

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

Alumno

Córdova Fernández Karla Lilia

Unidad de Aprendizaje: Programación
Avanzada

Profesor

M. en C. Niels Henrik Navarrete Manzanilla

Práctica 1

Uso de "scanf"

Ciudad de México; a 24 de octubre de 2020.

Índice

INTRODUCCIÓN	4
DESARROLLO	5
PROGRAMA 1. Secuencia ordenada	5
Descripción.....	5
Análisis.....	5
Requerimientos Funcionales	6
Requerimientos no Funcionales	6
Diagrama de flujo	7
Código en C	8
Resultados (compilación)	9
PROGRAMA 2. Tabla de multiplicar	10
Descripción.....	10
Análisis.....	10
Requerimientos Funcionales	11
Requerimientos no Funcionales	11
Diagrama de flujo	12
Código en C	13
Resultados (compilación)	14
PROGRAMA 3. Mayor y menor	15
Descripción.....	15
Análisis.....	15
Requerimientos Funcionales	16

Requerimientos no Funcionales	16
Diagrama de flujo	17
Código en C	18
Resultados (compilación)	19
PROGRAMA 4. Tres números comparados	20
Descripción.....	20
Análisis.....	20
Requerimientos Funcionales	21
Requerimientos no Funcionales	21
Diagrama de flujo	22
Código en C	23
Resultados (compilación)	24
PROGRAMA 5 División con restas	25
Descripción.....	25
Análisis.....	25
Requerimientos Funcionales	26
Requerimientos no Funcionales	26
Diagrama de flujo	27
Código en C	28
Resultados (compilación)	29
CONCLUSIONES	30

INTRODUCCIÓN

La función `scanf` en C sirve para leer los datos de entrada y almacenarlos en los argumentos que se pasan de referencia. La estructura de ésta función es `scanf("% tipo de dato ", &variable);`

Para el llenado de arreglos, también se puede utilizar dentro de un `for` o `for` anidados dependiendo si se trata de un array unidimensional o bidimensional.

En ésta práctica se utilizará para el traspaso de datos numéricos desde teclado, combinándolo con funciones de ciclos, condiciones, ternas, método burbuja y otros métodos vistos en clase.

DESARROLLO

PROGRAMA 1. SECUENCIA ORDENADA

Descripción

Crear un programa que reciba una secuencia de números (1,2,3....n), se debe obtener como salida los números ingresados por el usuario de acuerdo al orden (posición de cada uno de los números) y después mostrarlos en orden descendiente (n....3,2,1).

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

int longitud; *variable que indica la cantidad de números en la secuencia.*

float grupo[longitud]; *arreglo donde se almacenará la secuencia de números.*

Salidas:

float grupo [longitud];

2. ¿Qué es lo que hará el programa?

Se solicitarán primero la cantidad de números de la secuencia y los elementos, que se almacenarán en un entero y un arreglo de flotantes.

Se imprimirán todos los elementos conforme fueron guardados.

Usando el método burbuja, los elementos del arreglo se acomodarán en orden descendente (dos for's anidados, donde si un número en una posición es menor que el siguiente, se intercambian de posición).

Se imprimirá el arreglo en el nuevo orden.

3. ¿Qué espero de salida?

Imprimir -> float grupo [longitud]; en orden de introducción y ordenados descendientemente.

Requerimientos Funcionales

El programa permite introducir una secuencia de números enteros y flotantes (positivos o negativos) de longitud finita preestablecida por el usuario.

El programa muestra en pantalla la cadena en el orden de introducción y acomodados en orden descendente.

