

CADNA for C/C++ source codes

Reference manual

Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie - Paris 6
Paris, France
<http://www.lip6.fr/cadna>
cadna-team@lip6.fr



Contents

1	The operators	5
1.1	cadna_add/operator+	5
1.2	cadna_sub/operator-	6
1.3	cadna_mul/operator	6
1.4	cadna_div/operator/	7
1.5	cadna_ne/operator!=	7
1.6	cadna_ge/operator>=	8
1.7	cadna_gt/operator>	9
1.8	cadna_le/operator<=	9
1.9	cadna_lt/operator<	10
2	The mathematical functions	11
2.1	cadna_math/acos	11
2.2	cadna_math/acosh	12
2.3	cadna_math/acoshf	12
2.4	cadna_math/acoshf	13
2.5	cadna_math/acoshf	14
2.6	cadna_math/acoshf	14
2.7	cadna_math/asin	15
2.8	cadna_math/asinh	16
2.9	cadna_math/asinhf	16
2.10	cadna_math/asinhf	17
2.11	cadna_math/asinhf	18
2.12	cadna_math/asinhf	18
2.13	cadna_math/atan	19
2.14	cadna_math/atan2	20
2.15	cadna_math/atanf	20
2.16	cadna_math/atanf	21
2.17	cadna_math/atanh	22

2.18	cadna_math/atanhf	22
2.19	cadna_math/atanhf	23
2.20	cadna_math/cbrt	24
2.21	cadna_math/cbrt	24
2.22	cadna_math/cbrtf	25
2.23	cadna_math/cos	25
2.24	cadna_math/cos	26
2.25	cadna_math/cosf	26
2.26	cadna_math/cosh	27
2.27	cadna_math/coshf	27
2.28	cadna_math/coshf	28
2.29	cadna_math/exp	28
2.30	cadna_math/exp	29
2.31	cadna_math/exp2	29
2.32	cadna_math/exp2	30
2.33	cadna_math/exp2f	30
2.34	cadna_math/expf	31
2.35	cadna_math/expm1	31
2.36	cadna_math/expm1	32
2.37	cadna_math/expm1f	32
2.38	cadna_math/finite	33
2.39	cadna_math/finite	33
2.40	cadna_math/fmax	34
2.41	cadna_math/fmaxf	34
2.42	cadna_math/fmin	35
2.43	cadna_math/fminf	35
2.44	cadna_math/fmod	36
2.45	cadna_math/fmod	36
2.46	cadna_math/fmodf	37
2.47	cadna_math/frexp	38
2.48	cadna_math/frexp	38
2.49	cadna_math/frexp	39
2.50	cadna_math/hypot	40
2.51	cadna_math/hypotf	40
2.52	cadna_math/isfinite	41
2.53	cadna_math/isfinite	42
2.54	cadna_math/isinf	42
2.55	cadna_math/isinf	43
2.56	cadna_math/isnan	43
2.57	cadna_math/isnan	44

2.58	cadna_math/ldexp	44
2.59	cadna_math/ldexp	45
2.60	cadna_math/ldexpf	45
2.61	cadna_math/log	46
2.62	cadna_math/log	47
2.63	cadna_math/log	47
2.64	cadna_math/log10	48
2.65	cadna_math/log10	48
2.66	cadna_math/log10f	49
2.67	cadna_math/log1p	49
2.68	cadna_math/log1p	50
2.69	cadna_math/log1pf	50
2.70	cadna_math/log2	51
2.71	cadna_math/log2	51
2.72	cadna_math/log2	52
2.73	cadna_math/logb	52
2.74	cadna_math/logb	53
2.75	cadna_math/logbf	53
2.76	cadna_math/modf	54
2.77	cadna_math/modf	54
2.78	cadna_math/modff	55
2.79	cadna_math/pow	55
2.80	cadna_math/powf	56
2.81	cadna_math/sin	56
2.82	cadna_math/sinf	57
2.83	cadna_math/sinf	57
2.84	cadna_math/sinh	58
2.85	cadna_math/sinhf	58
2.86	cadna_math/sinhf	59
2.87	cadna_math/sqrt	59
2.88	cadna_math/sqrt	60
2.89	cadna_math/sqrtf	60
2.90	cadna_math/tan	61
2.91	cadna_math/tanf	61
2.92	cadna_math/tanf	62
2.93	cadna_math/tanh	62
2.94	cadna_math/tanhf	63
2.95	cadna_math/tanhf	63

3	The intrinsic functions	65
3.1	cadna_intr/ceil	65
3.2	cadna_intr/ceilf	66
3.3	cadna_intr/fabs	66
3.4	cadna_intr/fabsf	67
3.5	cadna_intr/floor	67
3.6	cadna_intr/floorf	68
3.7	cadna_intr/llrint	68
3.8	cadna_intr/llrintf	69
3.9	cadna_intr/lrint	70
3.10	cadna_intr/lrintf	70
3.11	cadna_intr/nearbyint	71
3.12	cadna_intr/nearbyintf	72
3.13	cadna_intr/rint	72
3.14	cadna_intr/rintf	73
3.15	cadna_intr/trunc	74
3.16	cadna_intr/truncf	74
4	I/O functions	75
4.1	cadna_str/display	75
4.2	cadna_str/operator<<	75
4.3	cadna_str/operator>>	76
4.4	cadna_str/str	77
4.5	cadna_str/strp	77
5	Cadna specific functions	79
5.1	cadna_type/cadna_disable	79
5.2	cadna_type/cadna_enable	79
5.3	cadna_type/cadna_end	80
5.4	cadna_type/cadna_init	81
5.5	cadna_digitnumber/approx_digit	81
5.6	cadna_digitnumber/nb_significant_digit	82
5.7	cadna_computedzero/computedzero	82
5.8	cadna_numericalnoise/numericalnoise	83
6	Conversion functions	85
6.1	cadna_convert/char	85
6.2	cadna_convert/data_st	85
6.3	cadna_convert/double	86
6.4	cadna_convert/float	86

6.5	<code>cadna_convert/int</code>	87
6.6	<code>cadna_convert/long</code>	87
6.7	<code>cadna_convert/long long</code>	88
6.8	<code>cadna_convert/short</code>	88
6.9	<code>cadna_convert/unsigned char</code>	88
6.10	<code>cadna_convert/unsigned int</code>	89
6.11	<code>cadna_convert/unsigned long</code>	89
6.12	<code>cadna_convert/unsigned long long</code>	90
6.13	<code>cadna_convert/unsigned short</code>	90
6.14	<code>cadna_to/operator=</code>	91
7	Expert functions	93
7.1	<code>cadna_type/cadna_set_rnd_arr()</code>	93
7.2	<code>cadna_type/cadna_set_rnd_moinf()</code>	93
7.3	<code>cadna_type/cadna_set_rnd_plinf()</code>	94
7.4	<code>cadna_type/cadna_set_rnd_zero()</code>	94
8	Internal functions	95
8.1	<code>cadna_unstab/instability</code>	95

