2 - 4*a*c])*n*(2 + n)) + (2*b*c^2*e*(2 - n)*x^(2 + n)*Hypergeometric2F
1[1, (2 + n)/n, 2*(1 + n^(-1)), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^
2 - 4*a*c)^(3/2)*(b + Sqrt[b^2 - 4*a*c])*n*(2 + n))
```

<u>Rule 251</u>

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] && !IGtQ[p
, 0] && !IntegerQ[1/n] && !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

<u>Rule 371</u>

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] && !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

<u>Rule 1359</u>

```
Int[((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(-
x)*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x^(2*n))^(p + 1)/(a*n*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c +
n*(p + 1)*(b^2 - 4*a*c) + b*c*(n*(2*p + 3) + 1)*x^n)*(a + b*x^n + c*x^(2*n)
)^(p + 1), x], x] /; FreeQ[{a, b, c, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*
a*c, 0] && ILtQ[p, -1]
```

<u>Rule 1397</u>

```
Int[((d_.)*(x_))^(m_.)/((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_)), x_Symb
ol] :> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[(d*x)^m/(b - q + 2*
c*x^n), x], x] - Dist[2*(c/q), Int[(d*x)^m/(b + q + 2*c*x^n), x], x]] /; Fr
eeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0]
```

<u>Rule 1398</u>

```
Int[((d_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x
_Symbol] :> Simp[(-(d*x)^(m + 1))*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x
^(2*n))^(p + 1)/(a*d*n*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b
^2 - 4*a*c)), Int[(d*x)^m*(a + b*x^n + c*x^(2*n))^(p + 1)*Simp[b^2*(n*(p +
1) + m + 1) - 2*a*c*(m + 2*n*(p + 1) + 1) + b*c*(2*n*p + 3*n + m + 1)*x^n,
x], x], x] /; FreeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a
*c, 0] && ILtQ[p + 1, 0]
```

Rule 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
*c] ||  !IGtQ[n/2, 0])
```

Rule 1574

```
Int[((f_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_.)*(
(d_) + (e_.)*(x_)^(n_))^(q_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d
+ e*x^n)^q*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c, d, e, f, m,
 n, p, q}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0] && (IGtQ[p, 0] || IGtQ
[q, 0])
```

Rule 1810

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(p_), x_Symbol] :>
Int[ExpandIntegrand[Pq*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c,
 n}, x] && EqQ[n2, 2*n] && PolyQ[Pq, x] && ILtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{d}{\left(a+bx^n+cx^{2n}\right)^2}+\frac{ex}{\left(a+bx^n+cx^{2n}\right)^2}\right)dx \\
&= d\int \frac{1}{\left(a+bx^n+cx^{2n}\right)^2}\,dx+e\int \frac{x}{\left(a+bx^n+cx^{2n}\right)^2}\,dx \\
&= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)}+\frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} \\
&\quad -\frac{d\int \frac{b^2-2ac-\left(b^2-4ac\right)n+bc(1-n)x^n}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n}-\frac{e\int \frac{x\left(-4ac(1-n)+b^2(2-n)+bc(2-n)x^n\right)}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n}
\end{aligned}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} - \frac{e\int\left(-\frac{b^2\left(1-\frac{4ac(-1+n)}{b^2(-2+n)}\right)(-2+n)x}{a+bx^n+cx^{2n}} - \frac{bc(-2+n)x^{1+n}}{a+bx^n+cx^{2n}}\right)dx}{a(b^2-4ac)n} + \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)\right)\int\frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^n}dx}{2a(b^2-4ac)^{3/2}n} - \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)\right)\int\frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^n}dx}{2a(b^2-4ac)^{3/2}n}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} - \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} + \frac{(e(4ac(1-n)-b^2(2-n)))\int\frac{x}{a+bx^n+cx^{2n}}dx}{a(b^2-4ac)n} - \frac{(bce(2-n))\int\frac{x^{1+n}}{a+bx^n+cx^{2n}}dx}{a(b^2-4ac)n}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} - \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} + \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b-\sqrt{b^2-4ac}+2cx^n}dx}{a(b^2-4ac)^{3/2}n} - \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b+\sqrt{b^2-4ac}+2cx^n}dx}{a(b^2-4ac)^{3/2}n} - \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b-\sqrt{b^2-4ac}+2cx^n}dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b+\sqrt{b^2-4ac}+2cx^n}dx}{a(b^2-4ac)^{3/2}n}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}$$
$$+ \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n}$$
$$- \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n}$$
$$+ \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n}$$
$$- \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,{}_2F_1\left(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n}$$
$$- \frac{2bc^2e(2-n)x^{2+n}\,{}_2F_1\left(1,\frac{2+n}{n};2\left(1+\frac{1}{n}\right);-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(2+n)}$$
$$+ \frac{2bc^2e(2-n)x^{2+n}\,{}_2F_1\left(1,\frac{2+n}{n};2\left(1+\frac{1}{n}\right);-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(2+n)}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 4162 vs. 2(738) = 1476.

Time = 6.42 (sec) , antiderivative size = 4162, normalized size of antiderivative = 5.64

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx = \text{Result too large to show}$$

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

```
[Out] (x*(d + e*x)*(-b^2 + 2*a*c - b*c*x^n))/(a*(-b^2 + 4*a*c)*n*(a + b*x^n + c*x
^(2*n))) - (b*c*e*x^(2 + n)*(x^n)^(2/n - (2 + n)/n)*(-(Hypergeometric2F1[-2
/n, -2/n, (-2 + n)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2
 - 4*a*c])/c + x^n))]/(Sqrt[b^2 - 4*a*c]*(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c]
)/c + x^n))^(2/n))) + Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, -1/2*(-b +
Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt[b^2 -
 4*a*c]*(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))))/(2*a*(-b^2 +
 4*a*c)) + (b*c*e*x^(2 + n)*(x^n)^(2/n - (2 + n)/n)*(-(Hypergeometric2F1[-2
/n, -2/n, (-2 + n)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2
 - 4*a*c])/c + x^n))]/(Sqrt[b^2 - 4*a*c]*(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c]
)/c + x^n))^(2/n))) + Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, -1/2*(-b +
Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt[b^2 -
 4*a*c]*(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))))/(a*(-b^2 + 4
```

```
*a*c)*n) + (b^2*e*x^2*((1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n)/n, -1/2*
(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/
(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))/((b*(-b - Sqrt[b^2 - 4*a*c]
))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hypergeometric2F1[-2/n,
 -2/n, (-2 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqrt[b^2 -
4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))/((b
*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2*c))))/(2*a
*(-b^2 + 4*a*c)) - (2*c*e*x^2*((1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n)/
n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n)
)]/(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))/((b*(-b - Sqrt[b^2
- 4*a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hypergeometric2
F1[-2/n, -2/n, (-2 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqr
t[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^(2
/n))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2*c)
)))/(-b^2 + 4*a*c) - (b^2*e*x^2*((1 - Hypergeometric2F1[-2/n, -2/n, (-2 + n
)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^
n))]/(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))/((b*(-b - Sqrt[b^
2 - 4*a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hypergeometri
c2F1[-2/n, -2/n, (-2 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + S
qrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^
(2/n))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2*
c))))/(a*(-b^2 + 4*a*c)*n) + (2*c*e*x^2*((1 - Hypergeometric2F1[-2/n, -2/n,
 (-2 + n)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c]
)/c + x^n))]/(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^(2/n))/((b*(-b -
 Sqrt[b^2 - 4*a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hyper
geometric2F1[-2/n, -2/n, (-2 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2
*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c
+ x^n))^(2/n))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c
])^2/(2*c))))/((-b^2 + 4*a*c)*n) - (b*c*d*x^(1 + n)*(x^n)^(n^(-1) - (1 + n)
/n)*(-(Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b - Sqrt[b^2
- 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt[b^2 - 4*a*c]*(
x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))) + Hypergeometric2F1[-
n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + S
qrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt[b^2 - 4*a*c]*(x^n/(-1/2*(-b + Sqrt[b^2 -
 4*a*c])/c + x^n))^n^(-1))))/(a*(-b^2 + 4*a*c)) + (b*c*d*x^(1 + n)*(x^n)^(n
^(-1) - (1 + n)/n)*(-(Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*
(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt
[b^2 - 4*a*c]*(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))) + Hype
rgeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(
c*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(Sqrt[b^2 - 4*a*c]*(x^n/(-1/2*(
-b + Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))))/(a*(-b^2 + 4*a*c)*n) + (b^2*d*x
*((1 - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b - Sqrt[b^2
- 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b - Sqr
t[b^2 - 4*a*c])/c + x^n))^n^(-1))/((b*(-b - Sqrt[b^2 - 4*a*c]))/(2*c) + (-b
 - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hypergeometric2F1[-n^(-1), -n^(-1), (
```

```
-1 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/
c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))/((b*(-b +
Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2*c))))/(a*(-b^2 +
4*a*c)) - (4*c*d*x*((1 - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1
/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))]/(x
^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))/((b*(-b - Sqrt[b^2 - 4*
a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hypergeometric2F1[-
n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b + S
qrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))^
n^(-1))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2
*c))))/(-b^2 + 4*a*c) - (b^2*d*x*((1 - Hypergeometric2F1[-n^(-1), -n^(-1),
(-1 + n)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b - Sqrt[b^2 - 4*a*c])
/c + x^n))]/(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))/((b*(-b -
 Sqrt[b^2 - 4*a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2/(2*c)) + (1 - Hyper
geometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b + Sqrt[b^2 - 4*a*c])/(c
*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b + Sqrt[b^2 - 4*a*
c])/c + x^n))^n^(-1))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*c) + (-b + Sqrt[b^2
- 4*a*c])^2/(2*c))))/(a*(-b^2 + 4*a*c)*n) + (2*c*d*x*((1 - Hypergeometric2F
1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b - Sqrt[b^2 - 4*a*c])/(c*(-1/2*(-b
- Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(-b - Sqrt[b^2 - 4*a*c])/c + x^n
))^n^(-1))/((b*(-b - Sqrt[b^2 - 4*a*c]))/(2*c) + (-b - Sqrt[b^2 - 4*a*c])^2
/(2*c)) + (1 - Hypergeometric2F1[-n^(-1), -n^(-1), (-1 + n)/n, -1/2*(-b + S
qrt[b^2 - 4*a*c])/(c*(-1/2*(-b + Sqrt[b^2 - 4*a*c])/c + x^n))]/(x^n/(-1/2*(
-b + Sqrt[b^2 - 4*a*c])/c + x^n))^n^(-1))/((b*(-b + Sqrt[b^2 - 4*a*c]))/(2*
c) + (-b + Sqrt[b^2 - 4*a*c])^2/(2*c))))/((-b^2 + 4*a*c)*n)
```

Maple [F]

$$\int \frac{ex+d}{\left(a+b\,x^{n}+c\,x^{2n}\right)^{2}}dx$$

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

```
[Out] int((e*x+d)/(a+b*x^n+c*x^(2*n))^2,x)
```

Fricas [F]

$$\int \frac{d+ex}{\left(a+bx^{n}+cx^{2n}\right)^{2}}\,dx=\int \frac{ex+d}{\left(cx^{2\,n}+bx^{n}+a\right)^{2}}\,dx$$

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

```
[Out] integral((e*x + d)/(c^2*x^(4*n) + b^2*x^(2*n) + 2*a*b*x^n + a^2 + 2*(b*c*x^
n + a*c)*x^(2*n)), x)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx = \text{Timed out}$$

```
[In] integrate((e*x+d)/(a+b*x**n+c*x**(2*n))**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

Giac [F]

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{d+e\,x}{(a+b\,x^n+c\,x^{2\,n})^2}\,dx$$

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

3.8 $\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx$

Optimal result

Integrand size = 27, antiderivative size = 1194

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{fx^3(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})}$$
$$- \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{ce(4ac(1-n)-b^2(2-n))x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{ce(4ac(1-n)-b^2(2-n))x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{2cf(2ac(3-2n)-b^2(3-n))x^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{2cf(2ac(3-2n)-b^2(3-n))x^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n}$$
$$- \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(2+n)}$$
$$+ \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(2+n)}$$
$$- \frac{2bc^2f(3-n)x^{3+n}\,\text{Hypergeometric2F1}\left(1,\frac{3+n}{n},2+\frac{3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(3+n)}$$
$$+ \frac{2bc^2f(3-n)x^{3+n}\,\text{Hypergeometric2F1}\left(1,\frac{3+n}{n},2+\frac{3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(3+n)}$$

```
[Out] d*x*(b^2-2*a*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+e*x^2*(b^2-2*a
*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+f*x^3*(b^2-2*a*c+b*c*x^n)/
a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))-2*b*c^2*e*(2-n)*x^(2+n)*hypergeom([1,
(2+n)/n],[2+2/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(2
+n)/(b-(-4*a*c+b^2)^(1/2))-2*b*c^2*f*(3-n)*x^(3+n)*hypergeom([1, (3+n)/n],[
2+3/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(3+n)/(b-(-4
```

```
*a*c+b^2)^(1/2))+2*b*c^2*e*(2-n)*x^(2+n)*hypergeom([1, (2+n)/n],[2+2/n],-2*
c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(2+n)/(b+(-4*a*c+b^2)^
(1/2))+2*b*c^2*f*(3-n)*x^(3+n)*hypergeom([1, (3+n)/n],[2+3/n],-2*c*x^n/(b+(
-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(3+n)/(b+(-4*a*c+b^2)^(1/2))-c*e
*(4*a*c*(1-n)-b^2*(2-n))*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b-(-4*a
*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2/3*c*f*(
2*a*c*(3-2*n)-b^2*(3-n))*x^3*hypergeom([1, 3/n],[(3+n)/n],-2*c*x^n/(b-(-4*a
*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-c*e*(4*a*
c*(1-n)-b^2*(2-n))*x^2*hypergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b+(-4*a*c+b^2
)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-2/3*c*f*(2*a*c*
(3-2*n)-b^2*(3-n))*x^3*hypergeom([1, 3/n],[(3+n)/n],-2*c*x^n/(b+(-4*a*c+b^2
)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))-c*d*x*hypergeom
([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)
-b*(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/
2))-c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))*(4*a*
c*(1-2*n)-b^2*(1-n)+b*(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c
+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 1.39 (sec) , antiderivative size = 1194, normalized size of antiderivative = 1.00, number of steps used = 24, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.296$, Rules used

= {1810, 1359, 1436, 251, 1398, 1574, 1397, 371}

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx$$

$$= -\frac{2bc^2e(2-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^{n+2}}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(n+2)}$$
$$+\frac{2bc^2e(2-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^{n+2}}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(n+2)}$$
$$-\frac{2bc^2f(3-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+3}{n},2+\frac{3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^{n+3}}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(n+3)}$$
$$+\frac{2bc^2f(3-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+3}{n},2+\frac{3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^{n+3}}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(n+3)}$$
$$-\frac{2cf\left(2ac(3-2n)-b^2(3-n)\right)\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^3}{3a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{2cf\left(2ac(3-2n)-b^2(3-n)\right)\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^3}{3a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{f\left(bcx^n+b^2-2ac\right)x^3}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$
$$-\frac{ce(4ac(1-n)-b^2(2-n))\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^2}{a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{ce(4ac(1-n)-b^2(2-n))\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^2}{a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{e(bcx^n+b^2-2ac)\,x^2}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$
$$-\frac{cd\left(-((1-n)b^2)-\sqrt{b^2-4ac}(1-n)b+4ac(1-2n)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x}{a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{cd\left(-((1-n)b^2)+\sqrt{b^2-4ac}(1-n)b+4ac(1-2n)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x}{a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{d(bcx^n+b^2-2ac)\,x}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$

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