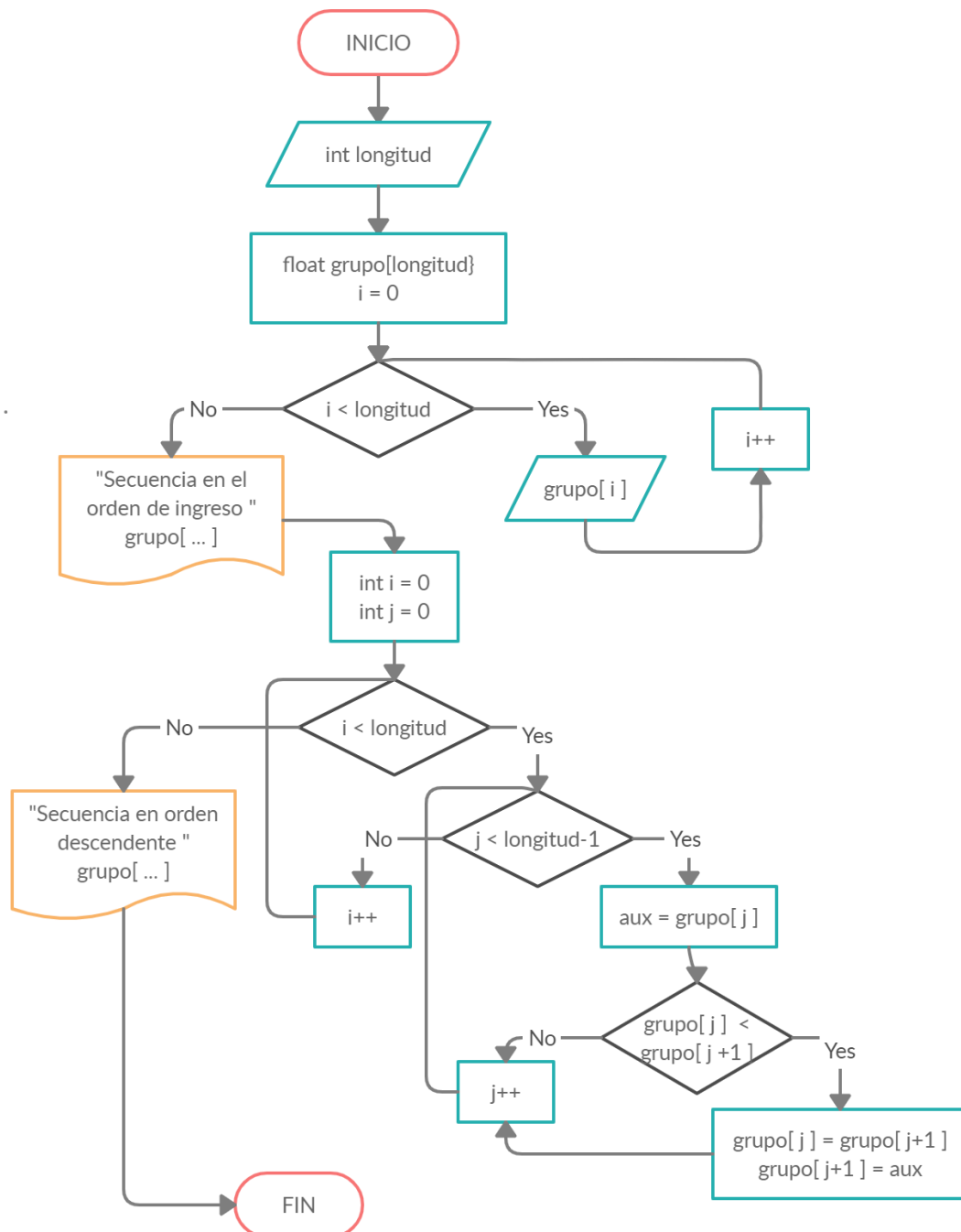
El programa se repite hasta que el usuario de en la opción de salir.

Requerimientos no Funcionales

El programa no permite introducir una secuencia de números que supere o sea menor a la longitud establecida inicialmente por el usuario.

El programa sólo imprime los números con dos decimales.

Diagrama de flujo



Código en C

```
#include <stdlib.h>
#include <stdio.h>

int main(){
    int longitud = 0, indice = 0;
    float aux = 0;

    while(indice != 2){
        printf("\nElija una opcion:\n1) Ordenar una secuencia de numeros.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);
        system("cls");

        if(indice == 1){
            printf("Cuantos numeros quiere en la secuencia?\n");
            scanf("%d",&longitud);
            fflush(stdin);

            float grupo[longitud] = {};
            printf("Ingrese los numeros:");
            for(int i=0; i<longitud; i++){
                scanf("%f",&grupo[i]);
            }

            fflush(stdin);
            printf("\nSecuencia en el orden de ingreso:\n");
            for(int i=0; i<longitud; i++){
                printf("%.2f",grupo[i]);
            }

            //ORDEN DESCENDENTE
            for(int i=0; i<longitud; i++){
                for(int j=0; j<longitud-1; j++){
                    aux = grupo[j];
                    if(grupo[j] < grupo[j+1]){
                        grupo[j] = grupo[j+1];
                        grupo[j+1] = aux;
                    }
                }
            }

            //IMPRIMIR
            printf("\n\nSecuencia en orden descendente:\n");
            for(int i=0; i<longitud; i++){
                printf("%.2f",grupo[i]);
            }
            printf("\n\n");
        }
    }

    return 0;
}
```


Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practica 1\Practica1_1_Ordenamiento secuen
Cuantos numeros quiere en la secuencia?
10
Ingrese los numeros:1
3
8.6
9
0.2
-1.5
-4
9
25
4.1

Secuencia en el orden de ingreso:
[1.00],[3.00],[8.60],[9.00],[0.20],[-1.50],[-4.00],[9.00],[25.00],[4.10],

Secuencia en orden descendente:
[25.00],[9.00],[9.00],[8.60],[4.10],[3.00],[1.00],[0.20],[-1.50],[-4.00],

Elija una opcion:
1) Ordenar una secuencia de numeros.
2) Salir.
```

longitud = 10

Secuencia inicial

grupo[] = { 1 , 3 , 8.6 , 9 , 0.2 , -1.5 , -4 , 9 , 25 , 4.1 }

Secuencia ordenada

grupo[] = { 25 , 9 , 9 , 8.6 , 4.1 , 3 , 1 , 0.2 , -1.5 , -4 }

PROGRAMA 2. TABLA DE MULTIPLICAR

Descripción

Realizar la tabla de multiplicar dado x's número.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

int numero; *variable del número que se va a multiplicar.*

int maximo; *variable del factor que será el máximo en la tabla.*

Salidas:

int mult; *todas las multiplicaciones del número.*

2. ¿Qué es lo que hará el programa?

Utilizando un ciclo for, el número del que se busca obtener la tabla se multiplicará por todos los factores enteros que van desde el uno hasta el valor introducido como máximo.

3. ¿Qué espero de salida?

Imprimir -> int mult; *el número de entrada multiplicado por todos los valores enteros de 1 hasta el número máximo.*

Requerimientos Funcionales

El programa permite obtener la tabla de multiplicar de cualquier número entero positivo o negativo.

El programa permite decidir hasta qué número desea el usuario que llegue la tabla de multiplicar (la tabla siempre iniciará desde uno).

El programa se repite hasta que el usuario de en la opción de salir.

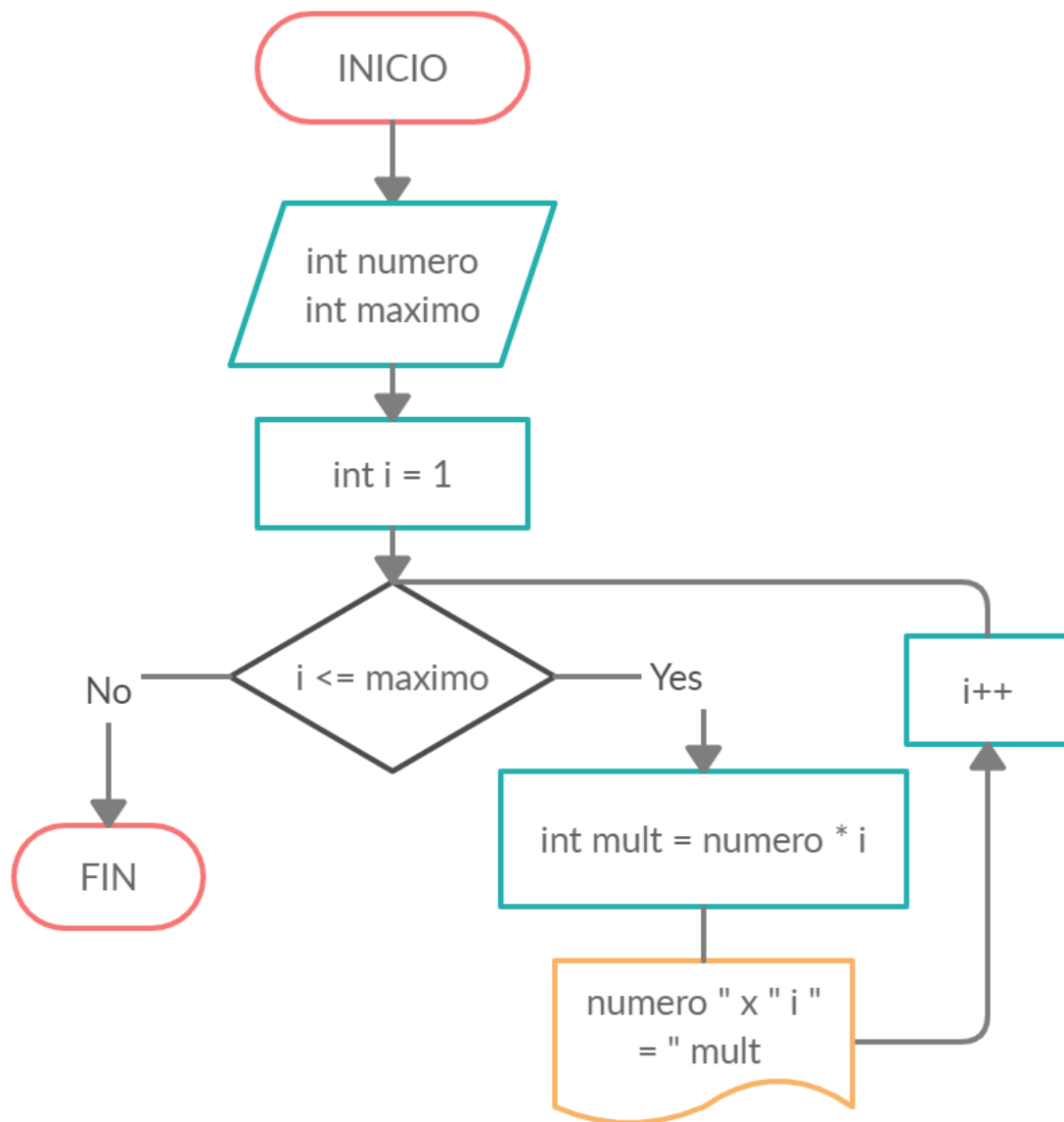
Requerimientos no Funcionales

El programa no acepta números decimales o flotantes tanto en el número como en el factor máximo.

