



Introducción

Para mejorar el entendimiento de los elementos de programación, se proponen problemas para los cuales deben: analizar, crear una solución, un diagrama de flujo y el programa en lenguaje C.

Objetivos

1. Implementar una solución numérica ante el problema que se le presente.
2. Crear la documentación necesaria para cada programa elaborado.
3. Aplicar los conocimientos adquiridos sobre el lenguaje C.
4. Utilizar solo las funciones aprendidas en clase.
5. Almacenar una copia de su código en GitHub.

Requerimientos

- Computador con sistema operativo Windows, Linux o MACOS y conexión a internet.
- Maquina virtual con una distribución de Linux (en caso de estar utilizando Windows).
- Cuenta de GitHub.

Implementación

Los elementos que se deben de presentar por cada programa realizado son los siguientes:

1. Enunciado del problema
2. Metodología para solucionar el problema
3. Entradas y Salidas
4. Diagrama de Flujo para 3 problemas, Pseudocódigo para los otros 3
5. Código del programa bajo los lineamientos planteados en clase.

El código de cada programa debe de estar en su repositorio de la plataforma GitHub. Este se debe de entregar en un documento PDF tamaño carta. El cual se debe de subir al Classroom, el código no es necesario que este en el documento, pero debe de haber un link el cual dirija a su GitHub.

Ejemplo: Fibonacci

1) Enunciado: Realizar un programa el cual solicite un numero y este devuelva la serie de fibonacci para este numero.

2) Metodología: La serie de fibonacci inicia desde el numero 0 y esta suma dos números para dar un tercero, asumamos que se debe obtener la serie de fibonacci para un valor $n=4$, esta se representa de la siguiente forma

0 1 1 2 3 5 8 = 0 1 (0 + 1) (1 + 1) (1 + 2) (2 + 3) (3 + 5)

La función general tiene dos partes, las condiciones iniciales y la suma de los elementos

$$f(a,n) = \begin{cases} a_1 = a_2 = 1 \\ a_n = a_{n-1} + a_{n-2}, n \geq 3 \end{cases}$$

Para las condiciones iniciales como estas no cambian simplemente podemos validar que se cumplan. Para las operaciones siguientes, necesitamos un set de datos en un tipo de vector para que se pueda ir navegando entre las posiciones de estos y realizar las operaciones. Dado a que la respuesta del numero siguiente es la operación de los anteriores, estamos bajo las condiciones de un elemento recursivo

0 1 (0+1) (1 + (0 + 1)) ((0 + 1) + (1 + (0 + 1))) ...

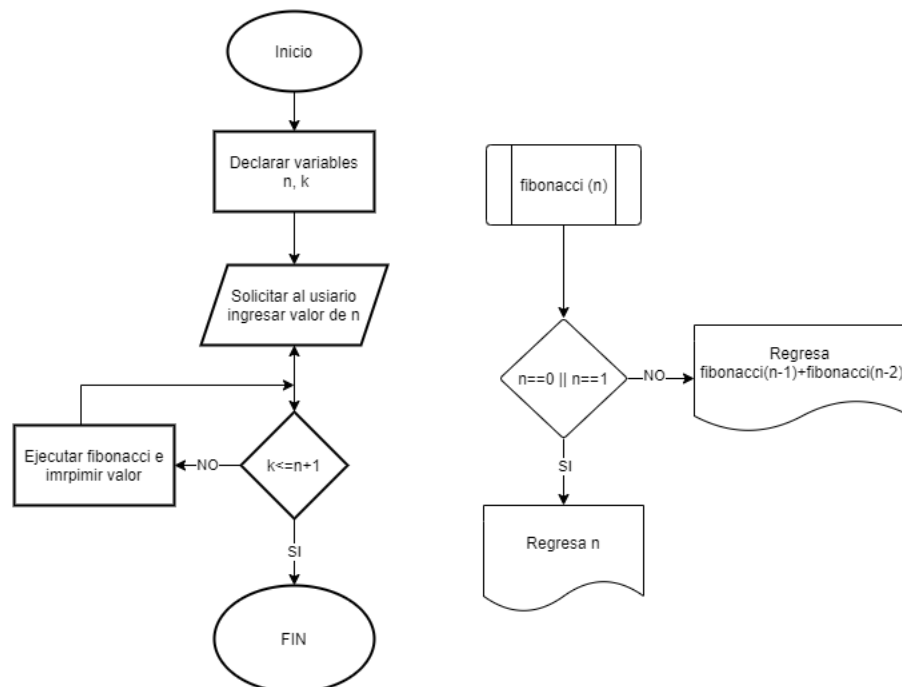
3) Variables de Entrada y salida: Debido a que solicitaremos el numero n , el usuario ingresara un valor entero, como salida obtendremos el numero entero de la serie en forma de texto.

