

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Desarrollo de Software y Tecnologías Emergentes

Lenguaje C

ALUMNO: Barrios Retana Lizeth

MATRÍCULA: 372813

GRUPO: 932

PROFESOR: Yulith Vanessa Altamirano Flores

Ensenada, Baja California a 10 de octubre de 2023.

# Práctica 4. Funciones con retorno y sin retorno

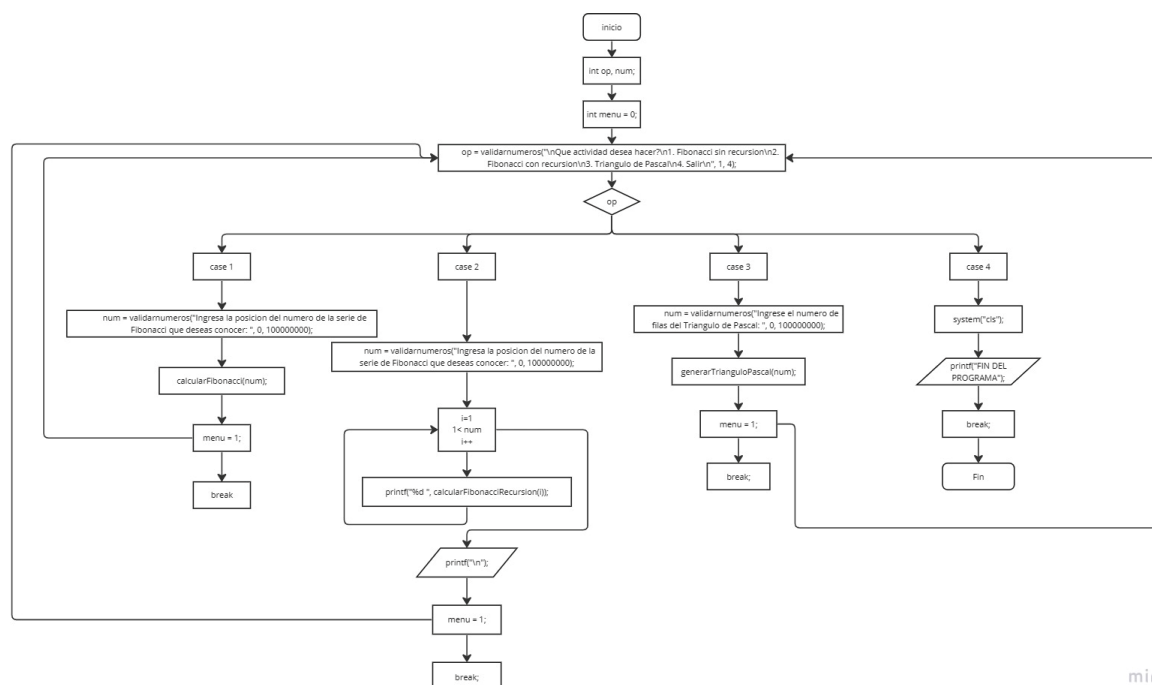
## instrucciones

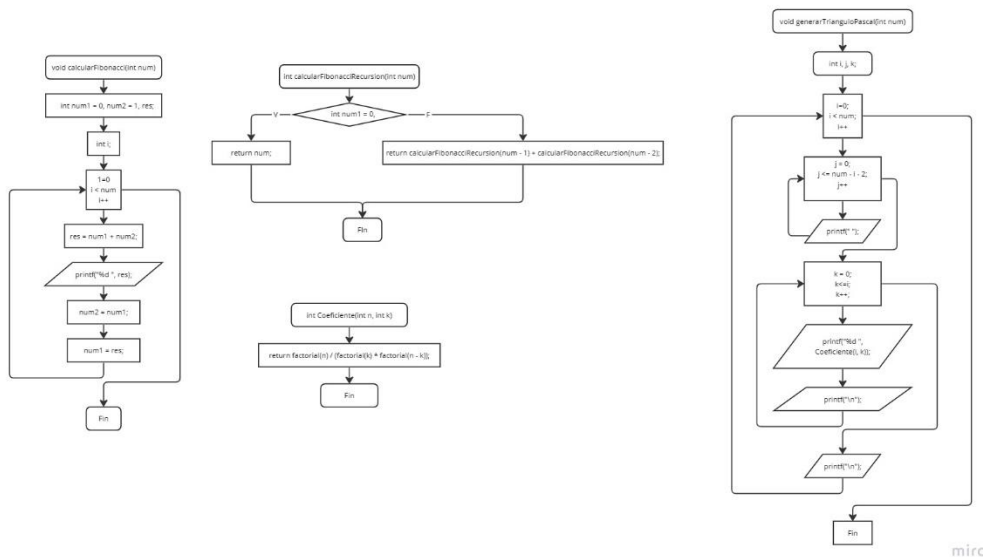
Desarrollen el código en lenguaje C y elaboren el diagrama de flujo correspondiente para los ejercicios. Será suficiente con un archivo .cpp que contenga todos los ejercicios organizados en un menú implementado mediante una estructura switch.

