MathLib Reference

This appendix provides a reference for the numeric implementation in the C programming language. It summarizes the data formats available and tells how to determine the floating-point class for a value. It also lists functions that control the floating-point environment, functions that perform floating-point operations, and the exceptions those functions might raise.

Floating-Point Data Formats

Figure E-1 Floating-point data formats

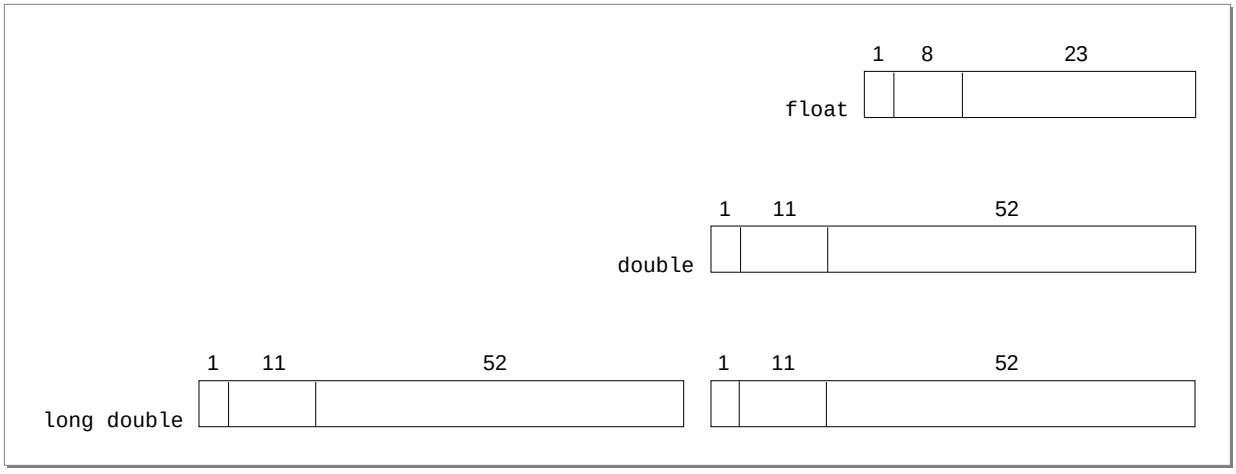

Table E-1 Interpreting floating-point values

If biased* exponent *e* is:	And fraction *f* is:	Then value *v* is:	And class of *v* is: †
$0 < e < max$ ‡	(any)	$v = (-1)^s \times 2^{(e-bias)} \times (1.f)$ §	FP_NORMAL
$e = 0$	$f \neq 0$	$v = (-1)^s \times 2^{minexp} \times (0.f)$ ¶	FP_SUBNORMAL
$e = 0$	$f = 0$	$v = (-1)^s \times 0$	FP_ZERO
$e = max$	$f = 0$	$v = (-1)^s \times \infty$	FP_INFINITE
$e = max$	$f \neq 0$	$v = \mathrm{NaN}$	FP_SNAN (first bit is 0) FP_QNAN (first bit is 1)

* *bias* = 127 for `float`; 1023 for `double` and `long double`.
† From enumerated type `NumKind`.
‡ *max* = 255 for `float`; 2047 for `double` and `long double`.
§ For `long double` both head and tail are evaluated this way and added together.
¶ *minexp* = –126 for `float`; –1022 for `double` and `long double`.

Table E-2 Class and sign inquiry macros

```
fpclassify(x)
isnormal(x)
isfinite(x)
isnan(x)
signbit(x)
```

Environmental Controls

Table E-3 Environmental access

Action	Function prototype
Get	`void fegetenv (fenv_t *envp);`
Set	`void fesetenv (const fenv_t *envp);`
Save	`int feholdexcept (fenv_t * envp);`
Restore	`void feupdateenv (const fenv_t *envp);`

Table E-4 Floating-point exceptions

Exceptions	Action	Function prototype
`FE_INEXACT`	Get	`void fegetexcept(fexcept_t *flagp, int excepts);`
`FE_DIVBYZERO`	Set	`void feraiseexcept (int excepts);`
`FE_UNDERFLOW`	Clear	`void feclearexcept (int excepts);`
`FE_OVERFLOW`		`void fesetexcept (const fexcept_t *flagp, int excepts);`
`FE_INVALID`	Test	`int fetestexcept (int excepts);`

Table E-5 Rounding direction modes

Modes	Action	Function prototype
`FE_TONEAREST`	Get	`int fegetround (void);`
`FE_TOWARDZERO`	Set	`int fesetround (int round);`
`FE_UPWARD`		
`FE_DOWNWARD`		

Operations and Functions

Note
Throughout the tables that follow, in the Exceptions column, I = invalid; X = inexact; O = overflow; U = underflow; D = divide-by-zero. ◆

Table E-6 Arithmetic operations

Compute	Syntax	Valid input range	Exceptions
Sum	`x + y`	–∞ to +∞	`I X O U -`
Difference	`x – y`	–∞ to +∞	`I X O U -`
Product	`x * y`	–∞ to +∞	`I X O U -`
Quotient	`x / y`	–∞ to +∞	`I X O U D`
Square root	`sqrt(x)`	0 to +∞	`I X - - -`

Remainder	`remainder(x,y)`
	`remquo(x,y,quo)`
	`fmod(x,y)`
		–∞ to +∞	`I - - - -`

Table E-7 Conversions to integer type

Compute	Syntax	Valid input range	Exceptions
Round in current direction	`rinttol(x)`*	-2^{31} to $2^{31}-1$	`I X - - -`
Add 1/2 to magnitude and chop	`roundtol(x)`*	-2^{31} to $2^{31}-1$	`I X - - -`

* Return type of `long int`.

