Universidad Autónoma de Baja California

Ingeniería en Software y Tecnologías Emergentes.

Práctica 4. Funciones con retorno y sin retorno.

Materia: Lenguaje C

Maestro: Yulith Altamirano

Alumno: Diego Quiros 372688

Fecha: 09/21/2023

Práctica 4. Funciones con retorno y sin retorno.

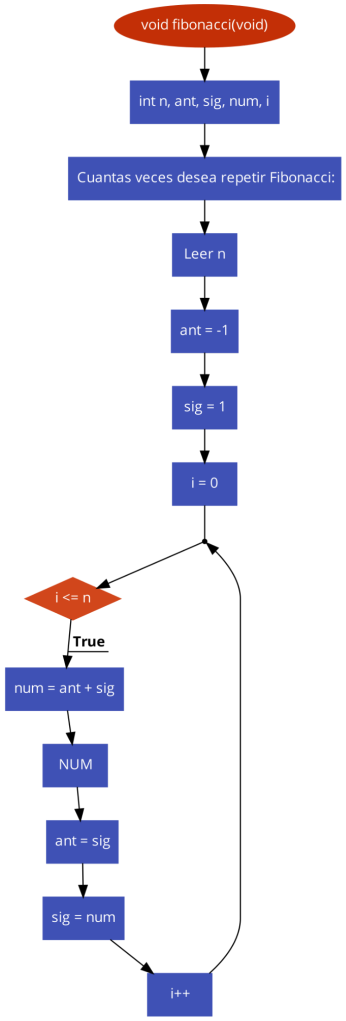
Instrucciones:

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:

```
C/C++
https://github.com/diegovq12/LenguajeC-QVDD_932/tree/main/Practica%204
```

Diagramas de flujo Ejercicio 1:



Problemas Ejercicio 1:

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

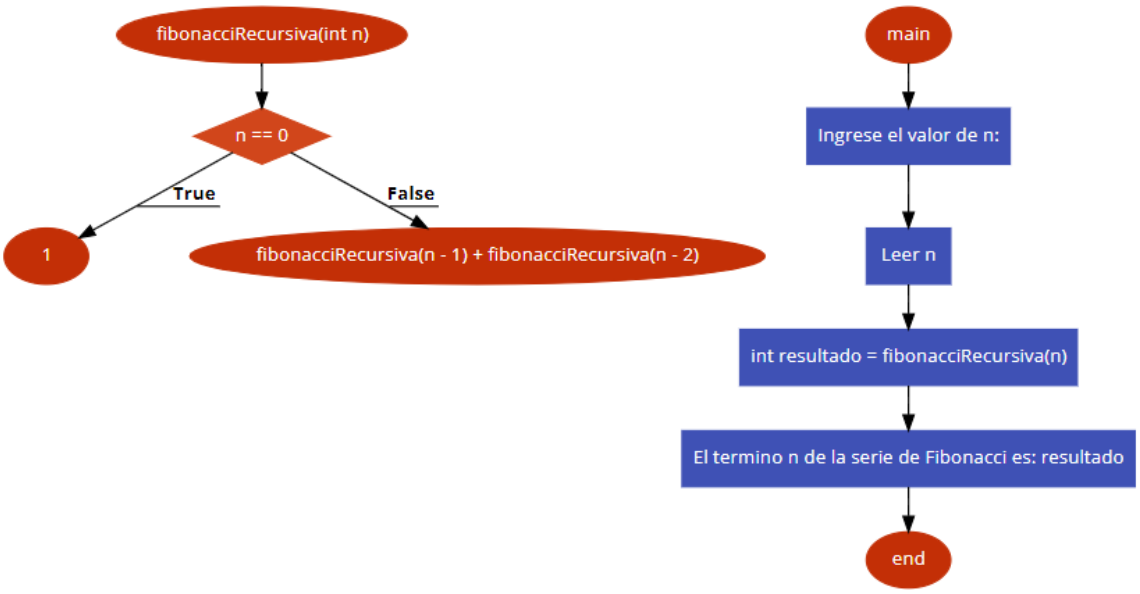
o En la función main, solicita al usuario que ingrese el valor de n, que representará el término de la serie de Fibonacci que desea calcular.

```C/C++
printf("Cuantas veces desea repetir Fibonacci: ");
  scanf("%d", &n);
```

o calcular Fibonacci: 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.

```C/C++
void fibonacci(void)
{
  int n, ant, sig, num, i;
  printf("Cuantas veces desea repetir Fibonacci: ");
  scanf("%d", &n);
  ant = -1;
  sig = 1;
  for (i = 0; i <= n; i++)
  {
    num = ant + sig;
    printf(" %d ", num);
    ant = sig;
    sig = num;
  }
}
```

Diagramas de flujo Ejercicio 2:



Problemas Ejercicio 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.
o Implementa una función llamada calcularFibonacciRecursion que tome un número entero n como argumento y devuelva el término n de la serie de Fibonacci.

```C/C++
int fibonacciRecursiva(int n)
{
  if (n == 0)
  {
    return 1;
  }
  else
  {
    return (fibonacciRecursiva(n - 1) + fibonacciRecursiva(n - 2));
  }
}
```

o 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.

```C/C++
printf("Ingrese el valor de n: ");
      scanf("%d", &n);
```

