```
> restart;
> trace(int);
> infolevel[all]:=2;
> printlevel:= 20;
> int(x^3*exp(arcsin(x))/sqrt(1-x^2),x);
```

$$[int:\text{-}ModuleApply]$$

$$infolevel_{all} := 2$$

$$printlevel := 20$$

```
{--> enter arcsin, args = x
{--> enter type/SymbolicInfinity, args = x
```

$$false$$

```
<-- exit type/SymbolicInfinity (now in arcsin) = false}
{--> enter arcsin/normal, args = x
{--> enter tools/csgn_k_times_k, args = x, x
```

$$zz := false$$

$$false$$

```
<-- exit tools/csgn_k_times_k (now in arcsin/normal) = false}
```

$$c := 1$$

$$x := x$$

```
{--> enter tools/sign, args = x
```

$$s := -x$$

$$1$$

```
<-- exit tools/sign (now in arcsin/normal) = 1}
```

$$s := 1$$

$$y := Cache\left(512, 'permanent' = \left[0 = 0, 1 = \frac{1}{2}\pi, FAIL = FAIL, \frac{1}{2}\sqrt{2-\sqrt{2}} = \frac{1}{8}\pi,\right.\right.$$
$$\frac{1}{4}\sqrt{2}\sqrt{5-\sqrt{5}} = \frac{1}{5}\pi, \frac{1}{4}\sqrt{6}\left(1+\frac{1}{3}\sqrt{3}\right) = \frac{5}{12}\pi, \frac{1}{4}\sqrt{2}\left(\sqrt{3}-1\right) = \frac{1}{12}\pi,$$
$$\frac{1}{4}\sqrt{5}-\frac{1}{4} = \frac{1}{10}\pi, \frac{1}{2}\sqrt{3} = \frac{1}{3}\pi, \frac{1}{2}\sqrt{2+\sqrt{2}} = \frac{3}{8}\pi, \frac{1}{4}\sqrt{2}\left(1+\sqrt{3}\right) = \frac{5}{12}\pi,$$
$$\frac{1}{4}\sqrt{6}\left(1-\frac{1}{3}\sqrt{3}\right) = \frac{1}{12}\pi, \frac{1}{4}\sqrt{2}\sqrt{5+\sqrt{5}} = \frac{2}{5}\pi, \frac{1}{4}\sqrt{5}+\frac{1}{4} = \frac{3}{10}\pi, I$$
$$\left.\left.= I\ln\left(1+\sqrt{2}\right), \frac{1}{2} = \frac{1}{6}\pi, \frac{1}{2}\sqrt{2} = \frac{1}{4}\pi\right]\right)$$

$$y := \arcsin(x)$$

$$\arcsin(-x) := -\arcsin(x)$$

$$\arcsin(x) := \arcsin(x)$$

```
<-- exit arcsin/normal (now in arcsin) = arcsin(x)}
```

$$res := \arcsin(x)$$

$$\arcsin(x) := \arcsin(x)$$

```
<-- exit arcsin (now at top level) = arcsin(x)}
{--> enter exp, args = arcsin(x)
{--> enter type/SymbolicInfinity, args = arcsin(x)
```

$$false$$

```
<-- exit type/SymbolicInfinity (now in exp) = false}
value remembered (in exp): type/SymbolicInfinity(arcsin(x)) ->
```

```
false
```

$$n := \arcsin$$

$$t := x$$

$$res := e^{\arcsin(x)}$$

$$e^{\arcsin(x)} := e^{\arcsin(x)}$$

```
<-- exit exp (now at top level) = exp(arcsin(x))}
{--> enter sqrt:-ModuleApply, args = -x^2+1
```

$$y := -x^2 + 1$$

$$c := 1$$

$$s := -1$$

$$y := x^2 - 1$$

```
{--> enter sqrt:-ModuleApply, args = 1
```

$$s := 1$$

```
<-- exit sqrt:-ModuleApply (now in sqrt:-ModuleApply) = 1}
```

$$r := 1$$

$$c := -1$$

$$s := 1$$

```
{--> enter psqrt, args = x^2-1
{--> enter psqrt/psqrt, args = x^2-1
```