```
[Out] (d*x*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))) +
 (e*x^2*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))
```

```
) + (f*x^3*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*
n))) - (c*d*(4*a*c*(1 - 2*n) - b^2*(1 - n) - b*Sqrt[b^2 - 4*a*c]*(1 - n))*x
*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c]
)])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*d*(4*a*c*(
1 - 2*n) - b^2*(1 - n) + b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hypergeometric2F1[1
, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)
*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a*c*(1 - n) - b^2*(2 - n)
)*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c
])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a*c*
(1 - n) - b^2*(2 - n))*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/
(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*
c])*n) - (2*c*f*(2*a*c*(3 - 2*n) - b^2*(3 - n))*x^3*Hypergeometric2F1[1, 3/
n, (3 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(3*a*(b^2 - 4*a*c)*(b^2
- 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (2*c*f*(2*a*c*(3 - 2*n) - b^2*(3 - n))*
x^3*Hypergeometric2F1[1, 3/n, (3 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])
])/(3*a*(b^2 - 4*a*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (2*b*c^2*e*(
2 - n)*x^(2 + n)*Hypergeometric2F1[1, (2 + n)/n, 2*(1 + n^(-1)), (-2*c*x^n)
/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b - Sqrt[b^2 - 4*a*c])*n
*(2 + n)) + (2*b*c^2*e*(2 - n)*x^(2 + n)*Hypergeometric2F1[1, (2 + n)/n, 2*
(1 + n^(-1)), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(
b + Sqrt[b^2 - 4*a*c])*n*(2 + n)) - (2*b*c^2*f*(3 - n)*x^(3 + n)*Hypergeome
tric2F1[1, (3 + n)/n, 2 + 3/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^2
 - 4*a*c)^(3/2)*(b - Sqrt[b^2 - 4*a*c])*n*(3 + n)) + (2*b*c^2*f*(3 - n)*x^(
3 + n)*Hypergeometric2F1[1, (3 + n)/n, 2 + 3/n, (-2*c*x^n)/(b + Sqrt[b^2 -
4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b + Sqrt[b^2 - 4*a*c])*n*(3 + n))
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 1359

```
Int[((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(-
x)*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x^(2*n))^(p + 1)/(a*n*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c +
n*(p + 1)*(b^2 - 4*a*c) + b*c*(n*(2*p + 3) + 1)*x^n)*(a + b*x^n + c*x^(2*n)
)^(p + 1), x], x] /; FreeQ[{a, b, c, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*
```

```
a*c, 0] && ILtQ[p, -1]
```

Rule 1397

```
Int[((d_.)*(x_))^(m_.)/((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_)), x_Symb
ol] :> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[(d*x)^m/(b - q + 2*
c*x^n), x], x] - Dist[2*(c/q), Int[(d*x)^m/(b + q + 2*c*x^n), x], x]] /; Fr
eeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1398

```
Int[((d_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x
_Symbol] :> Simp[(-(d*x)^(m + 1))*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x
^(2*n))^(p + 1)/(a*d*n*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b
^2 - 4*a*c)), Int[(d*x)^m*(a + b*x^n + c*x^(2*n))^(p + 1)*Simp[b^2*(n*(p +
1) + m + 1) - 2*a*c*(m + 2*n*(p + 1) + 1) + b*c*(2*n*p + 3*n + m + 1)*x^n,
x], x], x] /; FreeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a
*c, 0] && ILtQ[p + 1, 0]
```

Rule 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
*c] ||  !IGtQ[n/2, 0])
```

Rule 1574

```
Int[((f_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_.)*(
(d_) + (e_.)*(x_)^(n_))^(q_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d
+ e*x^n)^q*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c, d, e, f, m,
 n, p, q}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0] && (IGtQ[p, 0] || IGtQ
[q, 0])
```

Rule 1810

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(p_), x_Symbol] :>
Int[ExpandIntegrand[Pq*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c,
 n}, x] && EqQ[n2, 2*n] && PolyQ[Pq, x] && ILtQ[p, -1]
```

Rubi steps

$$\text{integral} = \int \left(\frac{d}{(a+bx^n+cx^{2n})^2}+\frac{ex}{(a+bx^n+cx^{2n})^2}+\frac{fx^2}{(a+bx^n+cx^{2n})^2}\right)\,dx$$

$$= d\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx + e\int \frac{x}{(a+bx^n+cx^{2n})^2}\,dx + f\int \frac{x^2}{(a+bx^n+cx^{2n})^2}\,dx$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{fx^3(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} - \frac{d\int \frac{b^2-2ac-(b^2-4ac)n+bc(1-n)x^n}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n} - \frac{e\int \frac{x\left(-4ac(1-n)+b^2(2-n)+bc(2-n)x^n\right)}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n} - \frac{f\int \frac{x^2\left(-2ac(3-2n)+b^2(3-n)+bc(3-n)x^n\right)}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{fx^3(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} - \frac{e\int \left(-\frac{b^2\left(1-\frac{4ac(-1+n)}{b^2(-2+n)}\right)(-2+n)x}{a+bx^n+cx^{2n}} - \frac{bc(-2+n)x^{1+n}}{a+bx^n+cx^{2n}}\right)dx}{a\left(b^2-4ac\right)n} - \frac{f\int \left(-\frac{b^2(-3+n)\left(1-\frac{2ac(-3+2n)}{b^2(-3+n)}\right)x^2}{a+bx^n+cx^{2n}} - \frac{bc(-3+n)x^{2+n}}{a+bx^n+cx^{2n}}\right)dx}{a\left(b^2-4ac\right)n} + \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)\right)\int \frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{2a\left(b^2-4ac\right)^{3/2}n} - \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)\right)\int \frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^n}\,dx}{2a\left(b^2-4ac\right)^{3/2}n}$$

$$= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{fx^3(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} - \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} + \frac{\left(e\left(4ac(1-n)-b^2(2-n)\right)\right)\int \frac{x}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n} + \frac{\left(f\left(2ac(3-2n)-b^2(3-n)\right)\right)\int \frac{x^2}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n} - \frac{(bce(2-n))\int \frac{x^{1+n}}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n} - \frac{(bcf(3-n))\int \frac{x^{2+n}}{a+bx^n+cx^{2n}}\,dx}{a\left(b^2-4ac\right)n}$$

$$
\begin{aligned}
&= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} \\
&\quad + \frac{fx^3(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} \\
&\quad + \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} \\
&\quad - \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} \\
&\quad + \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&\quad - \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&\quad + \frac{(2cf(2ac(3-2n)-b^2(3-n)))\int\frac{x^2}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&\quad - \frac{(2cf(2ac(3-2n)-b^2(3-n)))\int\frac{x^2}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&\quad - \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&\quad - \frac{(2bc^2f(3-n))\int\frac{x^{2+n}}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2f(3-n))\int\frac{x^{2+n}}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{dx(b^2-2ac+bcx^n)}{a\,(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}+\frac{ex^2(b^2-2ac+bcx^n)}{a\,(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}\\
&\quad+\frac{fx^3(b^2-2ac+bcx^n)}{a\,(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}\\
&\quad+\frac{cd\big(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\big)\,x\,{}_2F_1\Big(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b-\sqrt{b^2-4ac}\big)\,n}\\
&\quad-\frac{cd\big(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\big)\,x\,{}_2F_1\Big(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b+\sqrt{b^2-4ac}\big)\,n}\\
&\quad+\frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,{}_2F_1\Big(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b-\sqrt{b^2-4ac}\big)\,n}\\
&\quad-\frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,{}_2F_1\Big(1,\frac{2}{n};\frac{2+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b+\sqrt{b^2-4ac}\big)\,n}\\
&\quad+\frac{2cf(2ac(3-2n)-b^2(3-n))\,x^3\,{}_2F_1\Big(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\Big)}{3a\,(b^2-4ac)^{3/2}\,\big(b-\sqrt{b^2-4ac}\big)\,n}\\
&\quad-\frac{2cf(2ac(3-2n)-b^2(3-n))\,x^3\,{}_2F_1\Big(1,\frac{3}{n};\frac{3+n}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\Big)}{3a\,(b^2-4ac)^{3/2}\,\big(b+\sqrt{b^2-4ac}\big)\,n}\\
&\quad-\frac{2bc^2e(2-n)x^{2+n}\,{}_2F_1\Big(1,\frac{2+n}{n};2\big(1+\frac{1}{n}\big);-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b-\sqrt{b^2-4ac}\big)\,n(2+n)}\\
&\quad+\frac{2bc^2e(2-n)x^{2+n}\,{}_2F_1\Big(1,\frac{2+n}{n};2\big(1+\frac{1}{n}\big);-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b+\sqrt{b^2-4ac}\big)\,n(2+n)}\\
&\quad-\frac{2bc^2f(3-n)x^{3+n}\,{}_2F_1\Big(1,\frac{3+n}{n};2+\frac{3}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b-\sqrt{b^2-4ac}\big)\,n(3+n)}\\
&\quad+\frac{2bc^2f(3-n)x^{3+n}\,{}_2F_1\Big(1,\frac{3+n}{n};2+\frac{3}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\Big)}{a\,(b^2-4ac)^{3/2}\,\big(b+\sqrt{b^2-4ac}\big)\,n(3+n)}
\end{aligned}
$$

Mathematica [B] (warning: unable to verify)

Leaf count is larger than twice the leaf count of optimal. 6525 vs. 2(1194) = 2388.

Time = 6.55 (sec) , antiderivative size = 6525, normalized size of antiderivative = 5.46

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(d + e*x + f*x^2)/(a + b*x^n + c*x^(2*n))^2,x]
[Out] Result too large to show
```

Maple [F]

$$\int \frac{f\,x^2+ex+d}{(a+b\,x^n+c\,x^{2n})^2}dx$$

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

Fricas [F]

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \text{Timed out}$$

```
[In] integrate((f*x**2+e*x+d)/(a+b*x**n+c*x**(2*n))**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

```
[Out] ((b^2*f - 2*a*c*f)*x^3 + (b^2*e - 2*a*c*e)*x^2 + (b*c*f*x^3 + b*c*e*x^2 + b
*c*d*x)*x^n + (b^2*d - 2*a*c*d)*x)/(a^2*b^2*n - 4*a^3*c*n + (a*b^2*c*n - 4*
a^2*c^2*n)*x^(2*n) + (a*b^3*n - 4*a^2*b*c*n)*x^n) - integrate((2*a*c*d*(2*n
 - 1) - b^2*d*(n - 1) + (2*a*c*f*(2*n - 3) - b^2*f*(n - 3))*x^2 - (b*c*f*(n
 - 3)*x^2 + b*c*e*(n - 2)*x + b*c*d*(n - 1))*x^n + (4*a*c*e*(n - 1) - b^2*e
*(n - 2))*x)/(a^2*b^2*n - 4*a^3*c*n + (a*b^2*c*n - 4*a^2*c^2*n)*x^(2*n) + (
a*b^3*n - 4*a^2*b*c*n)*x^n), x)
```

Giac [F]

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

```
[Out] integrate((f*x^2 + e*x + d)/(c*x^(2*n) + b*x^n + a)^2, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{f\,x^2+e\,x+d}{(a+b\,x^n+c\,x^{2\,n})^2}\,dx$$

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

```
[Out] int((d + e*x + f*x^2)/(a + b*x^n + c*x^(2*n))^2, x)
```

3.9 $\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx$

Optimal result

Integrand size = 32, antiderivative size = 1654

$$
\begin{aligned}
\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx &= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} \\
&+ \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} \\
&+ \frac{fx^3(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} + \frac{gx^4(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})} \\
&- \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{ce(4ac(1-n)-b^2(2-n))\,x^2\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{2+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{2cf(2ac(3-2n)-b^2(3-n))\,x^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{3a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{2cf(2ac(3-2n)-b^2(3-n))\,x^3\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{3+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{3a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{cg(4ac(2-n)-b^2(4-n))\,x^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{4+n}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{2a(b^2-4ac)\left(b^2-4ac-b\sqrt{b^2-4ac}\right)n} \\
&- \frac{cg(4ac(2-n)-b^2(4-n))\,x^4\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{4+n}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{2a(b^2-4ac)\left(b^2-4ac+b\sqrt{b^2-4ac}\right)n} \\
&- \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(2+n)} \\
&+ \frac{2bc^2e(2-n)x^{2+n}\,\text{Hypergeometric2F1}\left(1,\frac{2+n}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(2+n)} \\
&- \frac{2bc^2f(3-n)x^{3+n}\,\text{Hypergeometric2F1}\left(1,\frac{3+n}{n},2+\frac{3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(3+n)} \\
&+ \frac{2bc^2f(3-n)x^{3+n}\,\text{Hypergeometric2F1}\left(1,\frac{3+n}{n},2+\frac{3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(3+n)} \\
&- \frac{2bc^2g(4-n)x^{4+n}\,\text{Hypergeometric2F1}\left(1,\frac{4+n}{n},2\left(1+\frac{2}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(4+n)}
\end{aligned}
$$