Chapter 1

The operators

1.1 `cadna.add/operator+`

[Methods]

NAME:

`operator+`

SYNOPSIS:

`res = a + b`

FUNCTION:

Defines all the functions involving at least one stochastic argument which overload the "+" operator in a statement such as "a+b" or "+a".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - a stochastic number

1.2 cadna_sub/operator-

[*Methods*]

NAME:

operator-

SYNOPSIS:

res = a - b

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the "-" operator in a statement such as "a-b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - a stochastic number

1.3 cadna_mul/operator

[*Methods*]

NAME:

operator*

SYNOPSIS:

res = a * b

FUNCTION:

Defines all the functions involving at least one argument of stochastic type which overload the "*" operator in a statement such as "a*b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - a stochastic number

1.4 `cadna_div/operator/`

[Methods]

NAME:

`operator/`

SYNOPSIS:

`res = a / b`

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the "/" operator in a statement such as "a/b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - a stochastic number

1.5 `cadna_ne/operator!=`

[Methods]

NAME:

`operator!=`

SYNOPSIS:

```
res = a != b
```

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the "!=" operator.

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - an integer value

1.6 cadna_ge/operator>=

[*Methods*]

NAME:

operator>=

SYNOPSIS:

```
res = a >= b
```

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the ">=" operator in a test such as "a>=b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - an integer value

1.7 `cadna_gt/operator>`

[*Methods*]

NAME:

`operator>`

SYNOPSIS:

`res = a > b`

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the ">" operator in a test such as "a>b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - an integer value

1.8 `cadna_le/operator<=`

[*Methods*]

NAME:

`operator<=`

SYNOPSIS:

`res = a <= b`

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the "<=" operator in a test such as "a<=b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - an integer value

1.9 cadna_lt/operator<

[*Methods*]

NAME:

operator<

SYNOPSIS:

res = a < b

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the "<" operator in a test such as "a<b".

INPUTS:

a - an integer, a float, a double or a stochastic number
b - an integer, a float, a double or a stochastic number
At least one argument must be of stochastic type.

RESULT:

res - an integer value

Chapter 2

The mathematical functions

2.1 `cadna_math/acos`

[Functions]

NAME:

`acos`

SYNOPSIS:

`res = acos(x)`

FUNCTION:

The `acos()` function computes the principal value of the arc cosine of `x`. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

`x` - `double_st`

RESULT:

`res` - `double_st`

SEE ALSO:

`acos(3)`, `asin(3)`, `atan(3)`, `atan2(3)`, `cos(3)`, `cosh(3)`, `sinh(3)`,
`tan(3)`, `tanh(3)`

2.2 cadna_math/acosf

[*Functions*]

NAME:

acos

SYNOPSIS:

```
res = acos(x)
```

FUNCTION:

The acos() function computes the principal value of the arc cosine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.3 cadna_math/acosf

[*Functions*]

NAME:

acosf

SYNOPSIS:

```
res = acosf(x)
```

FUNCTION:

The acos() function computes the principal value of the arc cosine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.4 cadna_math/acosh

[*Functions*]

NAME:

acosh

SYNOPSIS:

res = acosh(x)

FUNCTION:

The acosh() function computes the inverse hyperbolic cosine of the double_st argument

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.5 cadna_math/acoshf

[*Functions*]

NAME:

acoshf

SYNOPSIS:

res = acoshf(x)

FUNCTION:

The acoshf() function computes the inverse hyperbolic cose of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.6 cadna_math/acosh

[*Functions*]

NAME:

acosh

SYNOPSIS:

res = acosh(x)

FUNCTION:

The acosh() function computes the inverse hyperbolic cose of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.7 cadna_math/asin

[*Functions*]

NAME:

asin

SYNOPSIS:

res = asin(x)

FUNCTION:

The asin() function computes the principal value of the arc sine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.8 cadna_math/asin

[*Functions*]

NAME:

asin

SYNOPSIS:

res = asin(x)

FUNCTION:

The asin() function computes the principal value of the arc sine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.9 cadna_math/asinf

[*Functions*]

NAME:

asinf

SYNOPSIS:

res = asinf(x)

FUNCTION:

The asin() function computes the principal value of the arc sine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.10 cadna_math/asinh

[*Functions*]

NAME:

asinh

SYNOPSIS:

res = asinh(x)

FUNCTION:

The asinh() function computes the inverse hyperbolic sine of the double_st argument

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.11 cadna_math/asinhf

[*Functions*]

NAME:

asinh

SYNOPSIS:

res = asinh(x)

FUNCTION:

The asinh() function computes the inverse hyperbolic sine of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.12 cadna_math/asinhf

[*Functions*]

NAME:

asinhf

SYNOPSIS:

res = asinhf(x)

FUNCTION:

The asinhf() function computes the inverse hyperbolic sine of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.13 cadna_math/atan

[*Functions*]

NAME:

atan

SYNOPSIS:

res = atan(x)

FUNCTION:

The atan() function computes the principal value of the arc tangent of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.14 cadna_math/atan2

[*Functions*]

NAME:

atan2

SYNOPSIS:

res = atan2(x,y)

FUNCTION:

The atan2() function computes the principal value of the arc tangent of y/x, using the signs of both arguments to determine the quadrant of the return value.