$$a := x^2 - 1$$

$$i := 1$$

$$v := \{x\}$$

$$1$$

$$t := [2]$$

$$t := 1$$

```
<-- ERROR in psqrt/psqrt (now in psqrt) = _NOSQRT}
```

$$q := \_NOSQRT$$

$$\_NOSQRT$$

```
<-- exit psqrt (now in sqrt:-ModuleApply) = _NOSQRT}
```

$$r := \_NOSQRT$$

$$\sqrt{-x^2 + 1}$$

```
<-- exit sqrt:-ModuleApply (now at top level) = (-x^2+1)^(1/2)}
{--> enter int:-ModuleApply, args = x^3*exp(arcsin(x))/(-x^2+1)^
(1/2), x
{--> enter type/satisfies, args = exp(arcsin(x)), proc (f)
options operator, arrow; (op(0, f))::(Or(And(symbol, satisfies
(proc (f0) options operator, arrow; SearchText(%, f0, 1 .. 1) =
1 end proc)), And(indexed, satisfies(proc (f0) options operator,
arrow; (subsop(0 = op(0, f0), f))::'inertfunction' end proc))))
end proc
```

$$unknown := f \to (op(0, f))::(\mathrm{Or}(\mathrm{And}(symbol, satisfies(f0 \to SearchText("\%", f0, 1..1) = 1)), \mathrm{And}(indexed, satisfies(f0 \to (subsop(0 = op(0, f0), f))::'inertfunction'))))$$

```
{--> enter unknown, args = exp(arcsin(x))
```

$$\exp::(\mathrm{Or}(\mathrm{And}(symbol, satisfies(f0 \to SearchText("\%", f0, 1..1) = 1)), \mathrm{And}(indexed, satisfies(f0 \to (subsop(0 = op(0, f0), e^{\arcsin(x)}))::'inertfunction'))))$$

```
<-- exit unknown (now in type/satisfies) = exp::(Or(And(symbol,
```

```
satisfies(proc (f0) options operator, arrow; SearchText(%, f0, 1
.. 1) = 1 end proc)), And(indexed, satisfies(proc (f0) options
operator, arrow; (subsop(0 = op(0, f0), exp(arcsin(x))))
::'inertfunction' end proc))))}
```

$$answer := \mathrm{exp}::\left(\mathrm{Or}\left(\mathrm{And}\left(symbol, satisfies\left(f0 \to SearchText\left("\%", f0, 1..1\right) = 1\right)\right), \mathrm{And}\left(indexed, satisfies\left(f0 \to \left(subsop\left(0 = op\left(0, f0\right), e^{\arcsin(x)}\right)\right)::'inertfunction'\right)\right)\right)\right)$$

$$unknown := false$$

```
<-- exit type/satisfies (now in int:-ModuleApply) = false}
{--> enter type/satisfies, args = arcsin(x), proc (f) options
operator, arrow; (op(0, f))::(Or(And(symbol, satisfies(proc (f0)
options operator, arrow; SearchText(%, f0, 1 .. 1) = 1 end proc)
), And(indexed, satisfies(proc (f0) options operator, arrow;
(subsop(0 = op(0, f0), f))::'inertfunction' end proc)))) end
proc
```

$$unknown := f \to \left(op\left(0, f\right)\right)::\left(\mathrm{Or}\left(\mathrm{And}\left(symbol, satisfies\left(f0 \to SearchText\left("\%", f0, 1..1\right) = 1\right)\right), \mathrm{And}\left(indexed, satisfies\left(f0 \to \left(subsop\left(0 = op\left(0, f0\right), f\right)\right)::'inertfunction'\right)\right)\right)\right)$$

```
{--> enter unknown, args = arcsin(x)
```

$$\mathrm{arcsin}::\left(\mathrm{Or}\left(\mathrm{And}\left(symbol, satisfies\left(f0 \to SearchText\left("\%", f0, 1..1\right) = 1\right)\right), \mathrm{And}\left(indexed, satisfies\left(f0 \to \left(subsop\left(0 = op\left(0, f0\right), \arcsin\left(x\right)\right)\right)::'inertfunction'\right)\right)\right)\right)$$

```
<-- exit unknown (now in type/satisfies) = arcsin::(Or(And
(symbol, satisfies(proc (f0) options operator, arrow; SearchText
(%, f0, 1 .. 1) = 1 end proc)), And(indexed, satisfies(proc (f0)
options operator, arrow; (subsop(0 = op(0, f0), arcsin(x)))
::'inertfunction' end proc))))}
```

$$answer := \mathrm{arcsin}::\left(\mathrm{Or}\left(\mathrm{And}\left(symbol, satisfies\left(f0 \to SearchText\left("\%", f0, 1..1\right) = 1\right)\right), \mathrm{And}\left(indexed, satisfies\left(f0 \to \left(subsop\left(0 = op\left(0, f0\right), \arcsin\left(x\right)\right)\right)::'inertfunction'\right)\right)\right)\right)$$