→ entero (n)

← texto (valor)

* variable local de conteo (k)

4) Diagrama de flujo:



5) Código:

```
/*
Autor:      Maynor Ballina
Fecha:      Fri Mar 19 12:20:43 CST 2021
compilador: gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Para Compilar: gcc -o fibonacci fibonacci.c
Version: 1.0
librerias:  stdio (u otras)
Resumen: Ejemplo de funciones recursivas directas para obtener los numeros de fibonacci
*/

//Librerias
#include <stdio.h>
//Declaracion e inicializacion de variables globales
//prototipo de la funcion usar punto y coma
long fibonacci(int n);

int main(){
    //n valor de numero al que queremos ejecutar la secuencia de fibonacci
    //k variable de conteo para incrementar o decrementar
    int n, k;
    //solicita el valor de numero hasta que este sea mayor o igual a 1
    do{
        printf("introducir el valor de n: ");
        scanf("%d",&n);
    }while(n<=1);
    puts ("serie fibonacci \n");
    printf("0 ");
    printf("1 ");
    //se realiza la ejecucion de los numeros de fibonacci por cada secuencia, es decir
    //la variable k contara hasta llegar al valor de n
    for (k=2; k<=n+1; k++){
        //por cada valor de k se ejecuta la funcion fibonacci
        //se imprime una Cadena de caracteres los cuales se acumulan hasta cumplir la sentencia
        del if donde al
        //llegar a 7 caracteres ingresa un enter.
        printf("%d %s ", fibonacci(k), (k %7==0 ? "\n": " "));
    }

    puts ("\n");
    return 0;
}

//funcion fibonacci la cual se llama a si misma dependiendo del numero que ingresa en ella
long fibonacci(int n){
    //si el valor es 0 o 1 solo regresa este valor
    if (n==0 || n==1)
        return n;
    else
        //se utiliza el algoritmo de la secuencia de fibonacci
        return fibonacci(n-1)+fibonacci(n-2);
}
```

Otra forma de presentarlo: Ojo el texto es un link hacia el github del archivo

El código se encuentra en el github:

Usuario Mballina42, Repositorio LabSimu1s2021, Archivorecursividad_directa.c

Problemas

1. Debe de ingresar un vector de 10 elementos, llenarlo de números pares del 2 al 20. Al iniciar el programa debe preguntar al usuario como quiere ver los números, el menú debe de ser por medio de caracteres: “a”verlos de forma ascendente, “d”descendente, en caso que el usuario escriba otro valor debe de decir que no es correcto y preguntarle el carácter nuevamente, hasta que este sea el correcto, al ingresar el valor correcto muestra el vector en pantalla y termina el programa.
2. Crear un programa que solicite al usuario 5 números enteros, estos se deben de guardar en un vector, al terminar de guardar los valores, el programa debe de ordenarlos de forma ascendente y mostrar el vector ordenado. (utilice un método de ordenación.)
3. Crear un programa que solicite al usuario dos posiciones en coordenadas (x,y,z) al obtenerlas debe de almacenarlas en dos vectores, el programa automáticamente debe de mostrar los siguientes resultados:
 - a) magnitud de cada vector
 - b) suma de los dos vectores
 - c) producto escalar
 - d) producto vectorial
4. Crear un programa que solicite al usuario dos matrices de 3X3 almacenarlas como (matA, matB) y una constante, el programa automáticamente debe de mostrar las los siguientes resultados:
 - a) matA por constante
 - b) suma de las dos matrices
 - c) resta de las dos matrices
 - d) multiplicación de las dos matrices
 - e) determinante de matA
 - f) transpuesta de matB
 - g) inversa de matA
 - h) reducción de Gauss de matA
 - i) reducción de Gauss Jordan matB
5. Crear un programa que encuentre el factorial de un numero entero ingresado, debe de utilizar una función recursiva.
6. Crear un programa que realice la sumatoria desde 1 hasta un numero n que ingrese el usuario de las siguientes funciones.

a)

$$\sum_{k=1}^n k^2(k-3)$$

b)

$$\sum_{k=1}^n \frac{3}{k-1}$$

c)

$$\sum_{k=1}^n \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n$$

d) sucesión de Bode:

$$\sum_{k=2}^n 0.1(3 * 2^{k-2} + 4)$$