INPUTS:

x	- double_st
y	- double_st

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

xcos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.15 cadna_math/atanf

[*Functions*]

NAME:

atanf

SYNOPSIS:

res = atanf(x)

FUNCTION:

The atan() function computes the principal value of the arc sine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.16 cadna_math/atanf

[*Functions*]

NAME:

atan

SYNOPSIS:

res = atan(x)

FUNCTION:

The atan() function computes the principal value of the arc
sine of x. The result is in the range $[-\pi/2, +\pi/2]$.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.17 cadna_math/atanh

[*Functions*]

NAME:

atanh

SYNOPSIS:

res = atanh(x)

FUNCTION:

The atanh() function computes the inverse hyperbolic tangent of the double_st argument

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.18 cadna_math/atanhf

[*Functions*]

NAME:

atanh

SYNOPSIS:

res = atanh(x)

FUNCTION:

The atanh() function computes the inverse hyperbolic tangent of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.19 cadna_math/atanhf

[*Functions*]

NAME:

atanhf

SYNOPSIS:

res = atanhf(x)

FUNCTION:

The atanhf() function computes the inverse hyperbolic tangent of the float_st argument

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.20 cadna_math/cbrt

[*Functions*]

NAME:

cbrt

SYNOPSIS:

res = cbrt(x)

FUNCTION:

The cbrt() function computes the cube root of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

sqrt(3)

2.21 cadna_math/cbrt

[*Functions*]

NAME:

cbrt

SYNOPSIS:

res = cbrt(x)

FUNCTION:

The cbrt() function computes the cube root of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

sqrt(3)

2.22 cadna_math/cbrtf

[*Functions*]

NAME:

cbrtf

SYNOPSIS:

res = cbrtf(x)

FUNCTION:

The cbrt() function computes the cube root of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

sqrt(3)

2.23 cadna_math/cos

[*Functions*]

NAME:

cos

SYNOPSIS:

res = cos(x)

FUNCTION:

The cosf() function computes the cosine of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.24 cadna_math/cos

[*Functions*]

NAME:

cos

SYNOPSIS:

res = cos(x)

FUNCTION:

The cos() function computes the cosine of x (measured in radians).

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.25 cadna_math/cosf

[*Functions*]

NAME:

cosf

SYNOPSIS:

res = cosf(x)

FUNCTION:

The cos() function computes the cosine of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.26 cadna_math/cosh

[Functions]

NAME:

cosh

SYNOPSIS:

res = cosh(x)

FUNCTION:

The cosh() function computes the hyperbolic cosine of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.27 cadna_math/coshf

[Functions]

NAME:

cosh

SYNOPSIS:

res = cosh(x)

FUNCTION:

The cosh() function computes the hyperbolic cosine of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.28 cadna_math/coshf

[*Functions*]

NAME:

coshf

SYNOPSIS:

res = coshf(x)

FUNCTION:

The coshf() function computes the hyperbolic cosine of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.29 cadna_math/exp

[*Functions*]

NAME:

exp

SYNOPSIS:

res = exp(x)

FUNCTION:

The exp() function computes e**x, the base-e exponential of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.30 cadna_math/exp

[Functions]

NAME:

exp

SYNOPSIS:

res = exp(x)

FUNCTION:

The exp() function computes e**x, the base-e exponential of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p() , exp(3), exp2(3), expm1(3), pow(3)

2.31 cadna_math/exp2

[Functions]

NAME:

exp2

SYNOPSIS:

res = exp2(x)

FUNCTION:

The exp2() function computes 2**x, the base-2 exponential of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p() , exp(3), exp2(3), expm1(3), pow(3)

2.32 cadna_math/exp2

[*Functions*]

NAME:

exp2

SYNOPSIS:

res = exp2(x)

FUNCTION:

The exp() function computes 2**x, the base-e exponential of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.33 cadna_math/exp2f

[*Functions*]

NAME:

exp2f

SYNOPSIS:

res = exp2f(x)

FUNCTION:

The expf() function computes 2**x, the base-e exponential of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.34 cadna_math/expf

[Functions]

NAME:

expf

SYNOPSIS:

res = expf(x)

FUNCTION:

The expf() function computes e^{**x} , the base-e exponential of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.35 cadna_math/expm1

[Functions]

NAME:

expm1

SYNOPSIS:

res = expm1(x)

FUNCTION:

The expm1() function computes the base-e exponential of x ,
minus 1 accurately even for very small values of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.36 cadna_math/expm1

[*Functions*]

NAME:

expm1

SYNOPSIS:

res = expm1(x)

FUNCTION:

The expm1() function computes the base-e exponential of x ,
minus 1 accurately even for very small values of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.37 cadna_math/expm1f

[*Functions*]

NAME:

expm1f

SYNOPSIS:

res = expm1f(x)

FUNCTION:

The expm1() function computes the base-e exponential of x ,
minus 1 accurately even for very small values of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.38 cadna_math/finite

[Functions]

NAME:

finite

SYNOPSIS:

res = finite(a)

FUNCTION:

The finite() functions return a nonzero value if x is neither infinite nor a "not-a-number" (NaN) value, and 0 otherwise.

INPUTS:

a - float_st

RESULT:

res - int

SEE ALSO:

isinf(3), isnan(3)

2.39 cadna_math/finite

[Functions]

NAME:

finite

SYNOPSIS:

res = finite(a)

FUNCTION:

The finite() functions return a nonzero value if x is neither infinite nor a "not-a-number" (NaN) value, and 0 otherwise.

INPUTS:

a - double_st

RESULT:

res - int

SEE ALSO:

isinf(3), isnan(3)

2.40 cadna_math/fmax

[*Functions*]

NAME:

fmax

SYNOPSIS:

res = fmax(x,y)

FUNCTION:

The fmax() functions return x or y, whichever is larger.

INPUTS:

a	- double_st
b	- double_st

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

fmin(3)

2.41 cadna_math/fmaxf

[*Functions*]

NAME:

fmaxf

SYNOPSIS:

res = fmaxf(x,y)

FUNCTION:

The fmaxf() functions return x or y, whichever is larger.

INPUTS:

x	- float_st
b	- float_st

RESULT:

res	- float_st
-----	------------

SEE ALSO:

fminf(3)

2.42 cadna_math/fmin

[Functions]

NAME:

fmin

SYNOPSIS:

res = fmin(x,y)

FUNCTION:

The fmin() functions return x or y, whichever is smaller.

INPUTS:

a - double_st

b - double_st

RESULT:

res - double_st

SEE ALSO:

fmax(3)

2.43 cadna_math/fminf

[Functions]

NAME:

fminf

SYNOPSIS:

res = fminf(x,y)

FUNCTION:

The fminf() functions return x or y, whichever is smaller.

INPUTS:

x - float_st

b - float_st

RESULT:

res - float_st

SEE ALSO:

fmaxf(3)

2.44 cadna_math/fmod

[*Functions*]

NAME:

fmod

SYNOPSIS:

res = fmod(a,b)

FUNCTION:

The fmod() function computes the floating-point remainder of dividing
The return value is $x - n * y$, where n is the quotient of x / y , rounded to
zero to an integer.

INPUTS:

a	- double_st
b	- double_st

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

remainder(3)

2.45 cadna_math/fmod

[*Functions*]

NAME:

fmod

SYNOPSIS:

res = fmod(a,b)

FUNCTION:

The fmod() function computes the floating-point remainder of dividing
The return value is $x - n * y$, where n is the quotient of x / y , rounded to
zero to an integer.

