



## Práctica 4. Funciones con retorno y sin retorno.

Miércoles 23 de agosto del 2023

**Instrucciones:** Desarrollen el código en lenguaje C y elaboren el diagrama de flujo correspondiente para los ejercicios. Será suficiente con un archivo .cpp que contenga todos los ejercicios organizados en un menú implementado mediante una estructura switch.

### Problemas:

1. **Fibonacci sin Recursión:** Crea un programa que calcule y muestre los primeros n términos de la serie de Fibonacci sin utilizar recursión.

Instrucciones:

- En la función main, solicita al usuario que ingrese el valor de n, que representará el término de la serie de Fibonacci que desea calcular.
  - calcularFibonacci: Esta función debe tomar un número entero como entrada y calcular el término n de la serie de Fibonacci. El resultado debe mostrarse en pantalla de manera clara, indicando cuál es el término n de la serie. No debe devolver ningún valor, solo mostrar el resultado.
  - Utiliza la función calcularFibonacci para que calcule el término n de la serie de Fibonacci.
  - Asegúrate de que el programa maneje correctamente los casos base (n igual a 0 o 1).
  - Pregunta al usuario si desea calcular el término de la serie de Fibonacci para otro valor de n. Si es así, permite al usuario ingresar otro valor y calcular el término correspondiente. Si no, muestra un mensaje de despedida y termina el programa.
2. **Fibonacci con Recursión:** Crea un programa en C que calcule y muestre el término n de la serie de Fibonacci utilizando una función recursiva.

Instrucciones:

- Implementa una función llamada calcularFibonacciRecursion que tome un número entero n como argumento y devuelva el término n de la serie de Fibonacci.
- En la función main, solicita al usuario que ingrese el valor de n, que representará el término de la serie de Fibonacci que desea calcular.



- Utiliza la función `calcularFibonacciRecursion` para calcular el término  $n$  de la serie de Fibonacci.
  - Muestra el resultado en pantalla de manera clara, indicando cuál es el término  $n$  de la serie de Fibonacci.
- 
- Asegúrate de que el programa maneje correctamente los casos base, es decir, cuando  $n$  es igual a 0 o 1, y devuelve el valor correspondiente.
  - Pregunta al usuario si desea calcular el término de la serie de Fibonacci para otro valor de  $n$ . Si es así, permite al usuario ingresar otro valor y calcular el término correspondiente. Si no, muestra un mensaje de despedida y termina el programa.
3. **Triángulo de Pascal:** Crea un programa en C que genere y muestre las primeras  $n$  filas del Triángulo de Pascal.

Instrucciones:

- Implementa una función llamada `generarTrianguloPascal` que tome un número entero  $n$  como argumento y muestre el Triángulo de Pascal con  $n$  filas.
- En la función `main`, solicita al usuario que ingrese el número de filas  $n$  que desea en el Triángulo de Pascal.
- Utiliza la función `generarTrianguloPascal` para generar y mostrar el Triángulo de Pascal con las  $n$  filas especificadas.
- Asegúrate de que el Triángulo de Pascal se muestre de manera clara y legible, con los números alineados correctamente.
- Pregunta al usuario si desea generar el Triángulo de Pascal con otro número de filas  $n$ . Si es así, permite al usuario ingresar otro valor y generar el Triángulo correspondiente. Si no, muestra un mensaje de despedida y termina el programa.



Documentar el código de la siguiente manera:

✓ Comentarios de encabezado:

/\*

Nombre del archivo: mi\_programa.c

Autor: Tu Nombre

Fecha de creación: 12 de septiembre de 2023

Descripción: Este archivo contiene el código principal de mi programa.

\*/

✓ Comentarios en función:

/\*

Función: suma

Descripción: Esta función suma dos números enteros.

Parámetros:

- num1: El primer número entero a sumar.

- num2: El segundo número entero a sumar.

Valor de retorno: La suma de num1 y num2.

\*/

```
int suma(int num1, int num2) {  
    return num1 + num2;  
}
```

**Problemas:**

Se deberá crear un informe que incluya tanto los códigos fuente como los diagramas de flujo correspondientes. Este documento deberá contar con una portada adecuada. Los programas en formato .cpp serán sometidos a revisión a través de GitHub, por lo tanto, es necesario agregar el enlace al repositorio donde se encuentra la práctica. Por favor, asegúrense de seguir el formato de nomenclatura tanto para el archivo de la práctica como para el nombre del repositorio. El formato para el archivo será "Practica1\_NombrePractica\_Apellidos\_Nombre".