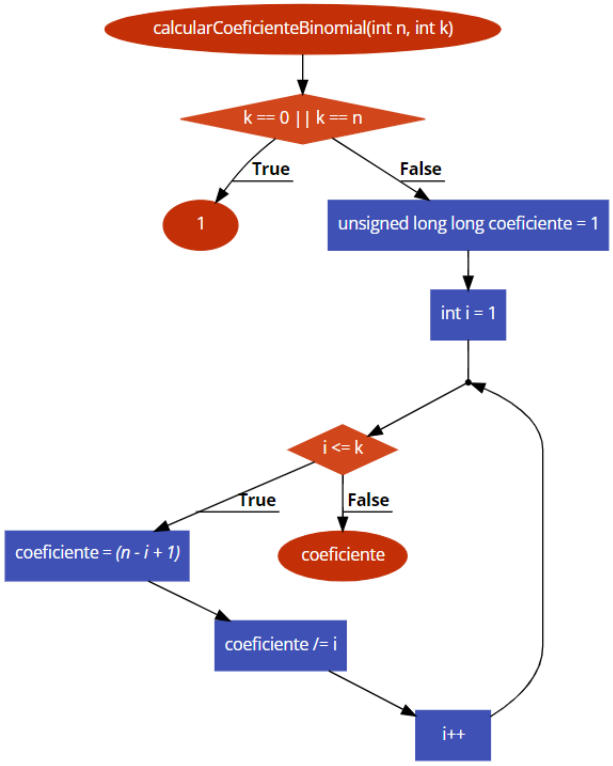
o Utiliza la función calcularFibonacciRecursion para calcular el término n de la serie de Fibonacci.

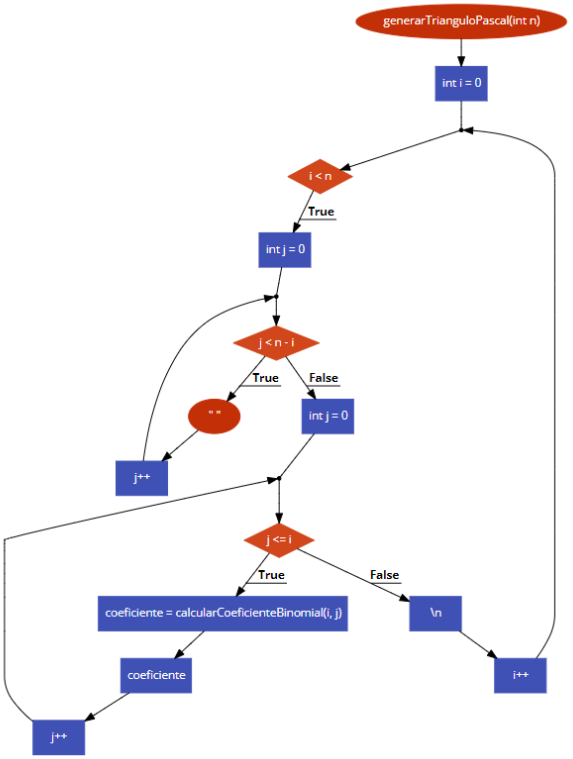
```
C/C++
int resultado = fibonacciRecursiva(n);
```

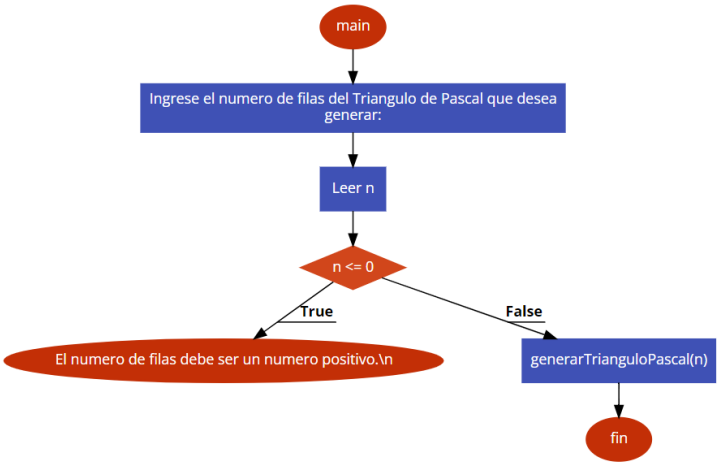
o Muestra el resultado en pantalla de manera clara, indicando cuál es el término n de la serie de Fibonacci.

```
C/C++
printf("El termino %d de la serie de Fibonacci es: %d\n", n, resultado);
```

Diagramas de flujo Ejercicio 3:







Problemas Ejercicio 3:

Triángulo de Pascal: Crea un programa en C que genere y muestre las primeras n filas del Triángulo de Pascal.

o Implementa una función llamada generarTrianguloPascal que tome un número entero n como argumento y muestre el Triángulo de Pascal con n filas.

```C/C++
void generarTrianguloPascal(int n)
{
  for (int i = 0; i < n; i++)
  {
    for (int j = 0; j < n - i; j++)
    {
      printf(" ");
    }

    for (int j = 0; j <= i; j++)
    {
      unsigned long long coeficiente = calcularCoeficienteBinomial(i, j);
      printf("%llu ", coeficiente);
    }

    printf("\n");
  }
}
```

o En la función main, solicita al usuario que ingrese el número de filas n que desea en el Triángulo de Pascal.

```C/C++
printf("Ingrese el numero de filas del Triangulo de Pascal que desea
generar: ");
 scanf("%d", &n);
```

o Utiliza la función generarTrianguloPascal para generar y mostrar el Triángulo de Pascal con las n filas especificadas.

```C/C++
if (n <= 0)
            {
                printf("El numero de filas debe ser un numero positivo.\n");
            }
            else
            {
                generarTrianguloPascal(n);
                break;
```