INPUTS:

a - float_st
b - float_st

RESULT:

res - float_st

SEE ALSO:

remainder(3)

2.46 `cadna_math/fmodf`

[*Functions*]

NAME:

fmodf

SYNOPSIS:

res = fmodf(a,b)

FUNCTION:

The `fmodf()` function computes the floating-point remainder of dividing `x` by `y`. The return value is `x - n * y`, where `n` is the quotient of `x / y`, rounded toward zero to an integer.

INPUTS:

a - float_st
b - float_st

RESULT:

res - float_st

SEE ALSO:

remainder(3)

2.47 cadna_math/frexp

[*Functions*]

NAME:

frexp

SYNOPSIS:

res = frexp(x,n)

FUNCTION:

The frexp() function is used to split the number x into a normalized fraction and an exponent which is stored in exp.

INPUTS:

x	- double_st
n	- int*

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

ldexp(3)

2.48 cadna_math/frexp

[*Functions*]

NAME:

frexp

SYNOPSIS:

res = frexpf(x,n)

FUNCTION:

The frexp() function is used to split the number x into a normalized fraction and an exponent which is stored in exp.

INPUTS:

x	- float_st
n	- int*

RESULT:

res	- float_st
-----	------------

SEE ALSO:

ldexp(3)

2.49 cadna_math/frexp

[*Functions*]

NAME:

frexp

SYNOPSIS:

res = frexp(x,n)

FUNCTION:

The `frexp()` function is used to split the number `x` into a normalized fraction and an exponent which is stored in `exp`.

INPUTS:

x	- float_st
n	- int*

RESULT:

res	- float_st
-----	------------

SEE ALSO:

ldexp(3)

2.50 cadna_math/hypot

[*Functions*]

NAME:

hypot

SYNOPSIS:

res = hypot(x,y)

FUNCTION:

The hypot() function computes the $\sqrt{x^2+y^2}$ without undue overflow or underflow.

INPUTS:

a	- double_st
b	- double_st

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

sqrtf(3)

2.51 cadna_math/hypotf

[*Functions*]

NAME:

hypotf

SYNOPSIS:

res = hypotf(x,y)

FUNCTION:

The hypotf() function computes the $\sqrt{x^2+y^2}$ without undue overflow or underflow.

SEE ALSO:

```
sqrtf(3)
```

INPUTS:

```
x          - float_st
b          - float_st
```

RESULT:

```
res        - float_st
```

2.52 cadna_math/isfinite

[*Functions*]

NAME:

```
isfinite
```

SYNOPSIS:

```
res = isfinite(a)
```

FUNCTION:

The finite() functions return a nonzero value if x is neither infinite nor a "not-a-number" (NaN) value, and 0 otherwise.

INPUTS:

```
a          - double_st
```

RESULT:

```
res        - int
```

SEE ALSO:

```
isinf(3), isnan(3)
```

2.53 cadna_math/isfinite

[*Functions*]

NAME:

isfinite

SYNOPSIS:

res = isfinitef(a)

FUNCTION:

The finite() functions return a nonzero value if x is neither infinite nor a "not-a-number" (NaN) value, and 0 otherwise.

INPUTS:

a - float_st

RESULT:

res - int

SEE ALSO:

isinf(3), isnan(3)

2.54 cadna_math/isinf

[*Functions*]

NAME:

isinf

SYNOPSIS:

res = isinf(a)

FUNCTION:

The isinf() functions return 1 if x is positive infinity, -1 if x is negative infinity, and 0 otherwise.

INPUTS:

a - float_st

RESULT:

res - int

SEE ALSO:

finite(3), isinf(3)

2.55 cadna_math/isinf

[Functions]

NAME:

isinf

SYNOPSIS:

res = isinf(a)

FUNCTION:

The isinf() functions return 1 if x is positive infinity, -1 if x is negative infinity.

INPUTS:

a - double_st

RESULT:

res - int

SEE ALSO:

finite(3), isinf(3)

2.56 cadna_math/isnan

[Functions]

NAME:

isnan

SYNOPSIS:

res = isnan(a)

FUNCTION:

The isnanf() functions return a nonzero value if x is a NaN value, and 0 otherwise.

INPUTS:

a - double_st

RESULT:

res - int

SEE ALSO:

finite(3), isinf(3)

2.57 cadna_math/isnan

[*Functions*]

NAME:

isnan

SYNOPSIS:

res = isnan(a)

FUNCTION:

The isnan() functions return a nonzero value if x is

INPUTS:

a - float_st

RESULT:

res - int

SEE ALSO:

finite(3), isinf(3)

2.58 cadna_math/ldexp

[*Functions*]

NAME:

ldexp

SYNOPSIS:

res = ldexp(x,n)

FUNCTION:

The ldexp() function returns the result of multiplying the floating-point number x by 2 raised to the power exp.

INPUTS:

x - float_st
n - int

RESULT:

res - float_st

SEE ALSO:

frexp(3), modf(3), scalbln(3)

2.59 cadna_math/ldexp

[Functions]

NAME:

ldexp

SYNOPSIS:

```
res = ldexp(x,n)
```

FUNCTION:

The ldexp() function returns the result of multiplying the floating-point number x by 2 raised to the power exp.

INPUTS:

x	- double_st
n	- int

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

frexp(3), modf(3), scalbln(3)

2.60 cadna_math/ldexpf

[Functions]

NAME:

ldexpf

SYNOPSIS:

```
res = ldexpf(x,n)
```

FUNCTION:

The ldexpf() function returns the result of multiplying the floating-point number x by 2 raised to the power exp.

INPUTS:

x - float_st
n - int

RESULT:

res - float_st

SEE ALSO:

frexpf(3), modf(3), scalbln(3)

2.61 cadna_math/log

[*Functions*]

NAME:

log

SYNOPSIS:

res = log(x)

FUNCTION:

The log() function computes the value of the natural logarithm of argument x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.62 cadna_math/log

[Functions]

NAME:

log

SYNOPSIS:

res = log(x)

FUNCTION:

The log() function computes the value of the natural logarithm of argument x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p() , exp(3) , exp2(3) , expm1(3) , pow(3)

2.63 cadna_math/log

[Functions]

NAME:

log

SYNOPSIS:

res = log(x)

FUNCTION:

The log() function computes the value of the natural logarithm of argument x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p() , exp(3) , exp2(3) , expm1(3) , pow(3)

2.64 cadna_math/log10

[*Functions*]

NAME:

log10

SYNOPSIS:

res = log10(x)

FUNCTION:

The log10() function computes the value of argument x to base 10.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.65 cadna_math/log10

[*Functions*]

NAME:

log10

SYNOPSIS:

res = log10(x)

FUNCTION:

The log10() function computes the value of argument x to base 10.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.66 cadna_math/log10f

[Functions]

NAME:

log10f

SYNOPSIS:

res = log10f(x)

FUNCTION:

The log10f() function computes the value of argument x to base 10.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.67 cadna_math/log1p

[Functions]

NAME:

log1p

SYNOPSIS:

res = log1p(x)

FUNCTION:

The log1p() function computes the value of $\log(1+x)$ accurately even for very small values of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.68 cadna_math/log1p

[*Functions*]

NAME:

log1p

SYNOPSIS:

res = log1p(x)

FUNCTION:

The log1p() function computes the value of $\log(1+x)$ accurately even for very small values of x .