Table E-8 Conversions to integer in floating-point type

Compute	Syntax	Valid input range	Exceptions
Round in current direction	`rint(x)`	–∞ to +∞	`- X - - -`
	`nearbyint(x)`	–∞ to +∞	`- - - - -`
Round upward	`ceil(x)`	–∞ to +∞	`- - - - -`
Round downward	`floor(x)`	–∞ to +∞	`- - - - -`
Add 1/2 to magnitude and chop	`round(x)`	–∞ to +∞	`- X - - -`
Round toward zero	`trunc(x)`	–∞ to +∞	`- - - - -`

Table E-9 Conversions between binary and decimal formats

Compute	Syntax	Valid input range	Exceptions
Convert `decimal` struct to binary	`dec2num(&d)`	`decimal` struct	– – – – –
Convert binary to `decimal` struct	`num2dec(&f,x,&d)`	$-\infty$ to $+\infty$	– – – – –

Table E-10 Conversions between decimal formats

Compute	Syntax	Valid input range	Exceptions
Convert `decimal` struct to string	`dec2str(&f,&d,s)`	`decimal` struct	– – – – –
Convert decimal string to struct	`str2dec(s,&ix,&d,&vp)`	Numeric string	– – – – –

Table E-11 Comparison operations

Compute	Syntax	Valid input range	Exceptions
Positive difference or 0	`fdim(x,y)`	–∞ to +∞	- X O U -
Maximum of 2 numbers	`fmax(x,y)`	–∞ to +∞	- - - - -
Minimum of 2 numbers	`fmin(x,y)`	–∞ to +∞	- - - - -
Relationship of x, y	`relation(x,y)`	–∞ to +∞	- - - - -

Table E-12 Sign manipulation functions

Compute	Syntax	Valid input range	Exceptions
Copy the sign	`copysign(x,y)`	–∞ to +∞	- - - - -
$\lvert x \rvert$	`fabs(x)`	–∞ to +∞	- - - - -

Table E-13 Exponential functions

Compute	Syntax	Valid input range	Exceptions
e^x	`exp(x)`	–∞ to +∞	- X O U -
2^x	`exp2(x)`	–∞ to +∞	- X O U -
$e^x - 1$	`expm1(x)`	–∞ to +∞	- X O U -
$x \times 2^n$	`ldexp(x,n)`	–∞ to +∞	- X O U -
	`scalb(x,n)`		- X O U -
x^y	`pow(x,y)`	–∞ to +∞	I X O U D

Table E-14 Logarithmic functions

Compute	Syntax	Valid input range	Exceptions
Fraction and exponent fields of floating-point number	`frexp(x,&n)`	$-\infty$ to $+\infty$	`- - - - -`
$\ln x$	`log(x)`	0 to $+\infty$	`I X - - D`
$\log_{10} x$	`log10(x)`	0 to $+\infty$	`I X - - D`
$\ln (x + 1)$	`log1p(x)`	> -1	`I X - - D`
$\log_2 x$	`log2(x)`	0 to $+\infty$	`I X - - D`
Exponent field of floating-point number	`logb(x)`	$-\infty$ to $+\infty$	`- - - - D`
Split real number into fractional part and integer part	`modf(x,&y)`	$-\infty$ to $+\infty$	`- - - - -`

Table E-15 Trigonometric functions

Compute	Syntax	Valid input range	Exceptions
$\cos x$	`cos(x)`	Any finite number	`I X - - -`
$\sin x$	`sin(x)`	Any finite number	`I X - U -`
$\tan x$	`tan(x)`	Any finite number	`I X - U -`
$\arccos x$	`acos(x)`	–1 to +1	`I X - - -`
$\arcsin x$	`asin(x)`	–1 to +1	`I X - U -`
$\arctan x$	`atan(x)`	$-\infty$ to $+\infty$	`- X - U -`
$\arctan y/x$	`atan2(x,y)`	$-\infty$ to $+\infty$	`- X - U -`

Table E-16 Hyperbolic functions

Compute	Syntax	Valid input range	Exceptions
cosh x	`cosh(x)`	$-\infty$ to $+\infty$	`- X O - -`
sinh x	`sinh(x)`	$-\infty$ to $+\infty$	`- X O U -`
tanh x	`tanh(x)`	$-\infty$ to $+\infty$	`- X - - -`
arccosh x	`acosh(x)`	1 to $+\infty$	`I X - - -`
arcsinh x	`asinh(x)`	$-\infty$ to $+\infty$	`- X - U -`
arctanh x	`atanh(x)`	–1 to +1	`I X - U -`

Table E-17 Financial functions

Compute	Syntax	Valid input range	Exceptions
Compound interest	`compound(r,p)`	0 to $+\infty$	`I X - - D`
Annuity	`annuity(r,p)`	0 to $+\infty$	`I X - - D`

Table E-18 Error and gamma functions

Compute	Syntax	Valid input range	Exceptions
error	`erf(x)`	$-\infty$ to $+\infty$	`- X - U -`
1 – error	`erfc(x)`	$-\infty$ to $+\infty$	`- X - U -`
$\Gamma(x)$	`gamma(x)`	0 to $+\infty$	`I X O - -`
$\ln(\lvert\Gamma(x)\rvert)$	`lgamma(x)`	0 to $+\infty$	`I X O - -`

Table E-19 Miscellaneous functions

Compute	Syntax	Valid input range	Exceptions
Create NaN	`nan(tagp)`	character string	`- - - - -`
Next representable number after x in direction of y	`nextafterd(x,y)`	$-\infty$ to $+\infty$	`- X O U -`
Hypotenuse	`hypot(x,y)`	$-\infty$ to $+\infty$	`- X O U -`
Random number generator	`randomx(&x)`	1 to $2^{31} - 2$	`- - - - -`