$$2bc^2g(4-n)x^{4+n}\,\text{Hypergeometric2F1}\left(1,\frac{4+n}{n},2\left(1+\frac{2}{n}\right),-\frac{2cx^n}{\ldots}\right)$$

```
[Out] d*x*(b^2-2*a*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+e*x^2*(b^2-2*a
*c+b*c*x^n)/a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+f*x^3*(b^2-2*a*c+b*c*x^n)/
a/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+g*x^4*(b^2-2*a*c+b*c*x^n)/a/(-4*a*c+b^
2)/n/(a+b*x^n+c*x^(2*n))-2*b*c^2*e*(2-n)*x^(2+n)*hypergeom([1, (2+n)/n],[2+
2/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(2+n)/(b-(-4*a
*c+b^2)^(1/2))-2*b*c^2*f*(3-n)*x^(3+n)*hypergeom([1, (3+n)/n],[2+3/n],-2*c*
x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(3+n)/(b-(-4*a*c+b^2)^(1
/2))-2*b*c^2*g*(4-n)*x^(4+n)*hypergeom([1, (4+n)/n],[2+4/n],-2*c*x^n/(b-(-4
*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(4+n)/(b-(-4*a*c+b^2)^(1/2))+2*b*c
^2*e*(2-n)*x^(2+n)*hypergeom([1, (2+n)/n],[2+2/n],-2*c*x^n/(b+(-4*a*c+b^2)^
(1/2)))/a/(-4*a*c+b^2)^(3/2)/n/(2+n)/(b+(-4*a*c+b^2)^(1/2))+2*b*c^2*f*(3-n)
*x^(3+n)*hypergeom([1, (3+n)/n],[2+3/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/
(-4*a*c+b^2)^(3/2)/n/(3+n)/(b+(-4*a*c+b^2)^(1/2))+2*b*c^2*g*(4-n)*x^(4+n)*h
ypergeom([1, (4+n)/n],[2+4/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^
2)^(3/2)/n/(4+n)/(b+(-4*a*c+b^2)^(1/2))-c*e*(4*a*c*(1-n)-b^2*(2-n))*x^2*hyp
ergeom([1, 2/n],[(2+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n
/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-2/3*c*f*(2*a*c*(3-2*n)-b^2*(3-n))*x^3*hyp
ergeom([1, 3/n],[(3+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n
/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-1/2*c*g*(4*a*c*(2-n)-b^2*(4-n))*x^4*hyper
geom([1, 4/n],[(4+n)/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(
b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-c*e*(4*a*c*(1-n)-b^2*(2-n))*x^2*hypergeom([
1, 2/n],[(2+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*
a*c+b*(-4*a*c+b^2)^(1/2))-2/3*c*f*(2*a*c*(3-2*n)-b^2*(3-n))*x^3*hypergeom([
1, 3/n],[(3+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*
a*c+b*(-4*a*c+b^2)^(1/2))-1/2*c*g*(4*a*c*(2-n)-b^2*(4-n))*x^4*hypergeom([1,
 4/n],[(4+n)/n],-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))/a/(-4*a*c+b^2)/n/(b^2-4*a*
c+b*(-4*a*c+b^2)^(1/2))-c*d*x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b-(-4*a*
c+b^2)^(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)-b*(1-n)*(-4*a*c+b^2)^(1/2))/a/(-4*a
*c+b^2)/n/(b^2-4*a*c-b*(-4*a*c+b^2)^(1/2))-c*d*x*hypergeom([1, 1/n],[1+1/n]
,-2*c*x^n/(b+(-4*a*c+b^2)^(1/2)))*(4*a*c*(1-2*n)-b^2*(1-n)+b*(1-n)*(-4*a*c+
b^2)^(1/2))/a/(-4*a*c+b^2)/n/(b^2-4*a*c+b*(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 1.99 (sec) , antiderivative size = 1654, normalized size of antiderivative = 1.00, number of steps used = 33, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.250$, Rules used

$= \{1810, 1359, 1436, 251, 1398, 1574, 1397, 371\}$

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx$$

$$= -\frac{2bc^2e(2-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^{n+2}}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(n+2)}$$
$$+\frac{2bc^2e(2-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+2}{n},2\left(1+\frac{1}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^{n+2}}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(n+2)}$$
$$-\frac{2bc^2f(3-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+3}{n},2+\frac{3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^{n+3}}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(n+3)}$$
$$+\frac{2bc^2f(3-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+3}{n},2+\frac{3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^{n+3}}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(n+3)}$$
$$-\frac{2bc^2g(4-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+4}{n},2\left(1+\frac{2}{n}\right),-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^{n+4}}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n(n+4)}$$
$$+\frac{2bc^2g(4-n)\,\text{Hypergeometric2F1}\left(1,\frac{n+4}{n},2\left(1+\frac{2}{n}\right),-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^{n+4}}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n(n+4)}$$
$$-\frac{cg(4ac(2-n)-b^2(4-n))\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{n+4}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^4}{2a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{cg(4ac(2-n)-b^2(4-n))\,\text{Hypergeometric2F1}\left(1,\frac{4}{n},\frac{n+4}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^4}{2a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{g(bcx^n+b^2-2ac)\,x^4}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$
$$-\frac{2cf(2ac(3-2n)-b^2(3-n))\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^3}{3a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{2cf(2ac(3-2n)-b^2(3-n))\,\text{Hypergeometric2F1}\left(1,\frac{3}{n},\frac{n+3}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^3}{3a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{f(bcx^n+b^2-2ac)\,x^3}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$
$$-\frac{ce(4ac(1-n)-b^2(2-n))\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x^2}{a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$
$$-\frac{ce(4ac(1-n)-b^2(2-n))\,\text{Hypergeometric2F1}\left(1,\frac{2}{n},\frac{n+2}{n},-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)x^2}{a\left(b^2-4ac\right)\left(b^2+\sqrt{b^2-4ac}b-4ac\right)n}$$
$$+\frac{e(bcx^n+b^2-2ac)\,x^2}{a\left(b^2-4ac\right)n\left(bx^n+cx^{2n}+a\right)}$$
$$-\frac{cd\left(-((1-n)b^2)-\sqrt{b^2-4ac}(1-n)b+4ac(1-2n)\right)\text{Hypergeometric2F1}\left(1,\frac{1}{n},1+\frac{1}{n},-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)x}{a\left(b^2-4ac\right)\left(b^2-\sqrt{b^2-4ac}b-4ac\right)n}$$

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

```
[Out] (d*x*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))) +
 (e*x^2*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n))
) + (f*x^3*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*
n))) + (g*x^4*(b^2 - 2*a*c + b*c*x^n))/(a*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^
(2*n))) - (c*d*(4*a*c*(1 - 2*n) - b^2*(1 - n) - b*Sqrt[b^2 - 4*a*c]*(1 - n)
)*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a
*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*d*(4*a*
c*(1 - 2*n) - b^2*(1 - n) + b*Sqrt[b^2 - 4*a*c]*(1 - n))*x*Hypergeometric2F
1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a
*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a*c*(1 - n) - b^2*(2 -
 n))*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*
a*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (c*e*(4*a
*c*(1 - n) - b^2*(2 - n))*x^2*Hypergeometric2F1[1, 2/n, (2 + n)/n, (-2*c*x^
n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4
*a*c])*n) - (2*c*f*(2*a*c*(3 - 2*n) - b^2*(3 - n))*x^3*Hypergeometric2F1[1,
 3/n, (3 + n)/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(3*a*(b^2 - 4*a*c)*(b
^2 - 4*a*c - b*Sqrt[b^2 - 4*a*c])*n) - (2*c*f*(2*a*c*(3 - 2*n) - b^2*(3 - n
))*x^3*Hypergeometric2F1[1, 3/n, (3 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*
c])])/(3*a*(b^2 - 4*a*c)*(b^2 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (c*g*(4*a
*c*(2 - n) - b^2*(4 - n))*x^4*Hypergeometric2F1[1, 4/n, (4 + n)/n, (-2*c*x^
n)/(b - Sqrt[b^2 - 4*a*c])])/(2*a*(b^2 - 4*a*c)*(b^2 - 4*a*c - b*Sqrt[b^2 -
 4*a*c])*n) - (c*g*(4*a*c*(2 - n) - b^2*(4 - n))*x^4*Hypergeometric2F1[1, 4
/n, (4 + n)/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(2*a*(b^2 - 4*a*c)*(b^2
 - 4*a*c + b*Sqrt[b^2 - 4*a*c])*n) - (2*b*c^2*e*(2 - n)*x^(2 + n)*Hypergeom
etric2F1[1, (2 + n)/n, 2*(1 + n^(-1)), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])
/(a*(b^2 - 4*a*c)^(3/2)*(b - Sqrt[b^2 - 4*a*c])*n*(2 + n)) + (2*b*c^2*e*(2
- n)*x^(2 + n)*Hypergeometric2F1[1, (2 + n)/n, 2*(1 + n^(-1)), (-2*c*x^n)/(
b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b + Sqrt[b^2 - 4*a*c])*n*(
2 + n)) - (2*b*c^2*f*(3 - n)*x^(3 + n)*Hypergeometric2F1[1, (3 + n)/n, 2 +
3/n, (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b - Sqrt[
b^2 - 4*a*c])*n*(3 + n)) + (2*b*c^2*f*(3 - n)*x^(3 + n)*Hypergeometric2F1[1
, (3 + n)/n, 2 + 3/n, (-2*c*x^n)/(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)
^(3/2)*(b + Sqrt[b^2 - 4*a*c])*n*(3 + n)) - (2*b*c^2*g*(4 - n)*x^(4 + n)*Hy
pergeometric2F1[1, (4 + n)/n, 2*(1 + 2/n), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c
])])/(a*(b^2 - 4*a*c)^(3/2)*(b - Sqrt[b^2 - 4*a*c])*n*(4 + n)) + (2*b*c^2*g
*(4 - n)*x^(4 + n)*Hypergeometric2F1[1, (4 + n)/n, 2*(1 + 2/n), (-2*c*x^n)/
(b + Sqrt[b^2 - 4*a*c])])/(a*(b^2 - 4*a*c)^(3/2)*(b + Sqrt[b^2 - 4*a*c])*n*
(4 + n))
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 1359

```
Int[((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(-
x)*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x^(2*n))^(p + 1)/(a*n*(p + 1)*(b
^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b^2 - 4*a*c)), Int[(b^2 - 2*a*c +
n*(p + 1)*(b^2 - 4*a*c) + b*c*(n*(2*p + 3) + 1)*x^n)*(a + b*x^n + c*x^(2*n)
)^(p + 1), x], x] /; FreeQ[{a, b, c, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*
a*c, 0] && ILtQ[p, -1]
```

Rule 1397

```
Int[((d_.)*(x_))^(m_.)/((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_)), x_Symb
ol] :> With[{q = Rt[b^2 - 4*a*c, 2]}, Dist[2*(c/q), Int[(d*x)^m/(b - q + 2*
c*x^n), x], x] - Dist[2*(c/q), Int[(d*x)^m/(b + q + 2*c*x^n), x], x]] /; Fr
eeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0]
```

Rule 1398

```
Int[((d_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_), x
_Symbol] :> Simp[(-(d*x)^(m + 1))*(b^2 - 2*a*c + b*c*x^n)*((a + b*x^n + c*x
^(2*n))^(p + 1)/(a*d*n*(p + 1)*(b^2 - 4*a*c))), x] + Dist[1/(a*n*(p + 1)*(b
^2 - 4*a*c)), Int[(d*x)^m*(a + b*x^n + c*x^(2*n))^(p + 1)*Simp[b^2*(n*(p +
1) + m + 1) - 2*a*c*(m + 2*n*(p + 1) + 1) + b*c*(2*n*p + 3*n + m + 1)*x^n,
x], x], x] /; FreeQ[{a, b, c, d, m, n}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a
*c, 0] && ILtQ[p + 1, 0]
```

Rule 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
*c] ||  !IGtQ[n/2, 0])
```

Rule 1574

```
Int[((f_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^(n2_.) + (b_.)*(x_)^(n_))^(p_.)*(
(d_) + (e_.)*(x_)^(n_))^(q_.), x_Symbol] :> Int[ExpandIntegrand[(f*x)^m*(d
```

```
+ e*x^n)^q*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c, d, e, f, m,
 n, p, q}, x] && EqQ[n2, 2*n] && NeQ[b^2 - 4*a*c, 0] && (IGtQ[p, 0] || IGtQ
[q, 0])
```

Rule 1810

```
Int[(Pq_)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(p_), x_Symbol] :>
Int[ExpandIntegrand[Pq*(a + b*x^n + c*x^(2*n))^p, x], x] /; FreeQ[{a, b, c,
 n}, x] && EqQ[n2, 2*n] && PolyQ[Pq, x] && ILtQ[p, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \left(\frac{d}{(a+bx^n+cx^{2n})^2}+\frac{ex}{(a+bx^n+cx^{2n})^2}+\frac{fx^2}{(a+bx^n+cx^{2n})^2}+\frac{gx^3}{(a+bx^n+cx^{2n})^2}\right)dx\\
&= d\int \frac{1}{(a+bx^n+cx^{2n})^2}\,dx+e\int \frac{x}{(a+bx^n+cx^{2n})^2}\,dx+f\int \frac{x^2}{(a+bx^n+cx^{2n})^2}\,dx+g\int \frac{x^3}{(a+bx^n+cx^{2n})^2}\,dx\\
&= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}+\frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}\\
&\quad+\frac{fx^3(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}+\frac{gx^4(b^2-2ac+bcx^n)}{a(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}\\
&\quad-\frac{d\int \frac{b^2-2ac-(b^2-4ac)n+bc(1-n)x^n}{a+bx^n+cx^{2n}}\,dx}{a(b^2-4ac)\,n}-\frac{e\int \frac{x(-4ac(1-n)+b^2(2-n)+bc(2-n)x^n)}{a+bx^n+cx^{2n}}\,dx}{a(b^2-4ac)\,n}\\
&\quad-\frac{f\int \frac{x^2(-2ac(3-2n)+b^2(3-n)+bc(3-n)x^n)}{a+bx^n+cx^{2n}}\,dx}{a(b^2-4ac)\,n}-\frac{g\int \frac{x^3(-4ac(2-n)+b^2(4-n)+bc(4-n)x^n)}{a+bx^n+cx^{2n}}\,dx}{a(b^2-4ac)\,n}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} \\
&\quad + \frac{fx^3(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{gx^4(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} \\
&\quad - \frac{e\int\left(-\frac{b^2\left(1-\frac{4ac(-1+n)}{b^2(-2+n)}\right)(-2+n)x}{a+bx^n+cx^{2n}} - \frac{bc(-2+n)x^{1+n}}{a+bx^n+cx^{2n}}\right)dx}{a\left(b^2-4ac\right)n} \\
&\quad - \frac{f\int\left(-\frac{b^2(-3+n)\left(1-\frac{2ac(-3+2n)}{b^2(-3+n)}\right)x^2}{a+bx^n+cx^{2n}} - \frac{bc(-3+n)x^{2+n}}{a+bx^n+cx^{2n}}\right)dx}{a\left(b^2-4ac\right)n} \\
&\quad - \frac{g\int\left(-\frac{b^2\left(1-\frac{4ac(-2+n)}{b^2(-4+n)}\right)(-4+n)x^3}{a+bx^n+cx^{2n}} - \frac{bc(-4+n)x^{3+n}}{a+bx^n+cx^{2n}}\right)dx}{a\left(b^2-4ac\right)n} \\
&\quad + \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)\right)\int\frac{1}{\frac{b}{2}-\frac{1}{2}\sqrt{b^2-4ac}+cx^n}dx}{2a\left(b^2-4ac\right)^{3/2}n} \\
&\quad - \frac{\left(cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)\right)\int\frac{1}{\frac{b}{2}+\frac{1}{2}\sqrt{b^2-4ac}+cx^n}dx}{2a\left(b^2-4ac\right)^{3/2}n}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{dx(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{ex^2(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} \\
&\quad + \frac{fx^3(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{gx^4(b^2-2ac+bcx^n)}{a\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} \\
&\quad + \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} \\
&\quad - \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a\left(b^2-4ac\right)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} \\
&\quad + \frac{\left(e\left(4ac(1-n)-b^2(2-n)\right)\right)\int\frac{x}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n} \\
&\quad + \frac{\left(f\left(2ac(3-2n)-b^2(3-n)\right)\right)\int\frac{x^2}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n} \\
&\quad + \frac{\left(g\left(4ac(2-n)-b^2(4-n)\right)\right)\int\frac{x^3}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n} - \frac{(bce(2-n))\int\frac{x^{1+n}}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n} \\
&\quad - \frac{(bcf(3-n))\int\frac{x^{2+n}}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n} - \frac{(bcg(4-n))\int\frac{x^{3+n}}{a+bx^n+cx^{2n}}dx}{a\left(b^2-4ac\right)n}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{dx(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{ex^2(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} \\
&+ \frac{fx^3(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} + \frac{gx^4(b^2-2ac+bcx^n)}{a(b^2-4ac)n(a+bx^n+cx^{2n})} \\
&+ \frac{cd\left(4ac(1-2n)-b^2(1-n)-b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b-\sqrt{b^2-4ac}\right)n} \\
&- \frac{cd\left(4ac(1-2n)-b^2(1-n)+b\sqrt{b^2-4ac}(1-n)\right)x\,{}_2F_1\left(1,\frac{1}{n};1+\frac{1}{n};-\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)}{a(b^2-4ac)^{3/2}\left(b+\sqrt{b^2-4ac}\right)n} \\
&+ \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2ce(4ac(1-n)-b^2(2-n)))\int\frac{x}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&+ \frac{(2cf(2ac(3-2n)-b^2(3-n)))\int\frac{x^2}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2cf(2ac(3-2n)-b^2(3-n)))\int\frac{x^2}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&+ \frac{(2cg(4ac(2-n)-b^2(4-n)))\int\frac{x^3}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2cg(4ac(2-n)-b^2(4-n)))\int\frac{x^3}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2e(2-n))\int\frac{x^{1+n}}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2bc^2f(3-n))\int\frac{x^{2+n}}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2f(3-n))\int\frac{x^{2+n}}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} \\
&- \frac{(2bc^2g(4-n))\int\frac{x^{3+n}}{b-\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n} + \frac{(2bc^2g(4-n))\int\frac{x^{3+n}}{b+\sqrt{b^2-4ac}+2cx^n}\,dx}{a(b^2-4ac)^{3/2}n}
\end{aligned}
$$

$= \text{Too large to display}$

Mathematica [B] (warning: unable to verify)

Leaf count is larger than twice the leaf count of optimal. 8737 vs. $2(1654) = 3308$.

