

# math.h in C

Header files named math.h in Library C define diverse math functions and a macro. All functions available in this library receive double as a parameter and return the result in type double.

## math.h in C

Header files named math.h in Library C define diverse math functions and a macro. All functions available in this library receive double as a parameter and return the result in type double.

## Macros are defined in math.h

There is only one Macro defined in this library:

**HUGE\_VAL:** This macro is used when the result of a function cannot be represented as a floating-point number. If the size to correctly represent the result is too large, the errno set function to ERANGE to indicate a Range Error, and return a value large enough, specifically named by HUGE\_VAL or its counter - HUGE\_VAL .

If the size of the result is too small, a value of 0 is returned. In this case, errno may or may not be set to ERANGE.

## Functions are defined in math.h

Here are some functions defined in math.h in Library C:

Description function double acos (double x)

Returns arcos of x (radian value)

double asin (double x)

Returns arcsin of x (radian value)

double atan (double x)

Returns arctan of x (radian value)

double atan2 (doubly y, double x)

Returns arctan of  $y / x$  (radian value)

double cos (double x)

Returns cos of the angle x (radian value)

double cosh (double x)  
Returns cosh (hyperbolic cosine function) of x (radian value)

double sin (double x)  
Returns the sine of the angle x (radian value)

double sinh (double x)  
Returns sinh (hyperbolic sine function) of x (radian value)

double tanh (double x)  
Returns tanh (tanh hyperbolic function) of x (radian value)

double exp (double x)  
Returns  $e^x$

double frexp (double x, int \* exponent)  
The value returned is the mantissa (the mantissa) and the integer pointed to by exponent (exponent). The result value is  $x = \text{mantissa} * 2^{\text{exponent}}$

double ldexp (double x, int exponent)  
Returns  $x * 2^{\text{exponent}}$

double log (double x)  
Returns  $\ln x$

double log10 (double x)  
Returns  $\log_{10}(x)$

double modf (double x, double \* integer)  
Returns the decimal part of x

double pow (double x, double y)  
Returns  $x^y$

double sqrt (double x)  
Returns the square root of x

double ceil (double x)  
Returns the smallest integer value greater than or equal to x

double fabs (double x)  
Returns the absolute value of x

double floor (double x)  
Returns the maximum integer value less than or equal to x

double fmod (double x, double y)  
Returns the remainder of the x / y division

## 1. The function `acos ()` in C

The function `double acos (double x)` in Library C returns the arcos of  $x$  (radian value).

### Declare `acos ()` function in C

Below is the declaration for `acos ()` function in C:

```
double acos ( double x )
```

*Parameters:*

$x$  - This is the actual floating point number in the paragraph  $[-1, + 1]$ .

*Return value:*

This function returns the arcos of  $x$ , in the segment  $[0, \pi]$  radian.

*For example:*

The following program C illustrates how to use the `acos ()` function in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , ret , v
```

Compiling and running the above C program will result:

```
Gia tri arccos cua 0.900000 la bang 25.841933
```

## 2. The function `asin ()` in C

The `double asin (double x)` function in Library C returns arcsin of  $x$  (radian value).

### Declare the function `asin ()` in C

Below is the declaration for the `asin ()` function in C:

```
double asin ( double x )
```

*Parameters:*

$x$  - This is the actual floating point number in the paragraph  $[-1, + 1]$ .

*Return value:*

This function returns arcsin of  $x$ , in the  $[-\pi / 2, + \pi / 2]$  radian segment.

*For example:*

The following program C illustrates the usage of the asin () function in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , ret , v
```

Compiling and running the above C program will result:

```
Gia tri arcsin cua 0.900000 la bang 64.158067
```

### 3. The atan () function in C

Double atan (double x) function in C Library returns arctan of x (radian value).

#### Declaring the function atan () in C

Below is the declaration for the atan () function in C:

```
double atan ( double x )
```

*Parameters:*

x - This is the floating point real number value.

*Return value:*

This function returns arctan of x, in the  $[-\pi / 2, +\pi / 2]$  radian.