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.69 cadna_math/log1pf

[*Functions*]

NAME:

log1pf

SYNOPSIS:

res = log1pf(x)

FUNCTION:

The log1pf() function computes the value of $\log(1+x)$ accurately even for very small values of x .

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.70 cadna_math/log2

[Functions]

NAME:

log2

SYNOPSIS:

res = log2(x)

FUNCTION:

The log() function computes the value of argument x to base 2.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.71 cadna_math/log2

[Functions]

NAME:

log2

SYNOPSIS:

res = log2(x)

FUNCTION:

The log() function computes the value of argument x to base 2.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.72 cadna_math/log2

[*Functions*]

NAME:

log2

SYNOPSIS:

res = log2(x)

FUNCTION:

The log() function computes the value of argument x to base 2.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.73 cadna_math/logb

[*Functions*]

NAME:

logb

SYNOPSIS:

res = logb(x)

FUNCTION:

The logb() functions return the exponent of x, represented as a floating-point number.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

log2() , log10() , log1p(), exp(3), exp2(3), expm1(3), pow(3)

2.74 cadna_math/logb

[Functions]

NAME:

logb

SYNOPSIS:

res = logb(x)

FUNCTION:

The logb() functions return the exponent of x, represented as a floating-point number.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p() , exp(3) , exp2(3) , expm1(3) , pow(3)

2.75 cadna_math/logbf

[Functions]

NAME:

logbf

SYNOPSIS:

res = logbf(x)

FUNCTION:

The logbf() functions return the exponent of x, represented as a floating-point number.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

log2() , log10() , log1p() , exp(3) , exp2(3) , expm1(3) , pow(3)

2.76 cadna_math/modf

[*Functions*]

NAME:

modf

SYNOPSIS:

res = modff(x,iptr)

FUNCTION:

The modf() function break value into integral and fractional parts

INPUTS:

x	- float_st
iptr	- float_st*

RESULT:

res	- float_st
-----	------------

SEE ALSO:

frexp(3), ldexp(3), math(3)

2.77 cadna_math/modf

[*Functions*]

NAME:

modf

SYNOPSIS:

res = modf(x,iptr)

FUNCTION:

The modf() function break value into integral and fractional parts

INPUTS:

x	- double_st
iptr	- double_st *

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

frexp(3), ldexp(3), math(3)

2.78 cadna_math/modff

[Functions]

NAME:

modff

SYNOPSIS:

```
res = modff(x,iptr)
```

FUNCTION:

The modf() function break value into integral and fractional parts

INPUTS:

x	- float_st
iptr	- float_st*

RESULT:

res	- float_st
-----	------------

SEE ALSO:

frexp(3), ldexp(3), math(3)

2.79 cadna_math/pow

[Functions]

NAME:

pow

SYNOPSIS:

```
res = pow(x,y)
```

FUNCTION:

The pow() functions compute x raised to the power y.

INPUTS:

a	- double_st
b	- double_st

RESULT:

res	- double_st
-----	-------------

2.80 cadna_math/powf

[*Functions*]

NAME:

powf

SYNOPSIS:

res = powf(x,y)

FUNCTION:

The powf() functions compute x raised to the power y.

INPUTS:

a	- float_st
b	- float_st

RESULT:

res	- float_st
-----	------------

2.81 cadna_math/sin

[*Functions*]

NAME:

sin

SYNOPSIS:

res = sin(x)

FUNCTION:

The sin() function computes the sine of x (measured in radians).

INPUTS:

x	- double_st
---	-------------

RESULT:

res	- double_st
-----	-------------

SEE ALSO:

asin(3), asin(3), atan(3), atan2(3), sin(3), sinh(3), sinh(3),
tan(3), tanh(3)

2.82 cadna_math/sinf

[Functions]

NAME:

sin

SYNOPSIS:

res = sin(x)

FUNCTION:

The sin() function computes the sine of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

asinf(3), asinf(3), atanf(3), atan2f(3), sinf(3), sinh(3),
sinh(3), tanf(3), tanh(3)

2.83 cadna_math/sinf

[Functions]

NAME:

sinf

SYNOPSIS:

res = sinf(x)

FUNCTION:

The sin() function computes the sine of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

asinf(3), asinf(3), atanf(3), atan2f(3), sinf(3), sinh(3),
sinh(3), tanf(3), tanh(3)

2.84 `cadna_math/sinh`

[*Functions*]

NAME:

`sinh`

SYNOPSIS:

`res = sinh(x)`

FUNCTION:

The `sinh()` function computes the hyperbolic sine of `x`.

INPUTS:

`x` - `double_st`

RESULT:

`res` - `double_st`

SEE ALSO:

`acos(3)`, `asin(3)`, `atan(3)`, `atan2(3)`, `cos(3)`, `cosh(3)`, `sinh(3)`,
`tan(3)`, `tanh(3)`

2.85 `cadna_math/sinhf`

[*Functions*]

NAME:

`sinhf`

SYNOPSIS:

`res = sinhf(x)`

FUNCTION:

The `sinhf()` function computes the hyperbolic sine of `x`.

INPUTS:

`x` - `float_st`

RESULT:

`res` - `float_st`

SEE ALSO:

`acosf(3)`, `asinf(3)`, `atanf(3)`, `atan2f(3)`, `cosf(3)`, `coshf(3)`,
`sinhf(3)`, `tanf(3)`, `tanhf(3)`

2.86 cadna_math/sinhf

[Functions]

NAME:

sinh

SYNOPSIS:

res = sinh(x)

FUNCTION:

The sinh() function computes the hyperbolic sine of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.87 cadna_math/sqrt

[Functions]

NAME:

sqrt

SYNOPSIS:

res = sqrt(x)

FUNCTION:

The sqrt() function compute the non-negative square root of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

cbrt()

2.88 `cadna_math/sqrt`

[*Functions*]

NAME:

`sqrt`

SYNOPSIS:

`res = sqrt(x)`

FUNCTION:

The `sqrt()` function compute the non-negative square root of `x`.

