



APRENDERAPROGRAMAR.COM

FUNCIONES MATEMÁTICAS  
EN C. REDONDEO, VALOR  
ABSOLUTO, POTENCIA,  
TRIGONOMÉTRICAS, RAÍZ  
CUADRADA . (CU00520F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

**Resumen:** Entrega nº20 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

## FUNCIONES MATEMÁTICAS EN LENGUAJE C

El lenguaje C nos facilita una biblioteca de funciones matemáticas entre las que se incluyen las de uso más habitual como pueden ser: valor absoluto, potencia de un número elevado a otro, raíz cuadrada, funciones trigonométricas (seno, coseno, tangente, etc.), redondeo, exponenciación, logaritmo decimal, logaritmo neperiano y otras.



Para utilizar las funciones matemáticas indicadas a continuación es necesario incluir en la cabecera de nuestros programas la siguiente declaración: `#include <math.h>`

El resultado de aplicar una función matemática es un valor numérico de tipo *double* (aunque el resultado puede ser un valor entero, internamente C lo considerará como un valor tipo decimal de doble precisión o tipo *double*)

Las funciones disponibles de uso más habitual son las siguientes:

Funciones Matemáticas	Significado	Funciones Matemáticas	Significado
<code>fabs (x)</code>	Valor absoluto de x	<code>ceil (x)</code>	Redondeo entero superior
<code>pow (x, y)</code>	Potencia de x elevado a y	<code>floor (x)</code>	Redondeo entero inferior
<code>sqrt (x)</code>	Raíz cuadrada de x	<code>exp (x)</code>	Exponenciación (número e elevado a x, siendo e = 2.71828)
<code>sin (x)</code>	Seno de x	<code>log (x)</code>	Logaritmo neperiano de x
<code>cos (x)</code>	Coseno de x	<code>log10 (x)</code>	Logaritmo decimal de x
<code>tan (x)</code>	Tangente de x	<code>atan (x)</code>	Arcotangente de x

La biblioteca de funciones matemáticas de C es más amplia de lo que hemos expuesto (por ejemplo se dispone de más funciones trigonométricas y de otros tipos que las aquí indicadas), pero con estas funciones quedan cubiertos los cálculos más habituales.

Las funciones relacionadas con ángulos trabajan en radianes. Por tanto para operar con grados habremos de realizar la conversión oportuna.

Vamos a comentar algunas funciones menos habituales. Respecto a *ceil* y *floor*, se trata de funciones que devuelven el redondeo entero de un número. La diferencia entre ambas es que *ceil* redondea al entero superior más próximo y *floor* redondea al entero inferior más próximo. Tener en cuenta que el

entero superior más próximo a un número negativo como  $-5.75$  es  $-5$  mientras que el entero superior más próximo a un número positivo como  $5.75$  es  $6$ . Recordar que aunque estas funciones devuelven enteros, internamente el tipo de datos devuelto por estas funciones en C es *double*. Si se aplican estas funciones a un número entero, obtenemos ese mismo número pero con formato *double*.

C incorpora funciones que permiten generar números aleatorios así como funciones de fecha y hora. Por su importancia y extensión no las trataremos ahora.

Escribe y ejecuta el código mostrado a continuación. Crea también nuevas variables y operaciones para comprobar el uso de las funciones matemáticas en C.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
// Ejemplo aprenderaprogramar.com
int main() {
    double M = -5.75; int N = 7; double Pi = 3.14159; //Equivale a 180 grados sexagesimales
    printf("El valor absoluto de M es %g y el de N es %g\n", fabs(M), fabs(N));
    printf("El coseno de Pi es %g\n", cos(Pi));    printf("2 elevado al cubo vale %g\n", pow(2, 3));
    printf("El numero pi redondeado con ceil vale %g\n", ceil(Pi));
    printf("M redondeado con ceil vale %g\n", ceil(M));
    printf("El numero pi redondeado con floor vale %g\n", floor(Pi));
    printf("M redondeado con floor vale %g\n", floor(M)); printf("-M redondeado con floor vale %g\n", floor(-M));
    printf("El numero e vale %g\n", exp(1)); printf("El logaritmo neperiano de e vale %g\n", log(exp(1)));
    printf("El logaritmo decimal de 100 vale %g\n", log10(100));
    printf("La raiz cuadrada de 81 vale %g\n", sqrt(81));
    return 0;
}
```



Si vas a trabajar con constantes físicas o matemáticas te resultará de interés el empleo de `const` para declaración de constantes. La ventaja de declarar una constante en vez de una variable es que la constante no puede variar en el transcurso del programa. Por tanto, se impide que por error pueda tener un valor no válido en un momento dado. Las constantes facilitan la documentación del programa y lo hacen fácil de modificar. Una declaración tipo de constante podría ser la siguiente:

```
const double Pi = 3.14159; //Declara una constante de tipo double
```

```
const char msgError [] = "Verifique los datos introducidos" ; //Declara una constante cadena
```

Existen otros tipos de constantes y palabras reservadas relacionadas con constantes en C. De momento nos resultará suficiente el uso de `const` para el desarrollo de nuestros programas.

## EJERCICIO

Tomando como punto de partida el código que hemos visto de ejemplo de uso de funciones matemáticas, define un programa análogo que realice lo siguiente. En primer lugar define una variable de tipo double de nombre var1 y valor 15.75, una variable de tipo entero de nombre var2 y valor 45, y una variable de tipo double a la que denominaremos Pi, con el valor del número Pi expresado con cinco decimales. A continuación muestra el contenido de cada variable por pantalla. A continuación muestra por pantalla el resultado de multiplicar num1 por -1. A continuación muestra por pantalla el resultado de aplicarle la función valor absoluto a la multiplicación de num1 por -1. A continuación muestra el valor en radianes de la variable var2. Para ello debes realizar la operación matemática oportuna, si no recuerdas cómo hacerlo consulta en internet. A continuación muestra por pantalla el valor del seno del ángulo anterior, recuerda que debes expresarlo en radianes. A continuación muestra por pantalla el valor del coseno del ángulo anterior, recuerda que debes expresarlo en radianes. A continuación muestra por pantalla el redondeo al entero superior de var1. Finalmente muestra por pantalla el redondeo al entero inferior de var1.

Responde a las siguientes cuestiones:

- a) ¿Cuál es el código del programa que has definido?
- b) ¿Cuáles son los resultados que obtienes por pantalla al ejecutar el programa?

Para comprobar si tus respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

**Próxima entrega:** CU00521F

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=82&Itemid=210](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210)