Time = 6.73 (sec) , antiderivative size = 8737, normalized size of antiderivative = 5.28

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(d + e*x + f*x^2 + g*x^3)/(a + b*x^n + c*x^(2*n))^2,x]
[Out] Result too large to show
```

Maple [F]

$$\int \frac{g\,x^3+f\,x^2+ex+d}{(a+b\,x^n+c\,x^{2n})^2}dx$$

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

Fricas [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

Sympy [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \text{Timed out}$$

```
[In] integrate((g*x**3+f*x**2+e*x+d)/(a+b*x**n+c*x**(2*n))**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

```
[Out] ((b^2*g - 2*a*c*g)*x^4 + (b^2*f - 2*a*c*f)*x^3 + (b^2*e - 2*a*c*e)*x^2 + (b
*c*g*x^4 + b*c*f*x^3 + b*c*e*x^2 + b*c*d*x)*x^n + (b^2*d - 2*a*c*d)*x)/(a^2
*b^2*n - 4*a^3*c*n + (a*b^2*c*n - 4*a^2*c^2*n)*x^(2*n) + (a*b^3*n - 4*a^2*b
*c*n)*x^n) - integrate((2*a*c*d*(2*n - 1) - b^2*d*(n - 1) + (4*a*c*g*(n - 2
) - b^2*g*(n - 4))*x^3 + (2*a*c*f*(2*n - 3) - b^2*f*(n - 3))*x^2 - (b*c*g*(
n - 4)*x^3 + b*c*f*(n - 3)*x^2 + b*c*e*(n - 2)*x + b*c*d*(n - 1))*x^n + (4*
a*c*e*(n - 1) - b^2*e*(n - 2))*x)/(a^2*b^2*n - 4*a^3*c*n + (a*b^2*c*n - 4*a
^2*c^2*n)*x^(2*n) + (a*b^3*n - 4*a^2*b*c*n)*x^n), x)
```

Giac [F]

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{gx^3+fx^2+ex+d}{(cx^{2n}+bx^n+a)^2}\,dx$$

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

```
[Out] integrate((g*x^3 + f*x^2 + e*x + d)/(c*x^(2*n) + b*x^n + a)^2, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{d+ex+fx^2+gx^3}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{g\,x^3+f\,x^2+e\,x+d}{(a+b\,x^n+c\,x^{2\,n})^2}\,dx$$

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

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

3.10 $\int \frac{-ahx^{-1+\frac{n}{2}}+cfx^{-1+n}+cgx^{-1+2n}+chx^{-1+\frac{5n}{2}}}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx$

Optimal result

Integrand size = 63, antiderivative size = 75

$$\int \frac{-ahx^{-1+\frac{n}{2}}+cfx^{-1+n}+cgx^{-1+2n}+chx^{-1+\frac{5n}{2}}}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(c(bf-2ag)+\left(b^2-4ac\right)hx^{n/2}+c(2cf-bg)x^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

```
[Out] -2*(c*(-2*a*g+b*f)+(-4*a*c+b^2)*h*x^(1/2*n)+c*(-b*g+2*c*f)*x^n)/(-4*a*c+b^2
)/n/(a+b*x^n+c*x^(2*n))^(1/2)
```

Rubi [A] (verified)

Time = 0.34 (sec) , antiderivative size = 75, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.032, Rules used = {6873, 1767}

$$\int \frac{-ahx^{-1+\frac{n}{2}}+cfx^{-1+n}+cgx^{-1+2n}+chx^{-1+\frac{5n}{2}}}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(hx^{n/2}\left(b^2-4ac\right)+c(bf-2ag)+cx^n(2cf-bg)\right)}{n\left(b^2-4ac\right)\sqrt{a+bx^n+cx^{2n}}}$$

```
[In] Int[(-(a*h*x^(-1 + n/2)) + c*f*x^(-1 + n) + c*g*x^(-1 + 2*n) + c*h*x^(-1 +
(5*n)/2))/(a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] (-2*(c*(b*f - 2*a*g) + (b^2 - 4*a*c)*h*x^(n/2) + c*(2*c*f - b*g)*x^n))/((b^
2 - 4*a*c)*n*Sqrt[a + b*x^n + c*x^(2*n)])
```

Rule 1767

```
Int[((x_)^(m_.)*((e_) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r_.) + (h_.)*(x_)^(s
_.)))/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(3/2), x_Symbol] :> Sim
p[-(2*c*(b*f - 2*a*g) + 2*h*(b^2 - 4*a*c)*x^(n/2) + 2*c*(2*c*f - b*g)*x^n)/
(c*n*(b^2 - 4*a*c)*Sqrt[a + b*x^n + c*x^(2*n)]), x] /; FreeQ[{a, b, c, e, f
, g, h, m, n}, x] && EqQ[n2, 2*n] && EqQ[q, n/2] && EqQ[r, 3*(n/2)] && EqQ[
s, 2*n] && NeQ[b^2 - 4*a*c, 0] && EqQ[2*m - n + 2, 0] && EqQ[c*e + a*h, 0]
```

Rule 6873

```
Int[u_, x_Symbol] :> With[{v = NormalizeIntegrand[u, x]}, Int[v, x] /; v =!
= u]
```

Rubi steps

$$\text{integral} = \int \frac{x^{-1+\frac{n}{2}}\left(-ah + cfx^{n/2} + cgx^{3n/2} + chx^{2n}\right)}{\left(a + bx^n + cx^{2n}\right)^{3/2}}\,dx$$

$$= -\frac{2\left(c(bf - 2ag) + \left(b^2 - 4ac\right)hx^{n/2} + c(2cf - bg)x^n\right)}{\left(b^2 - 4ac\right)n\sqrt{a + bx^n + cx^{2n}}}$$

Mathematica [A] (verified)

Time = 2.15 (sec) , antiderivative size = 84, normalized size of antiderivative = 1.12

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{\left(a + bx^n + cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(bcf - 2acg + b^2hx^{n/2} - 4achx^{n/2} + 2c^2fx^n - bcgx^n\right)}{\left(b^2 - 4ac\right)n\sqrt{a + bx^n + cx^{2n}}}$$

```
[In] Integrate[(-(a*h*x^(-1 + n/2)) + c*f*x^(-1 + n) + c*g*x^(-1 + 2*n) + c*h*x^
(-1 + (5*n)/2))/(a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] (-2*(b*c*f - 2*a*c*g + b^2*h*x^(n/2) - 4*a*c*h*x^(n/2) + 2*c^2*f*x^n - b*c*
g*x^n))/((b^2 - 4*a*c)*n*Sqrt[a + b*x^n + c*x^(2*n)])
```

Maple [F]

$$\int \frac{-ah\,x^{-1+\frac{n}{2}} + cf\,x^{-1+n} + cg\,x^{-1+2n} + ch\,x^{-1+\frac{5n}{2}}}{\left(a + b\,x^n + c\,x^{2n}\right)^{\frac{3}{2}}}dx$$

```
[In] int((-a*h*x^(-1+1/2*n)+c*f*x^(-1+n)+c*g*x^(-1+2*n)+c*h*x^(-1+5/2*n))/(a+b*x
^n+c*x^(2*n))^(3/2),x)
```

```
[Out] int((-a*h*x^(-1+1/2*n)+c*f*x^(-1+n)+c*g*x^(-1+2*n)+c*h*x^(-1+5/2*n))/(a+b*x
^n+c*x^(2*n))^(3/2),x)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 137, normalized size of antiderivative = 1.83

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{(a+bx^n+cx^{2n})^{3/2}}\,dx = -\frac{2\sqrt{cx^4x^{2n-4}+bx^2x^{n-2}+a}\left((2c^2f-bcg)x^2x^{n-2}+(b^2-4ac)hxx^{\frac{1}{2}n-1}+bcf-2acg\right)}{(b^2c-4ac^2)nx^4x^{2n-4}+(b^3-4abc)nx^2x^{n-2}+(ab^2-4a^2c)n}$$

```
[In] integrate((-a*h*x^(-1+1/2*n)+c*f*x^(-1+n)+c*g*x^(-1+2*n)+c*h*x^(-1+5/2*n))/
(a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="fricas")
```

```
[Out] -2*sqrt(c*x^4*x^(2*n - 4) + b*x^2*x^(n - 2) + a)*((2*c^2*f - b*c*g)*x^2*x^(
n - 2) + (b^2 - 4*a*c)*h*x*x^(1/2*n - 1) + b*c*f - 2*a*c*g)/((b^2*c - 4*a*c
^2)*n*x^4*x^(2*n - 4) + (b^3 - 4*a*b*c)*n*x^2*x^(n - 2) + (a*b^2 - 4*a^2*c)
*n)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{(a+bx^n+cx^{2n})^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((-a*h*x**(-1+1/2*n)+c*f*x**(-1+n)+c*g*x**(-1+2*n)+c*h*x**(-1+5/2*
n))/(a+b*x**n+c*x**(2*n))**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{(a+bx^n+cx^{2n})^{3/2}}\,dx = \int \frac{chx^{\frac{5}{2}n-1}+cgx^{2n-1}+cfx^{n-1}-ahx^{\frac{1}{2}n-1}}{(cx^{2n}+bx^n+a)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((-a*h*x^(-1+1/2*n)+c*f*x^(-1+n)+c*g*x^(-1+2*n)+c*h*x^(-1+5/2*n))/
(a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((c*h*x^(5/2*n - 1) + c*g*x^(2*n - 1) + c*f*x^(n - 1) - a*h*x^(1/2
*n - 1))/(c*x^(2*n) + b*x^n + a)^(3/2), x)
```

Giac [F]

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{(a + bx^n + cx^{2n})^{3/2}} \, dx = \int \frac{chx^{\frac{5}{2}n-1} + cgx^{2n-1} + cfx^{n-1} - ahx^{\frac{1}{2}n-1}}{(cx^{2n} + bx^n + a)^{\frac{3}{2}}} \, dx$$

```
[In] integrate((-a*h*x^(-1+1/2*n)+c*f*x^(-1+n)+c*g*x^(-1+2*n)+c*h*x^(-1+5/2*n))/
(a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="giac")
```

```
[Out] integrate((c*h*x^(5/2*n - 1) + c*g*x^(2*n - 1) + c*f*x^(n - 1) - a*h*x^(1/2
*n - 1))/(c*x^(2*n) + b*x^n + a)^(3/2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{-ahx^{-1+\frac{n}{2}} + cfx^{-1+n} + cgx^{-1+2n} + chx^{-1+\frac{5n}{2}}}{(a + bx^n + cx^{2n})^{3/2}} \, dx = \int \frac{c\,g\,x^{2n-1} - a\,h\,x^{\frac{n}{2}-1} + c\,h\,x^{\frac{5n}{2}-1} + c\,f\,x^{n-1}}{(a + b\,x^n + c\,x^{2n})^{3/2}} \, dx$$

```
[In] int((c*g*x^(2*n - 1) - a*h*x^(n/2 - 1) + c*h*x^((5*n)/2 - 1) + c*f*x^(n - 1
))/(a + b*x^n + c*x^(2*n))^(3/2),x)
```

```
[Out] int((c*g*x^(2*n - 1) - a*h*x^(n/2 - 1) + c*h*x^((5*n)/2 - 1) + c*f*x^(n - 1
))/(a + b*x^n + c*x^(2*n))^(3/2), x)
```

3.11 $\int (a+bx^n+cx^{2n})^p (a+b(1+n+np)x^n+c(1+2n(1$

Optimal result

Integrand size = 45, antiderivative size = 20

$$\int (a+bx^n+cx^{2n})^p (a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n})\,dx = x(a+bx^n+cx^{2n})^{1+p}$$

```
[Out] x*(a+b*x^n+c*x^(2*n))^(p+1)
```

Rubi [A] (verified)

Time = 0.02 (sec) , antiderivative size = 20, normalized size of antiderivative = 1.00, number of steps used = 1, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}} = 0.022$, Rules used = {1789}

$$\int (a+bx^n+cx^{2n})^p (a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n})\,dx = x(a+bx^n+cx^{2n})^{p+1}$$

```
[In] Int[(a + b*x^n + c*x^(2*n))^p*(a + b*(1 + n + n*p)*x^n + c*(1 + 2*n*(1 + p)
)*x^(2*n)),x]
```

```
[Out] x*(a + b*x^n + c*x^(2*n))^(1 + p)
```

Rule 1789

```
Int[((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(p_.)*((d_) + (e_.)*(x_)^
(n_.) + (f_.)*(x_)^(n2_.)), x_Symbol] :> Simp[d*x*((a + b*x^n + c*x^(2*n))^
(p + 1)/a), x] /; FreeQ[{a, b, c, d, e, f, n, p}, x] && EqQ[n2, 2*n] && EqQ
[a*e - b*d*(n*(p + 1) + 1), 0] && EqQ[a*f - c*d*(2*n*(p + 1) + 1), 0]
```

Rubi steps

$$\text{integral} = x(a+bx^n+cx^{2n})^{1+p}$$

Mathematica [A] (verified)

Time = 0.58 (sec) , antiderivative size = 19, normalized size of antiderivative = 0.95

$$\int \left(a+bx^n+cx^{2n}\right)^p \left(a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n}\right)\,dx=x(a+x^n(b+cx^n))^{1+p}$$