INPUTS:

`x` - `double_st`

RESULT:

`res` - `double_st`

SEE ALSO:

`cbrt()`

2.89 `cadna_math/sqrtf`

[*Functions*]

NAME:

`sqrtf`

SYNOPSIS:

`res = sqrtf(x)`

FUNCTION:

The `sqrtf()` function compute the non-negative square root of `x`.

INPUTS:

`x` - `float_st`

RESULT:

`res` - `float_st`

SEE ALSO:

`cbrt()`

2.90 cadna_math/tan

[Functions]

NAME:

tan

SYNOPSIS:

res = tan(x)

FUNCTION:

The tan() function computes the tangent of x (measured in radians).

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

atan(3), atan(3), atan(3), atan2(3), tan(3), tanh(3), tanh(3),
tan(3), tanh(3)

2.91 cadna_math/tanf

[Functions]

NAME:

tanf

SYNOPSIS:

res = tanf(x)

FUNCTION:

The tanf() function computes the tangent of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

atanf(3), atanf(3), atanf(3), atan2f(3), tanf(3), tanhf(3),
tanhf(3), tanf(3), tanhf(3)

2.92 cadna_math/tanf

[*Functions*]

NAME:

tan

SYNOPSIS:

res = tan(x)

FUNCTION:

The tan() function computes the tangent of x (measured in radians).

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

atanf(3), atanf(3), atanf(3), atan2f(3), tanf(3), tanhf(3),
tanhf(3), tanf(3), tanhf(3)

2.93 cadna_math/tanh

[*Functions*]

NAME:

tanh

SYNOPSIS:

res = tanh(x)

FUNCTION:

The tanh() function computes the hyperbolic tangent of x.

INPUTS:

x - double_st

RESULT:

res - double_st

SEE ALSO:

acos(3), asin(3), atan(3), atan2(3), cos(3), cosh(3), sinh(3),
tan(3), tanh(3)

2.94 cadna_math/tanhf

[Functions]

NAME:

tanhf

SYNOPSIS:

res = tanhf(x)

FUNCTION:

The tanhf() function computes the hyperbolic tangent of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

2.95 cadna_math/tanh

[Functions]

NAME:

tanh

SYNOPSIS:

res = tanh(x)

FUNCTION:

The tanh() function computes the hyperbolic tangent of x.

INPUTS:

x - float_st

RESULT:

res - float_st

SEE ALSO:

acosf(3), asinf(3), atanf(3), atan2f(3), cosf(3), coshf(3),
sinhf(3), tanf(3), tanhf(3)

Chapter 3

The intrinsic functions

3.1 `cadna_intr/ceil`

[*Functions*]

NAME:

`ceil`

SYNOPSIS:

`res = ceil(x)`

FUNCTION:

The `ceil()` functions return the smallest integral value greater than or equal to `x`.

INPUTS:

`a` - `double_st`

RESULT:

`res` - `double_st`

SEE ALSO:

`abs(3)`, `ceil(3)`, `floor(3)`, `rint(3)`

3.2 cadna_intr/ceilf

[*Functions*]

NAME:

ceilf

SYNOPSIS:

res = ceilf(x)

FUNCTION:

The ceilf() functions return the smallest integral value greater than or equal to x.

INPUTS:

a - float_st

RESULT:

res - float_st

SEE ALSO:

absf(3), ceilf(3), floorf(3), rintf(3)

3.3 cadna_intr/fabs

[*Functions*]

NAME:

fabs

SYNOPSIS:

res = fabs(x)

FUNCTION:

The fabs() functions compute the absolute value of a stochastic number x.

INPUTS:

a - double_st

RESULT:

res - double_st

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.4 cadna_intr/fabsf

[Functions]

NAME:

fabsf

SYNOPSIS:

```
res = fabsf(x)
```

FUNCTION:

The fabsf() functions compute the absolute value of a stochastic number x.

INPUTS:

a - float_st

RESULT:

res - float_st

SEE ALSO:

ceilf(3), floorf(3), rintf(3)

3.5 cadna_intr/floor

[Functions]

NAME:

floor

SYNOPSIS:

```
res = floor(x)
```

FUNCTION:

The floor() functions return the largest integral value less than or equal to x.

INPUTS:

a - double_st

RESULT:

res - double_st

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.6 cadna_intr/floorf

[*Functions*]

NAME:

floorf

SYNOPSIS:

res = floorf(x)

FUNCTION:

The floorf() functions return the largest integral value less than or equal to x.

INPUTS:

a - float_st

RESULT:

res - float_st

SEE ALSO:

absf(3), ceilf(3), floorf(3), rintf(3)

3.7 cadna_intr/llrint

[*Functions*]

NAME:

llrint

SYNOPSIS:

res = llrint(x)

FUNCTION:

The llrint() functions return the integral value nearest to x (according to the prevailing rounding mode) in the return formats specified. If the rounded value is outside the range of the return type, the numeric result is unspecified and the "invalid" floating-point exception is raised. A range error may occur if the magnitude of x is too large.

INPUTS:

a - double_st

RESULT:

res - long long int

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.8 cadna_intr/llrintf

[*Functions*]

NAME:

llrintf

SYNOPSIS:

res = llrintf(x)

FUNCTION:

The llrintf() functions return the integral value nearest to x (according to the prevailing rounding mode) in the return formats specified. If the rounded value is outside the range of the return type, the numeric result is unspecified and the "invalid" floating-point exception is raised. A range error may occur if the magnitude of x is too large.

INPUTS:

a - float_st

RESULT:

res - long long int

SEE ALSO:

absf(3), ceilf(3), floorf(3), rintf(3)

3.9 cadna_intr/lrint

[*Functions*]

NAME:

lrint

SYNOPSIS:

res = lrint(x)

FUNCTION:

The lrint() functions return the integral value nearest to x (according to the prevailing rounding mode) in the return formats specified. If the rounded value is outside the range of the return type, the numeric result is unspecified and the "invalid" floating-point exception is raised. A range error may occur if the magnitude of x is too large.

INPUTS:

a - double_st

RESULT:

res - long int

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.10 cadna_intr/lrintf

[*Functions*]

NAME:

lrintf

SYNOPSIS:

res = lrintf(x)

FUNCTION:

The `lrintf()` functions return the integral value nearest to `x` (according to the prevailing rounding mode) in the return formats specified. If the rounded value is outside the range of the return type, the numeric result is unspecified and the "invalid" floating-point exception is raised. A range error may occur if the magnitude of `x` is too large.

