

**PROGRAMACION BÁSICA**

**Tema: CREACIÓN BIBLIOTECA**

**Objetivo:** Al finalizar la clase el estudiante podrá crear librerías propias o personalizadas, que puedan ser implementadas posteriormente en cualquier programa.

**ACTIVIDAD EN CLASE**

Leer el siguiente material donde se explica que son librerías y los pasos a seguir para crear una biblioteca y desarrollar el programa bajo las indicaciones dadas.

Las librerías en C y C++ son archivos o bloques de código con funciones especiales que podemos importar o incluir en nuestros programas.

En el desarrollo de nuestros programas hasta ahora hemos utilizado muchas librerías que son predefinidas, es decir, vienen por defecto en el compilador de C y C++. Cada una de estas bibliotecas o librerías están clasificadas por el tipo de trabajos que hacen y cada una de las funciones en ellas definidas tiene una funcionalidad.

Algunas de éstas son:

- `iostream.h` (utilizado para operaciones de entrada/salida)
- `locale.h` (utilizada para modificar o consultar el escenario actual total del programa o partes del mismo)
- `math.h` (para operaciones matemáticas básica)
- `fstream` (para manejar archivos de entrada y salida)
- `string.h` (Para el manejo de cadena de caracteres )
- `windows.h` (contiene declaraciones de todas las funciones en el API de Windows), entre otras, donde cada una de las funciones que están definidas en estas bibliotecas o archivos de cabecera

La principal ventaja de crear nuestras propias librerías es la modularización de nuestros proyectos ya que nos permite compartir funciones o tipos de datos comunes entre varios códigos, permitiendo que nuestros programas puedan ser más legible

A continuación los pasos que se deben seguir para crear una librería personalizada.

1. En un **archivo fuente nuevo** escribe todas las funciones que te interesan, **incluyendo las cabeceras** que consideres necesarias para el buen funcionamiento de las mismas.
2. El **archivo creado** anteriormente, se debe **guardar con extensión .h** en la **carpeta include del compilador**. Ejemplo *milibreria.h*

**Nota1:** no ejecutarlo para que no le pueda cambiar la extensión

**Nota2:** Si la biblioteca no será de uso frecuente, se puede guardar en la misma carpeta del código del programa que queramos compilar.

3. **Llamar a la biblioteca en el programa.** Para llamar a la biblioteca creada deberemos colocar en la cabecera del programa a compilar, junto a las demás librerías o bibliotecas, considerando que **si la biblioteca se guardó en la carpeta include se llamará así:** `#include <milibreria.h>`

Si la biblioteca se guardó en la misma carpeta donde está el código a ejecutar se llamará así:

```
#include "milibreria.h"
```

4. **Uso de las funciones de la biblioteca:** Como es una librería, a partir de este momento es posible utilizar las funciones de *milibreria.h* sin necesidad de realizar o copiar funciones de uso frecuente.

## Ejercicio en Clase

Implementar un menú con las siguientes opciones

1. **Potencia**
2. **Suma de Dígitos de un número**
3. **Suma de divisores propios de un número**
4. **Cantidad de dígitos de un número**
5. **Cantidad de divisores propios de un número**
6. **Número primo**
7. **Número perfecto**
8. **Números amigos**
9. **Numero Narcisista**
10. **Numero Abundante o deficiente**

**Nota1:** Un **número narcisista o número de Armstrong** es aquel que es igual a la suma de cada uno de sus dígitos elevados a la "n" potencia (donde "n" es el número de cifras del número). La metáfora de su nombre alude a lo mucho que parecen "quererse a sí mismos" estas cifras.

### Ejemplos:

El **153** es un número narcisista puesto que:

$$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153.$$

Otros números narcisistas son: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474

**Nota2:** Un número es **ABUNDANTE**, cuando la suma de sus divisores propios es mayor que él.

**Ejemplo:** El número 12 es **abundante** porque sus divisores, 1, 2, 3, 4 y 6 suman 16 que es mayor que 12.

Un número es **DEFICIENTE**, cuando la suma de sus divisores propios es menor que él.

**Ejemplo:** El número 10 es **deficiente** porque sus divisores, 1, 2 y 5 suman 8 que es menor que 10

# Práctica # 1

## Nombre de la Práctica: Librerías Personalizadas

**Objetivo:** Al finalizar esta práctica el estudiante será capaz de:

1. Resolver problemas grandes y complejos dividiendo un problema en subproblemas a través del uso de funciones
2. Crear sus Librerías propias o personalizadas

A partir de la explicación dada donde se explicaron los pasos para crear librerías propias, desarrollar un aplicativo para una empresa de manejo de divisas.

Debe implementar un menú que permita hacer cambio de peso a otra moneda (dólar, euro, yen, libra, peso mexicano y viceversa)

Aplicando librerías propias, crear una biblioteca con las funciones pertinentes para cambio de peso a moneda otra moneda y viceversa), de igual forma para calcular e indicar la cantidad de billetes de la denominación más alta a la más baja para cada moneda, de acuerdo al valor al cambio.