```
[In] Integrate[(a + b*x^n + c*x^(2*n))^p*(a + b*(1 + n + n*p)*x^n + c*(1 + 2*n*(
1 + p))*x^(2*n)),x]
```

```
[Out] x*(a + x^n*(b + c*x^n))^(1 + p)
```

Maple [A] (verified)

Time = 5.16 (sec) , antiderivative size = 33, normalized size of antiderivative = 1.65

method	result	size
risch	$x\left(a+b\,x^{n}+c\,x^{2n}\right)\left(a+b\,x^{n}+c\,x^{2n}\right)^{p}$	33
parallelrisch	$\frac{x\,x^{n}\left(a+b\,x^{n}+c\,x^{2n}\right)^{p} bc+x\,x^{2n}\left(a+b\,x^{n}+c\,x^{2n}\right)^{p} c^{2}+x\left(a+b\,x^{n}+c\,x^{2n}\right)^{p} ac}{c}$	75

```
[In] int((a+b*x^n+c*x^(2*n))^p*(a+b*(n*p+n+1)*x^n+c*(1+2*n*(1+p))*x^(2*n)),x,met
hod=_RETURNVERBOSE)
```

```
[Out] x*(a+b*x^n+c*(x^n)^2)*(a+b*x^n+c*(x^n)^2)^p
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 35, normalized size of antiderivative = 1.75

$$\int \left(a+bx^n+cx^{2n}\right)^p \left(a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n}\right)\,dx$$
$$=\left(cxx^{2\,n}+bxx^n+ax\right)\left(cx^{2\,n}+bx^n+a\right)^p$$

```
[In] integrate((a+b*x^n+c*x^(2*n))^p*(a+b*(n*p+n+1)*x^n+c*(1+2*n*(1+p))*x^(2*n))
,x, algorithm="fricas")
```

```
[Out] (c*x*x^(2*n) + b*x*x^n + a*x)*(c*x^(2*n) + b*x^n + a)^p
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 63 vs. $2(17) = 34$.

Time = 29.90 (sec) , antiderivative size = 63, normalized size of antiderivative = 3.15

$$\int (a+bx^n+cx^{2n})^p\,(a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n})\,dx = ax(a+bx^n+cx^{2n})^p+bxx^n(a+bx^n+cx^{2n})^p+cxx^{2n}(a+bx^n+cx^{2n})^p$$

```
[In] integrate((a+b*x**n+c*x**(2*n))**p*(a+b*(n*p+n+1)*x**n+c*(1+2*n*(1+p))*x**(
2*n)),x)
```

```
[Out] a*x*(a + b*x**n + c*x**(2*n))**p + b*x*x**n*(a + b*x**n + c*x**(2*n))**p +
c*x*x**(2*n)*(a + b*x**n + c*x**(2*n))**p
```

Maxima [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 35, normalized size of antiderivative = 1.75

$$\int (a+bx^n+cx^{2n})^p\,(a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n})\,dx = (cxx^{2\,n}+bxx^n+ax)(cx^{2\,n}+bx^n+a)^p$$

```
[In] integrate((a+b*x^n+c*x^(2*n))^p*(a+b*(n*p+n+1)*x^n+c*(1+2*n*(1+p))*x^(2*n))
,x, algorithm="maxima")
```

```
[Out] (c*x*x^(2*n) + b*x*x^n + a*x)*(c*x^(2*n) + b*x^n + a)^p
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 66 vs. $2(20) = 40$.

Time = 0.37 (sec) , antiderivative size = 66, normalized size of antiderivative = 3.30

$$\int (a+bx^n+cx^{2n})^p\,(a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n})\,dx = (cx^{2\,n}+bx^n+a)^p cxx^{2\,n}+(cx^{2\,n}+bx^n+a)^p bxx^n+(cx^{2\,n}+bx^n+a)^p ax$$

```
[In] integrate((a+b*x^n+c*x^(2*n))^p*(a+b*(n*p+n+1)*x^n+c*(1+2*n*(1+p))*x^(2*n))
,x, algorithm="giac")
```

```
[Out] (c*x^(2*n) + b*x^n + a)^p*c*x*x^(2*n) + (c*x^(2*n) + b*x^n + a)^p*b*x*x^n +
 (c*x^(2*n) + b*x^n + a)^p*a*x
```

Mupad [B] (verification not implemented)

Time = 8.88 (sec) , antiderivative size = 35, normalized size of antiderivative = 1.75

$$\int (a+bx^n+cx^{2n})^p (a+b(1+n+np)x^n+c(1+2n(1+p))x^{2n}) \, dx = (a+b\,x^n+c\,x^{2\,n})^p \,(a\,x+b\,x\,x^n+c\,x\,x^{2\,n})$$

```
[In] int((a + b*x^n + c*x^(2*n))^p*(a + b*x^n*(n + n*p + 1) + c*x^(2*n)*(2*n*(p
+ 1) + 1)),x)
[Out] (a + b*x^n + c*x^(2*n))^p*(a*x + b*x*x^n + c*x*x^(2*n))
```

3.12 $\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx$

Optimal result

Integrand size = 52, antiderivative size = 45

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=-\frac{2\left(ag+2ahx^{n/4}-cfx^{n/2}\right)}{an\sqrt{a+cx^n}}$$

```
[Out] -2*(a*g+2*a*h*x^(1/4*n)-c*f*x^(1/2*n))/a/n/(a+c*x^n)^(1/2)
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 45, normalized size of antiderivative = 1.00, number of steps used = 1, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.019, Rules used = {1830}

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=-\frac{2\left(ag+2ahx^{n/4}-cfx^{n/2}\right)}{an\sqrt{a+cx^n}}$$

```
[In] Int[(x^(-1 + n/4)*(-(a*h) + c*f*x^(n/4) + c*g*x^((3*n)/4) + c*h*x^n))/(a +
c*x^n)^(3/2),x]
```

```
[Out] (-2*(a*g + 2*a*h*x^(n/4) - c*f*x^(n/2)))/(a*n*Sqrt[a + c*x^n])
```

Rule 1830

```
Int[((x_)^(m_.)*((e_) + (h_.)*(x_)^(n_.) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r
_.)))/((a_) + (c_.)*(x_)^(n_.))^(3/2), x_Symbol] :> Simp[-(2*a*g + 4*a*h*x^
(n/4) - 2*c*f*x^(n/2))/(a*c*n*Sqrt[a + c*x^n]), x] /; FreeQ[{a, c, e, f, g,
 h, m, n}, x] && EqQ[q, n/4] && EqQ[r, 3*(n/4)] && EqQ[4*m - n + 4, 0] && E
qQ[c*e + a*h, 0]
```

Rubi steps

$$\text{integral} = -\frac{2\left(ag + 2ahx^{n/4} - cfx^{n/2}\right)}{an\sqrt{a + cx^n}}$$

Mathematica [A] (verified)

Time = 0.71 (sec) , antiderivative size = 45, normalized size of antiderivative = 1.00

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah + cfx^{n/4} + cgx^{3n/4} + chx^n\right)}{(a + cx^n)^{3/2}}\,dx = \frac{2cfx^{n/2} - 2a\left(g + 2hx^{n/4}\right)}{an\sqrt{a + cx^n}}$$

```
[In] Integrate[(x^(-1 + n/4)*(-(a*h) + c*f*x^(n/4) + c*g*x^((3*n)/4) + c*h*x^n))
/(a + c*x^n)^(3/2),x]
[Out] (2*c*f*x^(n/2) - 2*a*(g + 2*h*x^(n/4)))/(a*n*Sqrt[a + c*x^n])
```

Maple [F]

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah + cf\,x^{\frac{n}{4}} + cg\,x^{\frac{3n}{4}} + ch\,x^n\right)}{(a + c\,x^n)^{\frac{3}{2}}}dx$$

```
[In] int(x^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)^(3/2)
,x)
[Out] int(x^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)^(3/2)
,x)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 48, normalized size of antiderivative = 1.07

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah + cfx^{n/4} + cgx^{3n/4} + chx^n\right)}{(a + cx^n)^{3/2}}\,dx = \frac{2\left(cfx^{\frac{1}{2}n} - 2\,ahx^{\frac{1}{4}n} - ag\right)\sqrt{cx^n + a}}{acnx^n + a^2n}$$

```
[In] integrate(x^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)
^(3/2),x, algorithm="fricas")
[Out] 2*(c*f*x^(1/2*n) - 2*a*h*x^(1/4*n) - a*g)*sqrt(c*x^n + a)/(a*c*n*x^n + a^2*
n)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=\text{Timed out}$$

```
[In] integrate(x**(-1+1/4*n)*(-a*h+c*f*x**(1/4*n)+c*g*x**(3/4*n)+c*h*x**n)/(a+c*
x**n)**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=\int \frac{\left(cgx^{\frac{3}{4}n}+cfx^{\frac{1}{4}n}+chx^n-ah\right)x^{\frac{1}{4}n-1}}{(cx^n+a)^{\frac{3}{2}}}\,dx$$

```
[In] integrate(x^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)
^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((c*g*x^(3/4*n) + c*f*x^(1/4*n) + c*h*x^n - a*h)*x^(1/4*n - 1)/(c*
x^n + a)^(3/2), x)
```

Giac [A] (verification not implemented)

none

Time = 5.16 (sec) , antiderivative size = 39, normalized size of antiderivative = 0.87

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=\frac{2\left(\left(\frac{cf(x^n)^{\frac{1}{4}}}{a}-2h\right)(x^n)^{\frac{1}{4}}-g\right)}{\sqrt{cx^n+a}n}$$

```
[In] integrate(x^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)
^(3/2),x, algorithm="giac")
```

```
[Out] 2*((c*f*(x^n)^(1/4)/a - 2*h)*(x^n)^(1/4) - g)/(sqrt(c*x^n + a)*n)
```

Mupad [B] (verification not implemented)

Time = 9.08 (sec) , antiderivative size = 39, normalized size of antiderivative = 0.87

$$\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{\left(a+cx^n\right)^{3/2}}\,dx=-\frac{2\left(a\,g-c\,f\,x^{n/2}+2\,a\,h\,x^{n/4}\right)}{a\,n\,\sqrt{a+c\,x^n}}$$

```
[In] int((x^(n/4 - 1)*(c*h*x^n - a*h + c*f*x^(n/4) + c*g*x^((3*n)/4)))/(a + c*x^
n)^(3/2),x)
```

```
[Out] -(2*(a*g - c*f*x^(n/2) + 2*a*h*x^(n/4)))/(a*n*(a + c*x^n)^(1/2))
```

3.13 $\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx$

Optimal result

Integrand size = 54, antiderivative size = 65

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx = -\frac{2x^{1-\frac{n}{4}}(dx)^{\frac{1}{4}(-4+n)}\left(ag+2ahx^{n/4}-cfx^{n/2}\right)}{an\sqrt{a+cx^n}}$$

```
[Out] -2*x^(1-1/4*n)*(d*x)^(-1+1/4*n)*(a*g+2*a*h*x^(1/4*n)-c*f*x^(1/2*n))/a/n/(a+
c*x^n)^(1/2)
```

Rubi [A] (verified)

Time = 0.10 (sec) , antiderivative size = 65, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.037, Rules used = {1831, 1830}

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx = -\frac{2x^{1-\frac{n}{4}}(dx)^{\frac{n-4}{4}}\left(ag+2ahx^{n/4}-cfx^{n/2}\right)}{an\sqrt{a+cx^n}}$$

```
[In] Int[((d*x)^(-1 + n/4)*(-(a*h) + c*f*x^(n/4) + c*g*x^((3*n)/4) + c*h*x^n))/(
a + c*x^n)^(3/2),x]
```

```
[Out] (-2*x^(1 - n/4)*(d*x)^((-4 + n)/4)*(a*g + 2*a*h*x^(n/4) - c*f*x^(n/2)))/(a*
n*Sqrt[a + c*x^n])
```

Rule 1830

```
Int[((x_)^(m_.)*((e_) + (h_.)*(x_)^(n_.) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r
_.)))/((a_) + (c_.)*(x_)^(n_.))^(3/2), x_Symbol] :> Simp[-(2*a*g + 4*a*h*x^
```

```
(n/4) - 2*c*f*x^(n/2))/(a*c*n*Sqrt[a + c*x^n]), x] /; FreeQ[{a, c, e, f, g,
 h, m, n}, x] && EqQ[q, n/4] && EqQ[r, 3*(n/4)] && EqQ[4*m - n + 4, 0] && E
qQ[c*e + a*h, 0]
```

Rule 1831

```
Int[(((d_)*(x_))^(m_.)*((e_) + (h_.)*(x_)^(n_.) + (f_.)*(x_)^(q_.) + (g_.)*
(x_)^(r_.)))/((a_) + (c_.)*(x_)^(n_.))^(3/2), x_Symbol] :> Dist[(d*x)^m/x^m
, Int[x^m*((e + f*x^(n/4) + g*x^((3*n)/4) + h*x^n)/(a + c*x^n)^(3/2)), x],
x] /; FreeQ[{a, c, d, e, f, g, h, m, n}, x] && EqQ[4*m - n + 4, 0] && EqQ[q
, n/4] && EqQ[r, 3*(n/4)] && EqQ[c*e + a*h, 0]
```

Rubi steps

$$\text{integral} = \left(x^{1-\frac{n}{4}}(dx)^{-1+\frac{n}{4}}\right)\int \frac{x^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx$$
$$= -\frac{2x^{1-\frac{n}{4}}(dx)^{\frac{1}{4}(-4+n)}\left(ag+2ahx^{n/4}-cfx^{n/2}\right)}{an\sqrt{a+cx^n}}$$

Mathematica [A] (verified)

Time = 0.67 (sec) , antiderivative size = 64, normalized size of antiderivative = 0.98

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx = \frac{2x^{-n/4}(dx)^{n/4}\left(cfx^{n/2}-a\left(g+2hx^{n/4}\right)\right)}{adn\sqrt{a+cx^n}}$$

```
[In] Integrate[((d*x)^(-1 + n/4)*(-(a*h) + c*f*x^(n/4) + c*g*x^((3*n)/4) + c*h*x
^n))/(a + c*x^n)^(3/2),x]
[Out] (2*(d*x)^(n/4)*(c*f*x^(n/2) - a*(g + 2*h*x^(n/4))))/(a*d*n*x^(n/4)*Sqrt[a +
 c*x^n])
```

Maple [F]

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cf\,x^{\frac{n}{4}}+cg\,x^{\frac{3n}{4}}+ch\,x^n\right)}{(a+c\,x^n)^{\frac{3}{2}}}dx$$

```
[In] int((d*x)^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)^(
3/2),x)
[Out] int((d*x)^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*x^n)^(
3/2),x)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.06

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=\frac{2\left(cd^{\frac{1}{4}\,n-1}fx^{\frac{1}{2}\,n}-2\,ad^{\frac{1}{4}\,n-1}hx^{\frac{1}{4}\,n}-ad^{\frac{1}{4}\,n-1}g\right)\sqrt{cx^n+a}}{acnx^n+a^2n}$$

```
[In] integrate((d*x)^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*
x^n)^(3/2),x, algorithm="fricas")
```

```
[Out] 2*(c*d^(1/4*n - 1)*f*x^(1/2*n) - 2*a*d^(1/4*n - 1)*h*x^(1/4*n) - a*d^(1/4*n
 - 1)*g)*sqrt(c*x^n + a)/(a*c*n*x^n + a^2*n)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 117.27 (sec) , antiderivative size = 160, normalized size of antiderivative = 2.46

$$\int \frac{(dx)^{-1+\frac{n}{4}}\left(-ah+cfx^{n/4}+cgx^{3n/4}+chx^n\right)}{(a+cx^n)^{3/2}}\,dx=\frac{2\sqrt{c}d^{\frac{n}{4}-1}f}{an\sqrt{\frac{ax^{-n}}{c}+1}}-\frac{2d^{\frac{n}{4}-1}g}{\sqrt{a}n\sqrt{1+\frac{cx^n}{a}}}-\frac{d^{\frac{n}{4}-1}hx^{\frac{n}{4}}\Gamma\left(\frac{1}{4}\right){}_2F_1\left(\begin{matrix}\frac{1}{4},\frac{3}{2}\\\frac{5}{4}\end{matrix}\middle|\frac{cx^ne^{i\pi}}{a}\right)}{\sqrt{a}n\Gamma\left(\frac{5}{4}\right)}+\frac{cd^{\frac{n}{4}-1}hx^{\frac{5n}{4}}\Gamma\left(\frac{5}{4}\right){}_2F_1\left(\begin{matrix}\frac{5}{4},\frac{3}{2}\\\frac{9}{4}\end{matrix}\middle|\frac{cx^ne^{i\pi}}{a}\right)}{a^{\frac{3}{2}}n\Gamma\left(\frac{9}{4}\right)}$$

```
[In] integrate((d*x)**(-1+1/4*n)*(-a*h+c*f*x**(1/4*n)+c*g*x**(3/4*n)+c*h*x**n)/(
a+c*x**n)**(3/2),x)
```

```
[Out] 2*sqrt(c)*d**(n/4 - 1)*f/(a*n*sqrt(a/(c*x**n) + 1)) - 2*d**(n/4 - 1)*g/(sqr
t(a)*n*sqrt(1 + c*x**n/a)) - d**(n/4 - 1)*h*x**(n/4)*gamma(1/4)*hyper((1/4,
 3/2), (5/4,), c*x**n*exp_polar(I*pi)/a)/(sqrt(a)*n*gamma(5/4)) + c*d**(n/4
 - 1)*h*x**(5*n/4)*gamma(5/4)*hyper((5/4, 3/2), (9/4,), c*x**n*exp_polar(I*
pi)/a)/(a**(3/2)*n*gamma(9/4))
```

Maxima [F]

$$\int \frac{(dx)^{-1+\frac{n}{4}} \left(-ah + cfx^{n/4} + cgx^{3n/4} + chx^n\right)}{(a + cx^n)^{3/2}}\, dx = \int \frac{\left(cgx^{\frac{3}{4}\,n} + cfx^{\frac{1}{4}\,n} + chx^n - ah\right)(dx)^{\frac{1}{4}\,n-1}}{(cx^n + a)^{\frac{3}{2}}}\, dx$$

```
[In] integrate((d*x)^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*
x^n)^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((c*g*x^(3/4*n) + c*f*x^(1/4*n) + c*h*x^n - a*h)*(d*x)^(1/4*n - 1)
/(c*x^n + a)^(3/2), x)
```

Giac [F]

$$\int \frac{(dx)^{-1+\frac{n}{4}} \left(-ah + cfx^{n/4} + cgx^{3n/4} + chx^n\right)}{(a + cx^n)^{3/2}}\, dx = \int \frac{\left(cgx^{\frac{3}{4}\,n} + cfx^{\frac{1}{4}\,n} + chx^n - ah\right)(dx)^{\frac{1}{4}\,n-1}}{(cx^n + a)^{\frac{3}{2}}}\, dx$$

```
[In] integrate((d*x)^(-1+1/4*n)*(-a*h+c*f*x^(1/4*n)+c*g*x^(3/4*n)+c*h*x^n)/(a+c*
x^n)^(3/2),x, algorithm="giac")
```

```
[Out] integrate((c*g*x^(3/4*n) + c*f*x^(1/4*n) + c*h*x^n - a*h)*(d*x)^(1/4*n - 1)
/(c*x^n + a)^(3/2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(dx)^{-1+\frac{n}{4}} \left(-ah + cfx^{n/4} + cgx^{3n/4} + chx^n\right)}{(a + cx^n)^{3/2}}\, dx = \int \frac{(d\,x)^{\frac{n}{4}-1}\,\left(c\,h\,x^n - a\,h + c\,f\,x^{n/4} + c\,g\,x^{\frac{3\,n}{4}}\right)}{(a + c\,x^n)^{3/2}}\, dx$$

```
[In] int(((d*x)^(n/4 - 1)*(c*h*x^n - a*h + c*f*x^(n/4) + c*g*x^((3*n)/4)))/(a +
c*x^n)^(3/2),x)
```

```
[Out] int(((d*x)^(n/4 - 1)*(c*h*x^n - a*h + c*f*x^(n/4) + c*g*x^((3*n)/4)))/(a +
c*x^n)^(3/2), x)
```

3.14 $\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx$

Optimal result

Integrand size = 61, antiderivative size = 75

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(c(bf-2ag)+\left(b^2-4ac\right)hx^{n/2}+c(2cf-bg)x^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

```
[Out] -2*(c*(-2*a*g+b*f)+(-4*a*c+b^2)*h*x^(1/2*n)+c*(-b*g+2*c*f)*x^n)/(-4*a*c+b^2
)/n/(a+b*x^n+c*x^(2*n))^(1/2)
```

Rubi [A] (verified)

Time = 0.07 (sec) , antiderivative size = 75, normalized size of antiderivative = 1.00, number of steps used = 1, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.016, Rules used = {1767}

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(hx^{n/2}\left(b^2-4ac\right)+c(bf-2ag)+cx^n(2cf-bg)\right)}{n\left(b^2-4ac\right)\sqrt{a+bx^n+cx^{2n}}}$$

```
[In] Int[(x^(-1 + n/2)*(-(a*h) + c*f*x^(n/2) + c*g*x^((3*n)/2) + c*h*x^(2*n)))/(
a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] (-2*(c*(b*f - 2*a*g) + (b^2 - 4*a*c)*h*x^(n/2) + c*(2*c*f - b*g)*x^n))/((b^
2 - 4*a*c)*n*Sqrt[a + b*x^n + c*x^(2*n)])
```

Rule 1767

```
Int[((x_)^(m_.)*((e_) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r_.) + (h_.)*(x_)^(s
_.)))/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(3/2), x_Symbol] :> Sim
p[-(2*c*(b*f - 2*a*g) + 2*h*(b^2 - 4*a*c)*x^(n/2) + 2*c*(2*c*f - b*g)*x^n)/
(c*n*(b^2 - 4*a*c)*Sqrt[a + b*x^n + c*x^(2*n)]), x] /; FreeQ[{a, b, c, e, f
, g, h, m, n}, x] && EqQ[n2, 2*n] && EqQ[q, n/2] && EqQ[r, 3*(n/2)] && EqQ[
s, 2*n] && NeQ[b^2 - 4*a*c, 0] && EqQ[2*m - n + 2, 0] && EqQ[c*e + a*h, 0]
```

Rubi steps

$$\text{integral} = -\frac{2\left(c(bf-2ag)+\left(b^2-4ac\right)hx^{n/2}+c(2cf-bg)x^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

Mathematica [A] (verified)

Time = 0.08 (sec) , antiderivative size = 84, normalized size of antiderivative = 1.12

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(bcf-2acg+b^2hx^{n/2}-4achx^{n/2}+2c^2fx^n-bcgx^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

```
[In] Integrate[(x^(-1 + n/2)*(-(a*h) + c*f*x^(n/2) + c*g*x^((3*n)/2) + c*h*x^(2*
n)))/(a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] (-2*(b*c*f - 2*a*c*g + b^2*h*x^(n/2) - 4*a*c*h*x^(n/2) + 2*c^2*f*x^n - b*c*
g*x^n))/((b^2 - 4*a*c)*n*Sqrt[a + b*x^n + c*x^(2*n)])
```

Maple [F]

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cf\,x^{\frac{n}{2}}+cg\,x^{\frac{3n}{2}}+ch\,x^{2n}\right)}{\left(a+b\,x^{n}+c\,x^{2n}\right)^{\frac{3}{2}}}dx$$

```
[In] int(x^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*x^n+c*
x^(2*n))^(3/2),x)
```

```
[Out] int(x^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*x^n+c*
x^(2*n))^(3/2),x)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 109, normalized size of antiderivative = 1.45

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(bcf-2\,acg+(b^2-4\,ac)hx^{\frac{1}{2}\,n}+(2\,c^2f-bcg)x^n\right)\sqrt{cx^{2\,n}+bx^n+a}}{(b^2c-4\,ac^2)nx^{2\,n}+(b^3-4\,abc)nx^n+(ab^2-4\,a^2c)n}$$

```
[In] integrate(x^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*
x^n+c*x^(2*n))^(3/2),x, algorithm="fricas")
```

```
[Out] -2*(b*c*f - 2*a*c*g + (b^2 - 4*a*c)*h*x^(1/2*n) + (2*c^2*f - b*c*g)*x^n)*sq
rt(c*x^(2*n) + b*x^n + a)/((b^2*c - 4*a*c^2)*n*x^(2*n) + (b^3 - 4*a*b*c)*n*
x^n + (a*b^2 - 4*a^2*c)*n)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate(x**(-1+1/2*n)*(-a*h+c*f*x**(1/2*n)+c*g*x**(3/2*n)+c*h*x**(2*n))/(
a+b*x**n+c*x**(2*n))**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \int \frac{\left(chx^{2\,n}+cgx^{\frac{3}{2}\,n}+cfx^{\frac{1}{2}\,n}-ah\right)x^{\frac{1}{2}\,n-1}}{\left(cx^{2\,n}+bx^n+a\right)^{\frac{3}{2}}}\,dx$$

```
[In] integrate(x^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*
x^n+c*x^(2*n))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((c*h*x^(2*n) + c*g*x^(3/2*n) + c*f*x^(1/2*n) - a*h)*x^(1/2*n - 1)
/(c*x^(2*n) + b*x^n + a)^(3/2), x)
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 187 vs. 2(71) = 142.

Time = 1.00 (sec) , antiderivative size = 187, normalized size of antiderivative = 2.49

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left(\sqrt{x^n}\left(\frac{(2\,b^2c^2f-8\,ac^3f-b^3cg+4\,abc^2g)\sqrt{x^n}}{b^4-8\,ab^2c+16\,a^2c^2}+\frac{b^4h-8\,ab^2ch+16\,a^2c^2h}{b^4-8\,ab^2c+16\,a^2c^2}\right)+\frac{b^3cf-4\,abc^2f-2\,ab^2cg+8\,a^2c^2g}{b^4-8\,ab^2c+16\,a^2c^2}\right)}{\sqrt{cx^{2\,n}+bx^n+a}n}$$

```
[In] integrate(x^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*
x^n+c*x^(2*n))^(3/2),x, algorithm="giac")
```

```
[Out] -2*(sqrt(x^n)*((2*b^2*c^2*f - 8*a*c^3*f - b^3*c*g + 4*a*b*c^2*g)*sqrt(x^n)/
(b^4 - 8*a*b^2*c + 16*a^2*c^2) + (b^4*h - 8*a*b^2*c*h + 16*a^2*c^2*h)/(b^4
- 8*a*b^2*c + 16*a^2*c^2)) + (b^3*c*f - 4*a*b*c^2*f - 2*a*b^2*c*g + 8*a^2*c
^2*g)/(b^4 - 8*a*b^2*c + 16*a^2*c^2))/(sqrt(c*x^(2*n) + b*x^n + a)*n)
```

Mupad [B] (verification not implemented)

Time = 8.88 (sec) , antiderivative size = 80, normalized size of antiderivative = 1.07

$$\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\,b^2\,h\,x^{n/2}-4\,a\,c\,g+2\,b\,c\,f+4\,c^2\,f\,x^n-8\,a\,c\,h\,x^{n/2}-2\,b\,c\,g\,x^n}{\left(b^2\,n-4\,a\,c\,n\right)\,\sqrt{a+b\,x^n+c\,x^{2\,n}}}$$

```
[In] int((x^(n/2 - 1)*(c*f*x^(n/2) - a*h + c*g*x^((3*n)/2) + c*h*x^(2*n)))/(a +
b*x^n + c*x^(2*n))^(3/2),x)
```

```
[Out] -(2*b^2*h*x^(n/2) - 4*a*c*g + 2*b*c*f + 4*c^2*f*x^n - 8*a*c*h*x^(n/2) - 2*b
*c*g*x^n)/((b^2*n - 4*a*c*n)*(a + b*x^n + c*x^(2*n))^(1/2))
```

3.15 $\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx$

Optimal result

Integrand size = 63, antiderivative size = 95

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2x^{1-\frac{n}{2}}(dx)^{\frac{1}{2}(-2+n)}\left(c(bf-2ag)+\left(b^2-4ac\right)hx^{n/2}+c(2cf-bg)x^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

```
[Out] -2*x^(1-1/2*n)*(d*x)^(-1+1/2*n)*(c*(-2*a*g+b*f)+(-4*a*c+b^2)*h*x^(1/2*n)+c*
(-b*g+2*c*f)*x^n)/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))^(1/2)
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.00, number of steps used = 2, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.032, Rules used = {1768, 1767}

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2x^{1-\frac{n}{2}}(dx)^{\frac{n-2}{2}}\left(hx^{n/2}(b^2-4ac)+c(bf-2ag)+cx^n(2cf-bg)\right)}{n\left(b^2-4ac\right)\sqrt{a+bx^n+cx^{2n}}}$$

```
[In] Int[((d*x)^(-1 + n/2)*(-(a*h) + c*f*x^(n/2) + c*g*x^((3*n)/2) + c*h*x^(2*n)
))/(a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] (-2*x^(1 - n/2)*(d*x)^((-2 + n)/2)*(c*(b*f - 2*a*g) + (b^2 - 4*a*c)*h*x^(n/
2) + c*(2*c*f - b*g)*x^n))/((b^2 - 4*a*c)*n*Sqrt[a + b*x^n + c*x^(2*n)])
```

Rule 1767

```
Int[((x_)^(m_.)*((e_) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r_.) + (h_.)*(x_)^(s
_.)))/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(3/2), x_Symbol] :> Sim
p[-(2*c*(b*f - 2*a*g) + 2*h*(b^2 - 4*a*c)*x^(n/2) + 2*c*(2*c*f - b*g)*x^n)/
(c*n*(b^2 - 4*a*c)*Sqrt[a + b*x^n + c*x^(2*n)]), x] /; FreeQ[{a, b, c, e, f
, g, h, m, n}, x] && EqQ[n2, 2*n] && EqQ[q, n/2] && EqQ[r, 3*(n/2)] && EqQ[
s, 2*n] && NeQ[b^2 - 4*a*c, 0] && EqQ[2*m - n + 2, 0] && EqQ[c*e + a*h, 0]
```

Rule 1768