INPUTS:

`a` - `float_st`

RESULT:

`res` - `long int`

SEE ALSO:

`absf(3)`, `ceilf(3)`, `floorf(3)`, `rintf(3)`

3.11 `cadna_intr/nearbyint`

[*Functions*]

NAME:

`nearbyint`

SYNOPSIS:

`res = nearbyint(x)`

FUNCTION:

The `nearbyint()` functions return the integral value (represented as a double precision number) nearest to `x` according to the prevailing rounding mode.

INPUTS:

`a` - `double_st`

RESULT:

`res` - `double_st`

SEE ALSO:

`abs(3)`, `ceil(3)`, `floor(3)`, `rint(3)`

3.12 `cadna_intr/nearbyintf`

[*Functions*]

NAME:

`nearbyintf`

SYNOPSIS:

`res = nearbyintf(x)`

FUNCTION:

The `nearbyintf()` functions return the integral value (represented as a double precision number) nearest to `x` according to the prevailing rounding mode.

INPUTS:

`a` - `float_st`

RESULT:

`res` - `float_st`

SEE ALSO:

`absf(3)`, `ceilf(3)`, `floorf(3)`, `rintf(3)`

3.13 `cadna_intr/rint`

[*Functions*]

NAME:

`rint`

SYNOPSIS:

`res = rint(x)`

FUNCTION:

The `rint()` functions return the integral value nearest to `x` (according to the prevailing rounding mode) in floating-point format.

INPUTS:

a - double_st

RESULT:

res - double_st

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.14 cadna_intr/rintf

[*Functions*]

NAME:

rintf

SYNOPSIS:

res = rintf(x)

FUNCTION:

The rintf() functions return the integral value nearest to x (according to the prevailing rounding mode) in floating-point format.

INPUTS:

a - float_st

RESULT:

res - float_st

SEE ALSO:

absf(3), ceilf(3), floorf(3), rintf(3)

3.15 cadna_intr/trunc

[*Functions*]

NAME:

trunc

SYNOPSIS:

res = trunc(x)

FUNCTION:

The trunc() functions return the integral value nearest to but no larger in magnitude than x.

INPUTS:

a - double_st

RESULT:

res - double_st

SEE ALSO:

abs(3), ceil(3), floor(3), rint(3)

3.16 cadna_intr/truncf

[*Functions*]

NAME:

truncf

SYNOPSIS:

res = truncf(x)

FUNCTION:

The truncf() functions return the integral value nearest to but no larger in magnitude than x.

INPUTS:

a - float_st

RESULT:

res - float_st

SEE ALSO:

absf(3), ceilf(3), floorf(3), rintf(3)

Chapter 4

I/O functions

4.1 `cadna_str/display`

[Methods]

NAME:

`display`

SYNOPSIS:

`display()`
`display(char *)`

FUNCTION:

The `display` method prints the triplet associated with a stochastic variable.

INPUTS: RESULT:

`void`

SEE ALSO:

`str(3)`

4.2 `cadna_str/operator<<`

[Methods]

NAME:

`operator<<`

SYNOPSIS:

```
ostream& operator <<(ostream&, const double_st &)
ostream& operator <<(ostream&, const float_st &)
```

FUNCTION:

`<<` operator for stochastic variables

INPUTS: RESULT:

`void`

SEE ALSO:

`str(3)`

4.3 `cadna_str/operator>>`

[Methods]

NAME:

`operator>>`

SYNOPSIS:

```
ostream& operator >>(ostream&, const double_st &)
ostream& operator >>(ostream&, const float_st &)
```

FUNCTION:

`>>` operator for stochastic variables

INPUTS: RESULT:

`void`

SEE ALSO:

`str(3)`

4.4 cadna_str/str

[*Functions*]

NAME:

str

SYNOPSIS:

```
char* double_st::str(char *s) const
char* float_st::str(char *s) const
char* str(char *, double_st&)
char* str(char *, float_st&)
```

FUNCTION:

The output string contains the scientific notation of the stochastic argument; only the exact significant digits appear in the string.

INPUTS:

The str function has a string argument and a stochastic argument.

RESULT:

It returns a pointer to the first argument.

SEE ALSO:

str(3)

4.5 cadna_str/strp

[*Functions*]

NAME:

strp

SYNOPSIS:

```
char* strp(double_st&)
char* strp(float_st&)
```

FUNCTION:

The output string contains the scientific notation of the stochastic argument; only the exact significant digits appear in the string. The `strp` function must be used only with the family of `printf` functions. The only restriction is that it is not possible to have more than 256 calls to the `strp` function in one call to the `printf` function.

INPUTS:

The `strp` function has a stochastic argument.

RESULT:

It returns a string.

SEE ALSO:

`str(3)`

Chapter 5

Cadna specific functions

5.1 cadna_type/cadna_disable

[Functions]

NAME:

cadna_disable

SYNOPSIS:

```
void cadna_disable(unsigned int tag)
```

FUNCTION:

The cadna_disable function disables the detection of a kind of numerical instability.

INPUTS:

unsigned int tag (see documentation)

RESULT: SEE ALSO:

cadna_enable(3)

5.2 cadna_type/cadna_enable

[Functions]

NAME:

cadna_enable

SYNOPSIS:

```
void cadna_enable(unsigned int tag)
```

FUNCTION:

The `cadna_enable` function enables the detection of a kind of numerical instability.

INPUTS:

unsigned int tag (see documentation)

RESULT: SEE ALSO:

```
cadna_disable(3)
```

5.3 `cadna_type/cadna_end`

[*Functions*]

NAME:

```
cadna_end
```

SYNOPSIS:

```
void cadna_end()
```

FUNCTION:

The `cadna_end` function "closes" the CADNA library (see documentation).

INPUTS:

no input parameter

RESULT:

```
void
```

5.4 cadna_type/cadna_init

[Functions]

NAME:

cadna_init

SYNOPSIS:

```
void cadna_init(int max_nb_instability)
void cadna_init(int max_nb_instability, unsigned int tag)
void cadna_init(int max_instability, unsigned int tag,
                unsigned int seed, unsigned int cancellation)
```

FUNCTION:

The cadna_init function initializes the CADNA library
(see documentation).

INPUTS: RESULT:

5.5 cadna_digitnumber/approx_digit

[Functions]

NAME:

approx_digit

SYNOPSIS:

```
res = x.approx_digit()
```

FUNCTION:

The approx_digit() function returns 0 if a stochastic
number is non significant and RELIABLE_RESULT otherwise.