*For example:*

The following C program illustrates the use of atan () function in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , ret , v
```

Compiling and running the above C program will result:

```
Gia tri arctan cua 1.000000 la bang 45.000000
```

### 4. The function atan2 () in C

Double atan2 (double y, double x) in Library C returns arctan of y / x (radian value).

#### Declare the function atan2 () in C

Here is the declaration for the atan2 () function in C:

```
double atan2 ( double y , double x )
```

*Parameters:*

1. x - This is a real floating point value representing the x coordinate.
2. y - This is the actual numerical value of floating point representing y coordinate.

*Return value:*

This function returns arctan of y / x, in the [-pi, + pi] radian segment.

*For example:*

The following C program illustrates the usage of atan2 () function in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , y , ret
```

Compiling and running the above C program will result:

```
Gia tri arctan2 cua x = -7.000000, y = 7.000000 la bang 135.000000
```

## 5. Function cos () in C

The double cos function (double x) in Library C returns the cos of the angle x (measured in radians).

### Declaring the function cos () in C

Below is the declaration for the function cos () in C:

```
double cos ( double x )
```

*Parameters:*

x - This is a real floating point value representing an angle expressed in radian values.

*Return value:*

This function returns the cos of x.

*For example:*

The following program C illustrates the usage of function cos () in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , ret , v
```

Compiling and running the above C program will result:

```
Gia tri cosin cua 60.000000 la bang 0.500000
Gia tri cosin cua 90.000000 la bang 0.000000
-----
```

## 6. Cosh () function in C

The double cosh (double x) function in Library C returns the hyperbolic cosine of x.

### Declare the function cosh () in C

Below is the declaration for the cosh () function in C:

```
double cosh ( double x )
```

*Parameters:*

x - This is the floating point real number value.

*Return value:*

This function returns the hyperbolic cosine of x.

*For example:*

The following C program illustrates the usage of the cosh () function in C:

```
#include <math.h>
int main () { double x ; x = 0.5 ; printf ( "Gia tri hyperbolic cosine cua %f la bang %f\n", x, cosh(x)) ; }
```

Compiling and running the above C program will result:

```
Gia tri hyperbolic cosine cua 0.500000 la bang 1.127626
Gia tri hyperbolic cosine cua 1.000000 la bang 1.543081
Gia tri hyperbolic cosine cua 1.500000 la bang 2.352410
-----
```

## 7. Function sin () in C

The double sin (double x) function in Library C returns the sine of the angle x (radian value).

### Declaring the function sin () in C

Below is the declaration for function sin () in C:

```
double sin ( double x )
```

*Parameters:*

x - This is a real floating point value representing an angle expressed in radian values.

*Return value:*

This function returns the sine of x.

*For example:*

The following C program illustrates the usage of the function `sin ()` in C:

```
#include <math.h> #include <stdio.h> #define PI 3.14159265 int main () { double x , ret , v;
```

Compiling and running the above C program will result:

```
Sin cua 45.000000 bang 0.707107
```

## 8. The generating function () in C

The double function `sinh (double x)` in Library C returns the hyperbolic sine of x.

### Declare the generating function () in C

Below is the declaration for the generating function () in C:

```
double sinh ( double x )
```

*Parameters:*

x - This is the floating point real number value.

*Return value:*

This function returns the hyperbolic sine of x.

*For example:*

The following program C illustrates the usage of the generating function () in C:

```
#include <math.h> #include <stdio.h> int main () { double x , ret ; x = 0.5 ; ret = sinh
```

Compiling and running the above C program will result:

```
Hyperbolic sine cua 0.500000 bang 0.521095
```

## 9. Ham function () in C

The function `double tanh (double x)` in Library C returns the hyperbolic tangent of x.

## Declaring fish function () in C

Below is the declaration for the fishy function () in C:

```
double tanh ( double x )
```

*Parameters:*

x - This is the floating point real number value.

*Return value:*

This function returns the hyperbolic tangent of x.

*For example:*

The following program C illustrates the usage of the fishy function () in C:

```
#include <math.h>
int main () { double x , ret ; x = 0.5 ; ret = tanh ( x ) ; printf ( "tanh ( %f ) = %f\n" , x , ret ) ; }
```

Compiling and running the above C program will result:

```
Hyperbolic tangent của 0.500000 bằng 0.462117
```

## 10. Function exp () in C

The function double exp (double x) in Library C returns e hats x.

### Declare the function exp () in C

Below is the declaration for exp () in C:

```
double exp ( double x )
```

*Parameters:*

x - This is the floating point real number value.

*Return value:*

This function returns ex.

*For example:*

The following C program illustrates the usage of exp () in C:

```
#include <math.h>
int main () { double x = 0 ; printf ( "e^%lf = %lf\n" , x , exp ( x ) ) ; }
```

Compiling and running the above C program will result:

```
e^0.000000 = 1.000000
e^1.000000 = 2.718282
e^2.000000 = 7.389056
```

## 11. Function frexp () in C

The double frexp function (double x, int \* exponent) in Library C returns the mantissa (the mantissa) and the integer pointed to by exponent (exponent). The result value is  $x = \text{mantissa} * 2^{\text{exponent}}$ .

### Declaring the function frexp () in C

Below is the declaration for frexp () function in C:

```
double frexp ( double x , int * exponent )
```

*Parameters:*

x - This is the actual floating point number value to be calculated.

exponent - This is the pointer to an int object where the exponent value is stored.

*Return value:*

This function returns the mantissa and the integer pointed to by exponent.

*For example:*

The following program C illustrates the usage of frexp () function in C:

```
#include <math.h>
int main () { double x = 1024 , fraction ; int e ; fra
```

Compiling and running the above C program will result:

```
x = 1024.00 = 0.50 * 2^11
```

## 12. The ldexp () function in C

The double ldexp (double x, int exponent) function in Library C returns  $x * 2^{\text{exponent}}$ .

**Declare the function ldexp () in C**

Below is the declaration for ldexp () in C:

```
double ldexp ( double x , int exponent )
```

**Parameters:**

x - This is the actual floating point number, also representing the number of digits after the comma.

exponent - This is the value of exponent.

**Return value:**

This function returns  $x * 2^{\text{exp}}$

**For example:**

The following C program illustrates the usage of ldexp () in C:

```
#include <math.h>
int main () { double x , ret ; int n ; x = 0.65 ; n = 3 ; ret = ldexp ( x , n ) ; printf ( "0.650000 * 2^3 = %f\n" , ret ) ; }
```

Compiling and running the above C program will result:



```
0.650000 * 2^3 = 5.200000
```

### 13. Log () function in C

The double log (double x) function in Library C returns  $\ln x$ .

**Declare the log () function in C**

Below is the declaration for the log () function in C:

```
double log ( double x )
```

**Parameters:**

x - This is the floating point real number value.

**Return value:**

This function returns  $\ln x$ .

**For example:**

The following C program illustrates the usage of the log () function in C:

```
#include <math.h>
int main () { double x , ret ; x = 2.7 ; /* tim gia tri ln x */ ret = log ( x ) ; printf ( "ln 2.7 = %f\n" , ret ) ; }
```

Compiling and running the above C program will result:

```
log<2.700000> = 0.993252
```

## 14. Log10 () function in C

The double log10 function (double x) in Library C returns log<sub>10</sub> (x).

### Declare log10 () function in C

Below is the declaration for log10 () function in C:

```
double log10 ( double x )
```

#### Parameters:

x - This is the floating point real number value.

#### Return value:

This function returns log<sub>10</sub> (x), with x greater than 0.

#### For example:

The following C program illustrates how to use the log10 () function in C:

```
#include <stdio.h>
int main () { double x , ret ; x = 1000 ; /* tim gia tra
log10 1000 */ ret = log10 ( x ); printf ( "log10(%lf) = %lfn" , x , ret ); return 0 ; }
```

Compiling and running the above C program will result:

```
log10<1000.000000> = 3.000000
```

## 15. Modf () function in C

The function double modf (double x, double \* integer) in Library C returns the decimal part of x.

### Declare the modf () function in C

Below is the declaration for the modf () function in C:

```
double modf ( double x , double * integer )
```

**Parameters:**

x - This is the floating point real number value.

integer - This is a pointer to an object where the integer part is stored.

**Return value:**

This function returns the decimal part of x, with the same sign.

**For example:**

The following C program illustrates the usage of the modf () function in C:

```
#include <math.h>
int main () { double x , fractpart , intpart ; x = 7.123456 ;
```

Compiling and running the above C program will result:

```
Phan nguyen = 7.000000
Phan thap phan = 0.123456
```

## 16. The pow () function in C

Double pow function (double x, double y) in Library C returns xy.

**Declare the function pow () in C**

Below is the declaration for pow () in C:

```
double pow ( double x , double y )
```

**Parameters:**

1. x - This is the radix, which is a floating point real number value.
2. y - This is an exponent, which is a floating point real number value.

**Return value:**

This function returns xy.

**For example:**

The following program C illustrates the usage of the pow () function in C:

```
#include <math.h>
int main () { printf ( "9.0 ^ 3 = %lfn" , pow ( 9.0 , 3 ) ) ;
```

Compiling and running the above C program will result:

```
9.0 ^ 3 = 729.000000
4.05 ^ 2.98 = 64.597538
```

## 17. The sqrt () function in C

The function double sqrt (double x) in Library C returns the square root of x.

### Declare the function sqrt () in C

Below is the declaration for the sqrt () function in C:

```
double sqrt ( double x )
```

### Parameters:

x - This is the floating point real number value.

### Return value:

This function returns the square root of x.

### For example:

The following C program illustrates the usage of the sqrt () function in C:

```
#include <math.h>
int main () { printf ( "Can bac hai cua %lf bang %lf\n" , 4
```

Compiling and running the above C program will result:

```
Can bac hai cua 4.000000 bang 2.000000
Can bac hai cua 5.000000 bang 2.236068
```

## 18. Function ceil () in C

The double ceil (double x) function in Library C returns the smallest integer value greater than or equal to x.

### Declare the function ceil () in C

Below is the declaration for the function ceil () in C:

```
double ceil ( double x )
```

### Parameters:

x - This is the floating point real number value.

**Return value:**

Returns the smallest integer value not less than x.

**For example:**

The following program C illustrates the usage of the function `ceil ()` in C:

```
#include <math.h>
#include <stdio.h>
int main () { float val1 , val2 , val3 , val4 ; val1 = 1.6 ; val2 = 1.2 ; val3 = 2.8 ; val4 = 2.3 ;
printf ("ceil (1.6) = %f\n", ceil (val1)) ;
printf ("ceil (1.2) = %f\n", ceil (val2)) ;
printf ("ceil (2.8) = %f\n", ceil (val3)) ;
printf ("ceil (2.3) = %f\n", ceil (val4)) ;
return 0 ;
}
```

Compiling and running the above C program will result:

```
Gia tri cua ceil(1.6) = 2.0
Gia tri cua ceil(1.2) = 2.0
Gia tri cua ceil(2.8) = 3.0
Gia tri cua ceil(2.3) = 3.0
-----
```

## 19. The `fabs ()` function in C

The function `double fabs (double x)` in Library C returns the absolute value of x.

**Declare `fabs ()` in C**

Below is the declaration for `fabs ()` in C:

```
double fabs ( double x )
```

**Parameters:**

x - This is the floating point real number value.

**Return value:**

This function returns the absolute value of x.

**For example:**

The following C program illustrates the usage of the `fabs ()` function in C:

```
#include <math.h>
#include <stdio.h>
int main () { int a , b ; a = 345 ; b = - 345 ; printf ("fabs (345) = %f\n", fabs (a)) ; printf ("fabs (-345) = %f\n", fabs (b)) ;
return 0 ;
}
```

Compiling and running the above C program will result:

```
Tri tuyet doi: |345| = 345.000000
Tri tuyet doi: |-345| = 345.000000
-----
```

## 20. Floor () function in C

The double floor (double x) function in Library C returns the largest integer value less than or equal to x.

### Declare the floor () function in C

Below is the declaration for the floor () function in C:

```
double floor ( double x )
```

#### Parameters:

x - This is the floating point real number value.

#### Return value:

This function returns the largest integer not greater than x.

#### For example:

The following program C illustrates the usage of the floor () function in C:

```
#include <math.h>
int main () { float val1 , val2 , val3 , val4 ; val1 = 1.6 ; val2 = 1.2 ; val3 = 2.8 ; val4 = 2.3 ;
```

Compiling and running the above C program will result:

```
Gia tri cua floor(1.6) = 1.0
Gia tri cua floor(1.2) = 1.0
Gia tri cua floor(2.8) = 2.0
Gia tri cua floor(2.3) = 2.0
-----
```

## 21. Function fmod () in C

Double fmod (double x, double y) in Library C returns the remainder of the division x / y.

### Declaring the function fmod () in C

Here is the declaration for fmod () function in C:

```
double fmod ( double x , double y )
```

### Parameters:

x - This is a floating point numerical value that serves as a numerator.

y - This is a floating point numerical value that serves as a denominator.

### Return value:

This function returns the remainder of the x / y division.

### For example:

The following program C illustrates the usage of the fmod () function in C:

```
#include <math.h>
int main () { float a , b ; int c ; a = 9.2 ; b = 3.7 ; c = fmod ( a , b ) ; printf ( "Phan du cua phep chia %f / %f la %d\n" , a , b , c ) ; }
```

Compiling and running the above C program will result:

```
Phan du cua phep chia 9.200000 / 2 la 1.200000
Phan du cua phep chia 9.200000 / 3.700000 la 1.800000
-----
```

According to Tutorialspoint

Previous lesson: locale.h in C

Next lesson: Library C:

You finished reading the article "**math.h in C**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.