```
Int[(((d_)*(x_))^(m_.)*((e_) + (f_.)*(x_)^(q_.) + (g_.)*(x_)^(r_.) + (h_.)*
(x_)^(s_.)))/((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(3/2), x_Symbol]
 :> Dist[(d*x)^m/x^m, Int[x^m*((e + f*x^(n/2) + g*x^((3*n)/2) + h*x^(2*n))/
(a + b*x^n + c*x^(2*n))^(3/2)), x], x] /; FreeQ[{a, b, c, d, e, f, g, h, m,
 n}, x] && EqQ[n2, 2*n] && EqQ[q, n/2] && EqQ[r, 3*(n/2)] && EqQ[s, 2*n] &&
 NeQ[b^2 - 4*a*c, 0] && EqQ[2*m - n + 2, 0] && EqQ[c*e + a*h, 0]
```

Rubi steps

$$\text{integral} = \left(x^{1-\frac{n}{2}}(dx)^{-1+\frac{n}{2}}\right)\int \frac{x^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx$$

$$= -\frac{2x^{1-\frac{n}{2}}(dx)^{\frac{1}{2}(-2+n)}\left(c(bf-2ag)+\left(b^2-4ac\right)hx^{n/2}+c(2cf-bg)x^n\right)}{\left(b^2-4ac\right)n\sqrt{a+bx^n+cx^{2n}}}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 242 vs. 2(95) = 190.

Time = 3.46 (sec) , antiderivative size = 242, normalized size of antiderivative = 2.55

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx =$$

$$-\frac{x^{-n/2}(dx)^{n/2}\left(-2ab^2hx^{n/2}-2abc(f-gx^n)+4ac\left(-cfx^n+a\left(g+2hx^{n/2}\right)\right)+b\sqrt{c}(bf-2ag)\sqrt{a+x^n\,(b}\right.}{a\left(-b^2+4ac\right)}$$

```
[In] Integrate[((d*x)^(-1 + n/2)*(-(a*h) + c*f*x^(n/2) + c*g*x^((3*n)/2) + c*h*x
^(2*n)))/(a + b*x^n + c*x^(2*n))^(3/2),x]
```

```
[Out] -(((d*x)^(n/2)*(-2*a*b^2*h*x^(n/2) - 2*a*b*c*(f - g*x^n) + 4*a*c*(-(c*f*x^n
) + a*(g + 2*h*x^(n/2))) + b*Sqrt[c]*(b*f - 2*a*g)*Sqrt[a + x^n*(b + c*x^n)
]*ArcTanh[(b + 2*c*x^n)/(2*Sqrt[c]*Sqrt[a + x^n*(b + c*x^n)])] + b*Sqrt[c]*
(b*f - 2*a*g)*Sqrt[a + x^n*(b + c*x^n)]*Log[b + 2*c*x^n - 2*Sqrt[c]*Sqrt[a
+ x^n*(b + c*x^n)]]))/(a*(-b^2 + 4*a*c)*d*n*x^(n/2)*Sqrt[a + x^n*(b + c*x^n
)]))
```

Maple [F]

$$\int \frac{(dx)^{-1+\frac{n}{2}} \left(-ah+cf\,x^{\frac{n}{2}}+cg\,x^{\frac{3n}{2}}+ch\,x^{2n}\right)}{\left(a+b\,x^{n}+c\,x^{2n}\right)^{\frac{3}{2}}}dx$$

```
[In] int((d*x)^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*x^
n+c*x^(2*n))^(3/2),x)
```

```
[Out] int((d*x)^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(a+b*x^
n+c*x^(2*n))^(3/2),x)
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 132, normalized size of antiderivative = 1.39

$$\int \frac{(dx)^{-1+\frac{n}{2}} \left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = -\frac{2\left((b^2-4\,ac)d^{\frac{1}{2}\,n-1}hx^{\frac{1}{2}\,n}+(2\,c^2f-bcg)d^{\frac{1}{2}\,n-1}x^n+(bcf-2\,acg)d^{\frac{1}{2}\,n-1}\right)\sqrt{cx^{2\,n}+bx^n+a}}{(b^2c-4\,ac^2)nx^{2\,n}+(b^3-4\,abc)nx^n+(ab^2-4\,a^2c)n}$$

```
[In] integrate((d*x)^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(
a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="fricas")
```

```
[Out] -2*((b^2 - 4*a*c)*d^(1/2*n - 1)*h*x^(1/2*n) + (2*c^2*f - b*c*g)*d^(1/2*n -
1)*x^n + (b*c*f - 2*a*c*g)*d^(1/2*n - 1))*sqrt(c*x^(2*n) + b*x^n + a)/((b^2
*c - 4*a*c^2)*n*x^(2*n) + (b^3 - 4*a*b*c)*n*x^n + (a*b^2 - 4*a^2*c)*n)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(dx)^{-1+\frac{n}{2}} \left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((d*x)**(-1+1/2*n)*(-a*h+c*f*x**(1/2*n)+c*g*x**(3/2*n)+c*h*x**(2*n
))/(a+b*x**n+c*x**(2*n))**(3/2),x)
```

```
[Out] Timed out
```

Maxima [F]

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \int \frac{\left(chx^{2\,n}+cgx^{\frac{3}{2}\,n}+cfx^{\frac{1}{2}\,n}-ah\right)(dx)^{\frac{1}{2}\,n-1}}{\left(cx^{2\,n}+bx^n+a\right)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((d*x)^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(
a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((c*h*x^(2*n) + c*g*x^(3/2*n) + c*f*x^(1/2*n) - a*h)*(d*x)^(1/2*n
- 1)/(c*x^(2*n) + b*x^n + a)^(3/2), x)
```

Giac [F]

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \int \frac{\left(chx^{2\,n}+cgx^{\frac{3}{2}\,n}+cfx^{\frac{1}{2}\,n}-ah\right)(dx)^{\frac{1}{2}\,n-1}}{\left(cx^{2\,n}+bx^n+a\right)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((d*x)^(-1+1/2*n)*(-a*h+c*f*x^(1/2*n)+c*g*x^(3/2*n)+c*h*x^(2*n))/(
a+b*x^n+c*x^(2*n))^(3/2),x, algorithm="giac")
```

```
[Out] integrate((c*h*x^(2*n) + c*g*x^(3/2*n) + c*f*x^(1/2*n) - a*h)*(d*x)^(1/2*n
- 1)/(c*x^(2*n) + b*x^n + a)^(3/2), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(dx)^{-1+\frac{n}{2}}\left(-ah+cfx^{n/2}+cgx^{3n/2}+chx^{2n}\right)}{\left(a+bx^n+cx^{2n}\right)^{3/2}}\,dx = \int \frac{(d\,x)^{\frac{n}{2}-1}\left(c\,f\,x^{n/2}-a\,h+c\,g\,x^{\frac{3\,n}{2}}+c\,h\,x^{2\,n}\right)}{\left(a+b\,x^n+c\,x^{2\,n}\right)^{3/2}}\,dx$$

```
[In] int(((d*x)^(n/2 - 1)*(c*f*x^(n/2) - a*h + c*g*x^((3*n)/2) + c*h*x^(2*n)))/(
a + b*x^n + c*x^(2*n))^(3/2),x)
```

```
[Out] int(((d*x)^(n/2 - 1)*(c*f*x^(n/2) - a*h + c*g*x^((3*n)/2) + c*h*x^(2*n)))/(
a + b*x^n + c*x^(2*n))^(3/2), x)
```

3.16 $\int (gx)^m (a+bx^n+cx^{2n})^p (a(1+m)+b(1+m+n+$

Optimal result

Integrand size = 56, antiderivative size = 29

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n + c(1+m+2n(1+p))x^{2n}\right) dx = \frac{(gx)^{1+m}\left(a+bx^n+cx^{2n}\right)^{1+p}}{g}$$

```
[Out] (g*x)^(1+m)*(a+b*x^n+c*x^(2*n))^(p+1)/g
```

Rubi [A] (verified)

Time = 0.05 (sec) , antiderivative size = 29, normalized size of antiderivative = 1.00, number of steps used = 1, number of rules used = 1, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.018, Rules used = {1761}

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n + c(1+m+2n(1+p))x^{2n}\right) dx = \frac{(gx)^{m+1}\left(a+bx^n+cx^{2n}\right)^{p+1}}{g}$$

```
[In] Int[(g*x)^m*(a + b*x^n + c*x^(2*n))^p*(a*(1 + m) + b*(1 + m + n + n*p)*x^n
+ c*(1 + m + 2*n*(1 + p))*x^(2*n)),x]
```

```
[Out] ((g*x)^(1 + m)*(a + b*x^n + c*x^(2*n))^(1 + p))/g
```

Rule 1761

```
Int[((g_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_.) + (c_.)*(x_)^(n2_.))^(p_.)*
((d_) + (e_.)*(x_)^(n_.) + (f_.)*(x_)^(n2_.)), x_Symbol] :> Simp[d*(g*x)^(m
 + 1)*((a + b*x^n + c*x^(2*n))^(p + 1)/(a*g*(m + 1))), x] /; FreeQ[{a, b, c
, d, e, f, g, m, n, p}, x] && EqQ[n2, 2*n] && EqQ[a*e*(m + 1) - b*d*(m + n*
```

```
(p + 1) + 1), 0] && EqQ[a*f*(m + 1) - c*d*(m + 2*n*(p + 1) + 1), 0] && NeQ[
m, -1]
```

Rubi steps

$$\text{integral} = \frac{(gx)^{1+m}\left(a+bx^n+cx^{2n}\right)^{1+p}}{g}$$

Mathematica [A] (verified)

Time = 1.98 (sec) , antiderivative size = 24, normalized size of antiderivative = 0.83

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n + c(1+m+2n(1+p))x^{2n}\right) dx = x(gx)^m \left(a+x^n(b+cx^n)\right)^{1+p}$$

```
[In] Integrate[(g*x)^m*(a + b*x^n + c*x^(2*n))^p*(a*(1 + m) + b*(1 + m + n + n*p
)*x^n + c*(1 + m + 2*n*(1 + p))*x^(2*n)),x]
[Out] x*(g*x)^m*(a + x^n*(b + c*x^n))^(1 + p)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 89 vs. 2(29) = 58.

Time = 102.39 (sec) , antiderivative size = 90, normalized size of antiderivative = 3.10

method	result	size
parallelrisch	$\frac{x\,x^{n}(gx)^{m}\left(a+b\,x^{n}+c\,x^{2n}\right)^{p}bc+x\,x^{2n}(gx)^{m}\left(a+b\,x^{n}+c\,x^{2n}\right)^{p}c^{2}+x(gx)^{m}\left(a+b\,x^{n}+c\,x^{2n}\right)^{p}ac}{c}$	90

```
[In] int((g*x)^m*(a+b*x^n+c*x^(2*n))^p*(a*(1+m)+b*(n*p+m+n+1)*x^n+c*(1+m+2*n*(1+
p))*x^(2*n)),x,method=_RETURNVERBOSE)
[Out] (x*x^n*(g*x)^m*(a+b*x^n+c*x^(2*n))^p*b*c+x*x^(2*n)*(g*x)^m*(a+b*x^n+c*x^(2*
n))^p*c^2+x*(g*x)^m*(a+b*x^n+c*x^(2*n))^p*a*c)/c
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 65 vs. $2(29) = 58$.

Time = 0.26 (sec) , antiderivative size = 65, normalized size of antiderivative = 2.24

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n+c(1+m+2n(1+p))x^{2n}\right)\,dx = \left(cxx^{2\,n}e^{(m\log(g)+m\log(x))}+bxx^n e^{(m\log(g)+m\log(x))}+axe^{(m\log(g)+m\log(x))}\right)\left(cx^{2\,n}+bx^n+a\right)^p$$

```
[In] integrate((g*x)^m*(a+b*x^n+c*x^(2*n))^p*(a*(1+m)+b*(n*p+m+n+1)*x^n+c*(1+m+2
*n*(1+p))*x^(2*n)),x, algorithm="fricas")
```

```
[Out] (c*x*x^(2*n)*e^(m*log(g) + m*log(x)) + b*x*x^n*e^(m*log(g) + m*log(x)) + a*
x*e^(m*log(g) + m*log(x)))*(c*x^(2*n) + b*x^n + a)^p
```

Sympy [F(-1)]

Timed out.

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n+c(1+m+2n(1+p))x^{2n}\right)\,dx = \text{Timed out}$$

```
[In] integrate((g*x)**m*(a+b*x**n+c*x**(2*n))**p*(a*(1+m)+b*(n*p+m+n+1)*x**n+c*(
1+m+2*n*(1+p))*x**(2*n)),x)
```

```
[Out] Timed out
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 60 vs. $2(29) = 58$.