INPUTS:

x - a stochastic number

RESULT:

res - an integer value

5.6 `cadna_digitnumber/nb_significant_digit`

[*Functions*]

NAME:

`nb_significant_digit`

SYNOPSIS:

```
res = x.nb_significant_digit()
```

FUNCTION:

The `nb_significant_digit()` function returns the number of exact significant digits of a stochastic `x`

INPUTS:

`x` - a stochastic number

RESULT:

`res` - an integer value

5.7 `cadna_computedzero/computedzero`

[*Methods*]

NAME:

`computedzero`

SYNOPSIS:

```
res = computedzero(x)
```

FUNCTION:

The `computedzero()` function returns 1 if `x` is a stochastic zero, 0 otherwise.

INPUTS:

`x` - a stochastic number

RESULT:

`res` - an integer value

5.8 cadna_numericalnoise/numericalnoise

[Methods]

NAME:

numericalnoise

SYNOPSIS:

```
res = numericalnoise(x)
```

FUNCTION:

The numericalnoise() function returns 1 if x is a numerical noise,
0 otherwise.

INPUTS:

x - a stochastic number

RESULT:

res - an integer value

Chapter 6

Conversion functions

6.1 `cadna_convert/char`

[Methods]

NAME:

`char`

SYNOPSIS:

```
res = (char)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - a char
```

6.2 `cadna_convert/data_st`

[Methods]

NAME:

`data_st`

SYNOPSIS:

```
res = data_st()  
res = data_st(x,y)
```

FUNCTION:

the `data_st(x,y)` method allows to take into account data uncertainty at the initialization of stochastic variables.
if `y == 0`, `x` is an absolute error
else `x` is a relative error

the `data_st()` method perturbs the last bit.

INPUTS:

`x` - a double
`y` - an integer

RESULT:

`res` - a stochastic number

6.3 `cadna_convert/double`

[Methods]

NAME:

`double`

SYNOPSIS:

`res = (double)x`

FUNCTION: INPUTS:

`x` - a stochastic

RESULT:

`res` - a double

6.4 `cadna_convert/float`

[Methods]

NAME:

`float`

SYNOPSIS:

```
res = (float)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - a float
```

6.5 cadna_convert/int

[Methods]

NAME:

```
int
```

SYNOPSIS:

```
res = (int)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - an int
```

6.6 cadna_convert/long

[Methods]

NAME:

```
long
```

SYNOPSIS:

```
res = (long)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - a long
```

6.7 `cadna_convert/long long`

[Methods]

NAME:

`long long`

SYNOPSIS:

`res = (long long)x`

FUNCTION: INPUTS:

`x` - a stochastic

RESULT:

`res` - a long long

6.8 `cadna_convert/short`

[Methods]

NAME:

`short`

SYNOPSIS:

`res = (short)x`

FUNCTION: INPUTS:

`x` - a stochastic

RESULT:

`res` - a short

6.9 `cadna_convert/unsigned char`

[Methods]

NAME:

`unsigned char`

SYNOPSIS:

```
res = (unsigned char)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - an unsigned char
```

6.10 cadna_convert/unsigned int

[Methods]

NAME:

```
unsigned int
```

SYNOPSIS:

```
res = (unsigned int)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - an unsigned int
```

6.11 cadna_convert/unsigned long

[Methods]

NAME:

```
unsigned long
```

SYNOPSIS:

```
res = (unsigned long)x
```

FUNCTION: INPUTS:

```
x          - a stochastic
```

RESULT:

```
res        - an unsigned long
```

6.12 `cadna_convert/unsigned long long`

[*Methods*]

NAME:

`unsigned long long`

SYNOPSIS:

`res = (unsigned long long)x`

FUNCTION: INPUTS:

`x` - a stochastic

RESULT:

`res` - an unsigned long long

6.13 `cadna_convert/unsigned short`

[*Methods*]

NAME:

`unsigned short`

SYNOPSIS:

`res = (unsigned short)x`

FUNCTION: INPUTS:

`x` - a stochastic

RESULT:

`res` - an unsigned short

6.14 cadna_to/operator=

[*Methods*]

NAME:

operator=

SYNOPSIS:

res = a

FUNCTION:

Define all the functions involving at least one argument of stochastic type which overload the assignment statement "=".

INPUTS:

a - an integer, a float, a double, float_st or double_st

RESULT:

res - float_st or double_st

Chapter 7

Expert functions

7.1 `cadna_type/cadna_set_rnd_arr()`

[Functions]

NAME:

`cadna_set_rnd_arr`

SYNOPSIS:

`void cadna_set_rnd_arr()`

FUNCTION:

The `cadna_set_rnd_arr()` function sets the rounding mode to the nearest

BE CARREFULL : this function is only for CADNA expert. It can change the CADNA behaviour and generates errors.

INPUTS: RESULT:

7.2 `cadna_type/cadna_set_rnd_moinf()`

[Functions]

NAME:

`cadna_set_rnd_moinf`

SYNOPSIS:

`void cadna_set_rnd_moinf()`

FUNCTION:

The `cadna_set_rnd_moinf()` function sets the rounding mode to the nearest

BE CARREFULL : this function is only for CADNA expert. It can change the CADNA behaviour and generates errors.

INPUTS: RESULT:**7.3 `cadna_type/cadna_set_rnd_plinf()`**

[*Functions*]

NAME:

`cadna_set_rnd_plinf`

SYNOPSIS:

`void cadna_set_rnd_plinf()`

FUNCTION:

The `cadna_set_rnd_plinf()` function sets the rounding mode to the nearest

BE CARREFULL : this function is only for CADNA expert. It can change the CADNA behaviour and generates errors.

INPUTS: RESULT:**7.4 `cadna_type/cadna_set_rnd_zero()`**

[*Functions*]

NAME:

`cadna_set_rnd_zero`

SYNOPSIS:

`void cadna_set_rnd_zero()`

FUNCTION:

The `cadna_set_rnd_zero()` function sets the rounding mode to the nearest

BE CARREFULL : this function is only for CADNA expert. It can change the CADNA behaviour and generates errors.

INPUTS: RESULT:

Chapter 8

Internal functions

8.1 cadna_unstab/instability

[*Functions*]

NAME:

instability

SYNOPSIS:

```
void instability(unsigned long *unstab)
```

FUNCTION:

It is an internal subroutine of the CADNA library. It manages the different kinds of instabilities detected by CADNA.

INPUTS:

An unsigned long integer which indicates the kind of instability that has been detected.

RESULT: SEE ALSO: