

Librerías

Definición de librerías

Son archivos (no siempre externos) que nos permiten llevar a cabo diferentes tareas sin necesidad de preocuparnos por cómo se hacen sino simplemente entender cómo usarlas. Las librerías en C++ permiten hacer nuestros programas más modulares y reutilizables, facilitando además crear programas con funcionalidades bastante complejas en unas pocas líneas de código.

Nota: El nombre correcto en español de estos archivos son “archivos de biblioteca”, pero también se los conoce como “librerías”. Este es un error que proviene del nombre en inglés, que es library. Este término se traduce como biblioteca, y no como librería.

Declaración de librerías

Sintaxis para declarar Librerías en C++: La declaración de librerías se debe hacer al principio de todo nuestro código, antes de la declaración de cualquier función o línea de código.

#include <nombre de la librería>
#include "nombre de la librería".

Referencias y Descargas

- <http://www.cplusplus.com/info/>
- <http://en.cppreference.com/w/cpp/links/libs>

Librería de Matemáticas

- 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:

Librería de Matemáticas

Funciones	Significado	Funciones	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

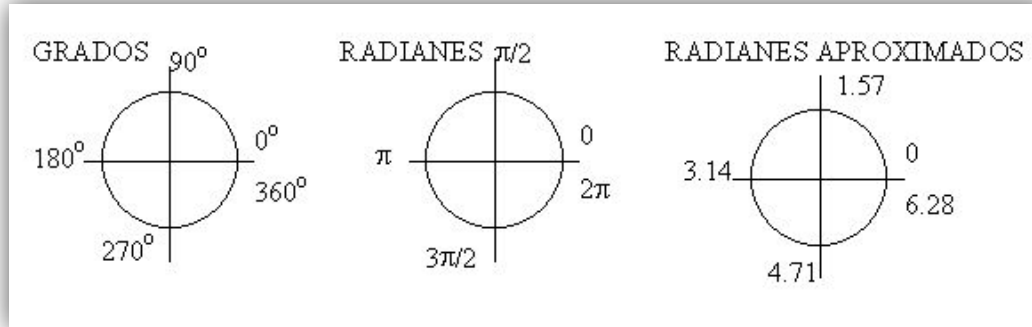
Librería de Matemáticas

La biblioteca de funciones matemáticas de C/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. Ejemplo del uso de raíz y potencia.

Las funciones relacionadas con ángulos trabajan en radianes. Por tanto, para operar con grados habremos de realizar la conversión oportuna. Recordemos que los grados y los radianes son dos diferentes sistemas para medir ángulos. Un ángulo de 360° equivale a 2π radianes; un ángulo de 180° equivale a π radianes +(recordemos que el número $\pi = 3.1415\dots$).

Librería de Matemáticas

Las equivalencias entre los cinco principales ángulos se muestran en las siguientes tres figuras:



Librería de Matemáticas

El número π está incluido en la librería de math.h, el mismo lo llamamos M_PI. Ejemplo del cálculo del seno.

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.

Librería de Time: Números Aleatorios

Función rand() int rand(void);

La función rand calcula un número aleatorio en el intervalo entre 0 y RAND_MAX. El valor de RAND_MAX es una constante predefinida que representa el mayor valor que puede devolver la función rand.

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    cout<<"Numero: "<<rand()<<"\n\n";
    return 0;
}
```

Librería de Time: Números Aleatorios

La función `rand` genera números a partir de un valor inicial o semilla. Si ejecutamos el programa anterior varias veces seguidas obtendremos siempre la misma secuencia de números. Para que esto no se produzca debemos cambiar el valor inicial de la semilla cada vez que se ejecute el programa. Esto se hace con la función `srand`.

Función `srand()`

```
void srand(unsigned int semilla);
```

Librería de Time: Números Aleatorios

La función `srand` fija el punto de comienzo para generar números aleatorios. El generador de números aleatorios obtiene los números en función del valor de su argumento. Cuando esta función no se utiliza, el valor del primer número generado siempre es el mismo para cada ejecución (corresponde a un argumento de valor 1).

Para obtener números aleatorios mediante estas funciones, es recomendable utilizar como semilla un valor que cambie constantemente, como por ejemplo, el valor devuelto por la función `time` (librería `ctime`). Esta función devuelve el número de segundos transcurridos desde las 0 horas del 1 de Enero de 1970.

Librería de Time: Números Aleatorios

Podemos comprobar el funcionamiento modificando el ejemplo anterior para que genere números diferentes en cada ejecución:

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    srand(time(NULL));
    cout<<"Numero: "<<rand()%20<<"\n\n";
    return 0;
}
```

Librería de Time: Números Aleatorios

Para generar números aleatorios entre dos límites DESDE – HASTA utilizaremos la siguiente expresión: **número aleatorio = desde + rand()% hasta**

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    int n=0;
    cout<<"Numeros random entre 10 y 20.\n\n";
    srand(time(NULL));
    n=10+rand()%20;
    cout<<"Numero: "<<n<<"\n\n";
    return 0;
}
```