Time = 0.26 (sec) , antiderivative size = 60, normalized size of antiderivative = 2.07

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n+c(1+m+2n(1+p))x^{2n}\right)\,dx = \left(ag^m xx^m+cg^m xe^{(m\log(x)+2\,n\log(x))}+bg^m xe^{(m\log(x)+n\log(x))}\right)\left(cx^{2\,n}+bx^n+a\right)^p$$

```
[In] integrate((g*x)^m*(a+b*x^n+c*x^(2*n))^p*(a*(1+m)+b*(n*p+m+n+1)*x^n+c*(1+m+2
*n*(1+p))*x^(2*n)),x, algorithm="maxima")
```

```
[Out] (a*g^m*x*x^m + c*g^m*x*e^(m*log(x) + 2*n*log(x)) + b*g^m*x*e^(m*log(x) + n*
log(x)))*(c*x^(2*n) + b*x^n + a)^p
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 96 vs. $2(29) = 58$.

Time = 0.44 (sec) , antiderivative size = 96, normalized size of antiderivative = 3.31

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n + c(1+m+2n(1+p))x^{2n}\right) dx = \left(cx^{2n}+bx^n+a\right)^p cxx^{2n}e^{(m\log(g)+m\log(x))} + \left(cx^{2n}+bx^n+a\right)^p bxx^n e^{(m\log(g)+m\log(x))} + \left(cx^{2n}+bx^n+a\right)^p axe^{(m\log(g)+m\log(x))}$$

```
[In] integrate((g*x)^m*(a+b*x^n+c*x^(2*n))^p*(a*(1+m)+b*(n*p+m+n+1)*x^n+c*(1+m+2
*n*(1+p))*x^(2*n)),x, algorithm="giac")
```

```
[Out] (c*x^(2*n) + b*x^n + a)^p*c*x*x^(2*n)*e^(m*log(g) + m*log(x)) + (c*x^(2*n)
+ b*x^n + a)^p*b*x*x^n*e^(m*log(g) + m*log(x)) + (c*x^(2*n) + b*x^n + a)^p*
a*x*e^(m*log(g) + m*log(x))
```

Mupad [B] (verification not implemented)

Time = 8.73 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.72

$$\int (gx)^m \left(a+bx^n+cx^{2n}\right)^p \left(a(1+m)+b(1+m+n+np)x^n + c(1+m+2n(1+p))x^{2n}\right) dx = \left(a\,x\,{\left(g\,x\right)}^m+b\,x\,x^n\,{\left(g\,x\right)}^m + c\,x\,x^{2\,n}\,{\left(g\,x\right)}^m\right)\,{\left(a+b\,x^n+c\,x^{2\,n}\right)}^p$$

```
[In] int((g*x)^m*(a + b*x^n + c*x^(2*n))^p*(a*(m + 1) + b*x^n*(m + n + n*p + 1)
+ c*x^(2*n)*(m + 2*n*(p + 1) + 1)),x)
```

```
[Out] (a*x*(g*x)^m + b*x*x^n*(g*x)^m + c*x*x^(2*n)*(g*x)^m)*(a + b*x^n + c*x^(2*n
))^p
```

3.17 $\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{\left(a+bx^n+cx^{2n}\right)^2} \, dx$

Optimal result

Integrand size = 38, antiderivative size = 494

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{\left(a+bx^n+cx^{2n}\right)^2} \, dx$$
$$= \frac{x(Ac(b^2-2ac)-a(bBc-2acC+abD)+(bc(Ac+aC)-ab^2D-2ac(Bc-aD))\,x^n)}{ac\,(b^2-4ac)\,n\,(a+bx^n+cx^{2n})}$$
$$+ \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))+\frac{Ac^2(4ac(1-2n)-b^2(1-n))-a(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}\right.}{ac\,(b^2-4ac)\left(b-\sqrt{b^2-4ac}\right)n}$$
$$+ \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))-\frac{Ac^2(4ac(1-2n)-b^2(1-n))-a(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}\right.}{ac\,(b^2-4ac)\left(b+\sqrt{b^2-4ac}\right)n}$$

```
[Out] x*(A*c*(-2*a*c+b^2)-a*(B*b*c-2*C*a*c+D*a*b)+(b*c*(A*c+C*a)-a*b^2*D-2*a*c*(B
*c-D*a))*x^n)/a/c/(-4*a*c+b^2)/n/(a+b*x^n+c*x^(2*n))+x*hypergeom([1, 1/n],[
1+1/n],-2*c*x^n/(b-(-4*a*c+b^2)^(1/2)))*(a*b^2*D-b*c*(A*c+C*a)*(1-n)+2*a*c*
(B*c*(1-n)-a*D*(1+n))+(A*c^2*(4*a*c*(1-2*n)-b^2*(1-n))-a*(4*a*c^2*C+b^3*D-b
^2*c*C*(1-n)-2*b*c*(B*c*n+a*D*(2+n))))/(-4*a*c+b^2)^(1/2))/a/c/(-4*a*c+b^2)
/n/(b-(-4*a*c+b^2)^(1/2))+x*hypergeom([1, 1/n],[1+1/n],-2*c*x^n/(b+(-4*a*c+
b^2)^(1/2)))*(a*b^2*D-b*c*(A*c+C*a)*(1-n)+2*a*c*(B*c*(1-n)-a*D*(1+n))+(-A*c
^2*(4*a*c*(1-2*n)-b^2*(1-n))+a*(4*a*c^2*C+b^3*D-b^2*c*C*(1-n)-2*b*c*(B*c*n+
a*D*(2+n))))/(-4*a*c+b^2)^(1/2))/a/c/(-4*a*c+b^2)/n/(b+(-4*a*c+b^2)^(1/2))
```

Rubi [A] (verified)

Time = 0.96 (sec) , antiderivative size = 494, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.079, Rules used = {1808, 1436, 251}

$$\int \frac{A + Bx^n + Cx^{2n} + Dx^{3n}}{(a + bx^n + cx^{2n})^2}\,dx$$
$$= \frac{x(x^n(bc(aC + Ac) - ab^2D - 2ac(Bc - aD)) + Ac(b^2 - 2ac) - a(abD - 2acC + bBc))}{acn\,(b^2 - 4ac)\,(a + bx^n + cx^{2n})}$$
$$+ \frac{x\,\text{Hypergeometric2F1}\left(1, \frac{1}{n}, 1 + \frac{1}{n}, -\frac{2cx^n}{b-\sqrt{b^2-4ac}}\right)\left(\frac{Ac^2(4ac(1-2n)-b^2(1-n))-a(-2bc(aD(n+2)+Bcn)+4ac^2C+b^3D-b^2c}{\sqrt{b^2-4ac}}\right.}{acn\,(b^2 - 4ac)\left(b - \sqrt{b^2 - 4ac}\right)}$$
$$+ \frac{x\,\text{Hypergeometric2F1}\left(1, \frac{1}{n}, 1 + \frac{1}{n}, -\frac{2cx^n}{b+\sqrt{b^2-4ac}}\right)\left(-\frac{Ac^2(4ac(1-2n)-b^2(1-n))-a(-2bc(aD(n+2)+Bcn)+4ac^2C+b^3D-}{\sqrt{b^2-4ac}}\right.}{acn\,(b^2 - 4ac)\left(\sqrt{b^2 - 4ac} + b\right.}$$

```
[In] Int[(A + B*x^n + C*x^(2*n) + D*x^(3*n))/(a + b*x^n + c*x^(2*n))^2,x]
```

```
[Out] (x*(A*c*(b^2 - 2*a*c) - a*(b*B*c - 2*a*c*C + a*b*D) + (b*c*(A*c + a*C) - a*
b^2*D - 2*a*c*(B*c - a*D))*x^n))/(a*c*(b^2 - 4*a*c)*n*(a + b*x^n + c*x^(2*n
))) + ((a*b^2*D - b*c*(A*c + a*C)*(1 - n) + 2*a*c*(B*c*(1 - n) - a*D*(1 + n
)) + (A*c^2*(4*a*c*(1 - 2*n) - b^2*(1 - n)) - a*(4*a*c^2*C + b^3*D - b^2*c*
C*(1 - n) - 2*b*c*(B*c*n + a*D*(2 + n))))/Sqrt[b^2 - 4*a*c])*x*Hypergeometr
ic2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b - Sqrt[b^2 - 4*a*c])])/(a*c*(b^2
 - 4*a*c)*(b - Sqrt[b^2 - 4*a*c])*n) + ((a*b^2*D - b*c*(A*c + a*C)*(1 - n)
+ 2*a*c*(B*c*(1 - n) - a*D*(1 + n)) - (A*c^2*(4*a*c*(1 - 2*n) - b^2*(1 - n)
) - a*(4*a*c^2*C + b^3*D - b^2*c*C*(1 - n) - 2*b*c*(B*c*n + a*D*(2 + n))))/
Sqrt[b^2 - 4*a*c])*x*Hypergeometric2F1[1, n^(-1), 1 + n^(-1), (-2*c*x^n)/(b
 + Sqrt[b^2 - 4*a*c])])/(a*c*(b^2 - 4*a*c)*(b + Sqrt[b^2 - 4*a*c])*n)
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p
, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 1436

```
Int[((d_) + (e_.)*(x_)^(n_))/((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_)), 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^n), x], x] + Dist[e/2 - (2*c*d - b*e)/(2*q), Int[1/(
b/2 + q/2 + c*x^n), x], x]] /; FreeQ[{a, b, c, d, e, n}, x] && EqQ[n2, 2*n]
 && NeQ[b^2 - 4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && (PosQ[b^2 - 4*a
*c] ||  !IGtQ[n/2, 0])
```

Rule 1808

```
Int[(P3_)*((a_) + (b_.)*(x_)^(n_) + (c_.)*(x_)^(n2_))^(p_), x_Symbol] :> Wi
th[{d = Coeff[P3, x^n, 0], e = Coeff[P3, x^n, 1], f = Coeff[P3, x^n, 2], g
= Coeff[P3, x^n, 3]}, Simp[(-x)*(b^2*c*d - 2*a*c*(c*d - a*f) - a*b*(c*e + a
*g) + (b*c*(c*d + a*f) - a*b^2*g - 2*a*c*(c*e - a*g))*x^n)*((a + b*x^n + c*
x^(2*n))^(p + 1)/(a*c*n*(p + 1)*(b^2 - 4*a*c))), x] - Dist[1/(a*c*n*(p + 1)
*(b^2 - 4*a*c)), Int[(a + b*x^n + c*x^(2*n))^(p + 1)*Simp[a*b*(c*e + a*g) -
 b^2*c*d*(n + n*p + 1) - 2*a*c*(a*f - c*d*(2*n*(p + 1) + 1)) + (a*b^2*g*(n*
(p + 2) + 1) - b*c*(c*d + a*f)*(n*(2*p + 3) + 1) - 2*a*c*(a*g*(n + 1) - c*e
*(n*(2*p + 3) + 1)))*x^n, x], x], x]] /; FreeQ[{a, b, c, n}, x] && EqQ[n2,
2*n] && PolyQ[P3, x^n, 3] && NeQ[b^2 - 4*a*c, 0] && ILtQ[p, -1]
```

Rubi steps

integral

$$= \frac{x\left(Ac(b^2-2ac)-a(bBc-2acC+abD)+\left(bc(Ac+aC)-ab^2D-2ac(Bc-aD)\right)x^n\right)}{ac\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{\int \frac{ab(Bc+aD)-2ac(aC-Ac(1-2n))-Ab^2c(1-n)+\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))\right)x^n}{a+bx^n+cx^{2n}}\,dx}{ac\left(b^2-4ac\right)n}$$

$$= \frac{x\left(Ac(b^2-2ac)-a(bBc-2acC+abD)+\left(bc(Ac+aC)-ab^2D-2ac(Bc-aD)\right)x^n\right)}{ac\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))-\frac{Ac^2\left(4ac(1-2n)-b^2(1-n)\right)-a\left(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}}{2ac\left(b^2-4ac\right)n} + \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))+\frac{Ac^2\left(4ac(1-2n)-b^2(1-n)\right)-a\left(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}}{2ac\left(b^2-4ac\right)n}$$

$$= \frac{x\left(Ac(b^2-2ac)-a(bBc-2acC+abD)+\left(bc(Ac+aC)-ab^2D-2ac(Bc-aD)\right)x^n\right)}{ac\left(b^2-4ac\right)n\left(a+bx^n+cx^{2n}\right)} + \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))+\frac{Ac^2\left(4ac(1-2n)-b^2(1-n)\right)-a\left(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}}{ac\left(b^2-4ac\right)\left(b-\sqrt{b^2-4ac}\right)n} + \frac{\left(ab^2D-bc(Ac+aC)(1-n)+2ac(Bc(1-n)-aD(1+n))-\frac{Ac^2\left(4ac(1-2n)-b^2(1-n)\right)-a\left(4ac^2C+b^3D-b^2cC(1-}{\sqrt{b^2-4ac}}}{ac\left(b^2-4ac\right)\left(b+\sqrt{b^2-4ac}\right)n}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 5439 vs. 2(494) = 988.

Time = 8.36 (sec) , antiderivative size = 5439, normalized size of antiderivative = 11.01

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(A + B*x^n + C*x^(2*n) + D*x^(3*n))/(a + b*x^n + c*x^(2*n))^2,x]
[Out] Result too large to show
```

Maple [F]

$$\int \frac{A+B\,x^n+C\,x^{2n}+Dx^{3n}}{(a+b\,x^n+c\,x^{2n})^2}\,dx$$

```
[In] int((A+B*x^n+C*x^(2*n)+D*x^(3*n))/(a+b*x^n+c*x^(2*n))^2,x)
[Out] int((A+B*x^n+C*x^(2*n)+D*x^(3*n))/(a+b*x^n+c*x^(2*n))^2,x)
```

Fricas [F]

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{capitalDx^{3\,n}+Cx^{2\,n}+Bx^n+A}{(cx^{2\,n}+bx^n+a)^2}\,dx$$

```
[In] integrate((A+B*x^n+C*x^(2*n)+D*x^(3*n))/(a+b*x^n+c*x^(2*n))^2,x, algorithm=
"fricas")
[Out] integral((D*x^(3*n) + C*x^(2*n) + B*x^n + A)/(c^2*x^(4*n) + b^2*x^(2*n) + 2
*a*b*x^n + a^2 + 2*(b*c*x^n + a*c)*x^(2*n)), x)
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*x**n+C*x**(2*n)+D*x**(3*n))/(a+b*x**n+c*x**(2*n))**2,x)
[Out] Timed out
```

Maxima [F]

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{Dx^{3n}+Cx^{2n}+Bx^n+A}{(cx^{2n}+bx^n+a)^2}\,dx$$

```
[In] integrate((A+B*x^n+C*x^(2*n)+D*x^(3*n))/(a+b*x^n+c*x^(2*n))^2,x, algorithm=
"maxima")
```

```
[Out] ((C*a*b*c - 2*B*a*c^2 + A*b*c^2 - (a*b^2 - 2*a^2*c)*D)*x*x^n - (D*a^2*b - 2
*C*a^2*c + B*a*b*c - (b^2*c - 2*a*c^2)*A)*x)/(a^2*b^2*c*n - 4*a^3*c^2*n + (
a*b^2*c^2*n - 4*a^2*c^3*n)*x^(2*n) + (a*b^3*c*n - 4*a^2*b*c^2*n)*x^n) - int
egrate(-(D*a^2*b - 2*C*a^2*c + B*a*b*c - (2*a*c^2*(2*n - 1) - b^2*c*(n - 1)
)*A + (C*a*b*c*(n - 1) - 2*B*a*c^2*(n - 1) + A*b*c^2*(n - 1) - (2*a^2*c*(n
+ 1) - a*b^2)*D)*x^n)/(a^2*b^2*c*n - 4*a^3*c^2*n + (a*b^2*c^2*n - 4*a^2*c^3
*n)*x^(2*n) + (a*b^3*c*n - 4*a^2*b*c^2*n)*x^n), x)
```

Giac [F]

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{Dx^{3n}+Cx^{2n}+Bx^n+A}{(cx^{2n}+bx^n+a)^2}\,dx$$

```
[In] integrate((A+B*x^n+C*x^(2*n)+D*x^(3*n))/(a+b*x^n+c*x^(2*n))^2,x, algorithm=
"giac")
```

```
[Out] integrate((D*x^(3*n) + C*x^(2*n) + B*x^n + A)/(c*x^(2*n) + b*x^n + a)^2, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+Bx^n+Cx^{2n}+Dx^{3n}}{(a+bx^n+cx^{2n})^2}\,dx = \int \frac{A+C\,x^{2n}+x^{3n}\,D+B\,x^n}{(a+b\,x^n+c\,x^{2n})^2}\,dx$$

```
[In] int((A + C*x^(2*n) + x^(3*n)*D + B*x^n)/(a + b*x^n + c*x^(2*n))^2,x)
```

```
[Out] int((A + C*x^(2*n) + x^(3*n)*D + B*x^n)/(a + b*x^n + c*x^(2*n))^2, x)
```

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_co
                  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 opt
                                    convert(leaf_count_result,string),"$ vs. $2(",
                                    convert(leaf_count_optimal,string),")=",convert(2*leaf_count
              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 large
                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 isinst
    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_resul


print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

return grade, grade_annotation
```