$$unknown := false$$

```
<-- exit type/satisfies (now in int:-ModuleApply) = false}
{--> enter Main, args = x^3*exp(arcsin(x))/(-x^2+1)^(1/2), x
{--> enter Initialize, args =
<-- exit Initialize (now in Main) = }
{--> enter EnvToOptions, args = [x^3*exp(arcsin(x))/(-x^2+1)^
(1/2), x], [CauchyPrincipalValue = false, formula = true]
```

$$opts := \{formula = true, CauchyPrincipalValue = false\}$$

$$oargs := \left[\frac{x^3 e^{\arcsin(x)}}{\sqrt{-x^2+1}}, x\right]$$

$$oargs := \left[\frac{x^3 e^{\arcsin(x)}}{\sqrt{-x^2+1}}, x\right]$$

$$oargs := \left[\frac{x^3 e^{\arcsin(x)}}{\sqrt{-x^2+1}}, x\right]$$

$$envvar := CauchyPrincipalValue$$

$$envvar := AllSolutions$$

$$envvar := Continuous$$

$$opts := \{formula = true, CauchyPrincipalValue = false\}$$

```
<-- exit EnvToOptions (now in Main) = formula = true,
CauchyPrincipalValue = false}
{--> enter Exact, args = x^3*exp(arcsin(x))/(-x^2+1)^(1/2), x,
```

```
Main, formula = true, CauchyPrincipalValue = false
```

$$opts := \{CauchyPrincipalValue = false\}$$

$$envvar := CauchyPrincipalValue$$

$$_EnvCauchyPrincipalValue := false$$

$$envvar := AllSolutions$$

$$envvar := Continuous$$

$$tmp, backsubs := \left[\frac{x^3 \, e^{\arcsin(x)}}{\sqrt{-x^2+1}}, x \right], [\,]$$

$$f := \frac{x^3 \, e^{\arcsin(x)}}{\sqrt{-x^2+1}}$$

$$x := x$$

$$_Env\_z\_in\_use := \{\,\}$$

$$_EnvIntWarning := false$$

```
gcd/LinZip: Using 8-byte integer mod
gcd/LinZip: Using 8-byte integer mod
int/indef1: first-stage indefinite integration
int/indef2: second-stage indefinite integration
int/indef2: trying integration by parts
simplify/do: applying simplify/exp function to expression
combine: combining with respect to exp
int/rischnorm: enter Risch-Norman integrator
radfield: computing a basis for  {}
radfield: over  {}
radfield: options are {}
radfield: field is {}
radfield: extension degree is 1
radfield: computing a basis for  {}
radfield: over  {}
radfield: options are {}
radfield: field is {}
radfield: extension degree is 1
int/rischnorm: exit Risch-Norman integrator
int/risch: enter Risch integration
radfield: computing a basis for  {I}
radfield: over  {}
radfield: options are {}
radnormal/addtop: adding 1 algebraics
radnormal/addtop: over  {}
radnormal/addrads: adding 1 radicals
radnormal/addrads: over  {}
radnormal/addone: adding a single radical
radfield: field is {RootOf(_Z^2+1 index = 1)}
radfield: extension degree is 2
radfield: computing a basis for  {I}
radfield: over  {}
radfield: options are {}
radfield: field is {RootOf(_Z^2+1 index = 1)}
radfield: extension degree is 2
simplify/do: applying simplify/radical function to expression
simplify/do: applying simplify/sqrt function to expression
sqrfree/Yun: square-free factorization in x
```

```
sqrfree/Yun: square-free factorization in x
sqrfree/Yun: square-free factorization in x
simplify/do: applying simplify/power function to expression
simplify/do: applying simplify/power function to expression
```

$$answer := \int \frac{x^3 \, e^{\arcsin(x)}}{\sqrt{-x^2+1}} \, dx$$

$$answer := \int \frac{x^3 \, e^{\arcsin(x)}}{\sqrt{-x^2+1}} \, dx$$

```
<-- exit Exact (now in Main) = int(x^3*exp(arcsin(x))/(-x^2+1)^
(1/2), x)}
<-- exit Main (now in int:-ModuleApply) = int(x^3*exp(arcsin(x))
/(-x^2+1)^(1/2), x)}
<-- exit int:-ModuleApply (now at top level) = int(x^3*exp
(arcsin(x))/(-x^2+1)^(1/2), x)}
```

$$\int \frac{x^3 \, e^{\arcsin(x)}}{\sqrt{-x^2+1}} \, dx \tag{1}$$