## Repositorio

[https://github.com/LizBarrios/practica4\\_FuncionesConRetornoySinRetorno\\_BarriosRetana\\_Lizeth.git](https://github.com/LizBarrios/practica4_FuncionesConRetornoySinRetorno_BarriosRetana_Lizeth.git)

## Diagrama de Flujo





## 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.

```

/*
Nombre del archivo: Practica4.c
Autor: Lizeth Barrios Retana
Fecha de creación: 20 de septiembre de 2023
Descripción: programa que contiene un menu donde hay 3 opciones: uno que hace los
numero de fibonacci sin recursion, fibonacci con recursion y el triangulo de pascal.
*/
#include <stdio.h>
#include "BRL.h"

int calcularFibonacciRecursion(int num);
void calcularFibonacci(int num);
int Coeficiente(int n, int k);
void generarTrianguloPascal(int num);

int main()
{

```

```

int op;
int menu = 0;
int num;

do
{
    op = validarnumeros("\nQue actividad desea hacer?\n1. Fibonacci sin recursion\n2.
Fibonacci con recursion\n3. Triangulo de Pascal\n4. Salir\n", 1, 4);
    switch (op)
    {
        case 1:
            num = validarnumeros("Ingresa la posicion del numero de la serie de Fibonacci
que deseas conocer: ", 0, 100000000);
            calcularFibonacci(num);
            menu = 1;
            break;
        case 2:
            num = validarnumeros("Ingresa la posicion del numero de la serie de Fibonacci
que deseas conocer: ", 0, 100000000);
            for (int i = 1; i <= num; i++)
            {
                printf("%d ", calcularFibonacciRecursion(i));
            }
            printf("\n");
            menu = 1;
            break;
        case 3:
            num = validarnumeros("Ingrese el numero de filas del Triangulo de Pascal: ", 0,
100000000);
            generarTrianguloPascal(num);
            menu = 1;
            break;
        case 4:
            system("cls");
            printf("FIN DEL PROGRAMA");

            return 0;
            break;
    }
} while (menu);
}

/*
Función: calcularFibonacci
Descripción: calcula los numero de fibonacci.
Parámetros:
    - num1: El primer número de la sucesion.
    - num2: El segundo número de la sucesion.
    - res: la suma del num1 y num2.
Valor de retorno: solo imprime los numeros.
*/

void calcularFibonacci(int num)
{
    int num1 = 0, num2 = 1, res;
    int i;
    for (i = 0; i < num; i++)
    {

```

```

        res = num1 + num2;
        printf("%d ", res);
        num2 = num1;
        num1 = res;
    }
}

/*
Función: calcularFibonacciRecursion
Descripción: calcula los numero de fibonacci con recursion.
Parámetros:
    - num: es el numero que ingresa el usuario.
Valor de retorno: retorna los numeros de de la sucesion.
*/

int calcularFibonacciRecursion(int num)
{
    if (num <= 1)
    {
        return num;
    }
    else
    {
        return calcularFibonacciRecursion(num - 1) + calcularFibonacciRecursion(num - 2);
    }
}

/*
Función: factorial
Descripción: calcula el factorial para el triangulo de pascal.
Parámetros:
    - n: el numero que contiene i.
Valor de retorno: i modificada.
*/

int factorial(int n)
{
    if (n <= 1)
    {
        return 1;
    }
    else
    {
        return n * factorial(n - 1);
    }
}

/*
Función: coeficiente
Descripción: calcula el coeficiente para el triangulo de pascal.
Parámetros:
    - n: el numero que contiene i.
    - k: el numero que contiene k.
Valor de retorno: operacion realizada con la funcion de factorial.
*/

int Coeficiente(int n, int k)
{

```

```

    return factorial(n) / (factorial(k) * factorial(n - k));
}

/*
Función: generarTrianguloPascal
Descripción: genera el triangulo.
Parámetros:
    - num: es el numero que ingresa el usuario.
    - i: recorre los numeros para las lineas.
    - j: recorre los numeros para imprimir.
    - k: recorre el valor para imprimir los numeros.
Valor de retorno: los numeros y las filas del triangulo.
*/

void generarTrianguloPascal(int num)
{
    int i, j, k;
    for (i = 0; i < num; i++)
    {
        for (j = 0; j <= num - i - 2; j++)
        {
            printf(" ");
        }
        for (k = 0; k <= i; k++)
        {
            printf("%d ", Coeficiente(i, k));
        }
        printf("\n");
    }
}

```