El programa no permite que el factor máximo sea menor a uno.

El programa no pregunta el factor mínimo, siempre empieza desde uno.

Diagrama de flujo



Código en C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

    int numero = 0, maximo = 0, mult = 0, indice = 0;

    while(indice != 2){
        printf("\nElija una opcion:\n1) Tabla de multiplicar.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);

        if(indice == 1){
            printf("Ingrese el numero del que quiere obtener la tabla de multiplicar:\n");
            scanf("%d",&numero);
            fflush(stdin);

            printf("Hasta que numero desea la tabla de multiplicar?:\n");
            scanf("%d",&maximo);
            fflush(stdin);

            for(int i=1; i<=maximo; i++){
                mult = numero*i;
                printf("%d x %d = %d\n",numero,i,mult);
            }
            printf("\n\n\n");
        }
    }

    return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practica 1\Practica1_2_Tabla Multiplicar.exe
Elija una opcion:
1) Tabla de multiplicar.
2) Salir.
1
Ingrese el numero del que quiere obtener la tabla de multiplicar:
7
Hasta que numero desea la tabla de multiplicar?:
15
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84
7 x 13 = 91
7 x 14 = 98
7 x 15 = 105

Elija una opcion:
1) Tabla de multiplicar.
2) Salir.
```

numero = 7

maximo = 15

mult = 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105

PROGRAMA 3. MAYOR Y MENOR

Descripción

Crear un programa que obtenga el mayor y el menor de una secuencia de números.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

int numero; *variable de la cantidad de elementos de la secuencia.*

float secuencia[numero]; *arreglo de longitud establecida para almacenar la secuencia.*

Salidas:

float mayor; *número mayor de la secuencia.*

float menor; *número menor de la secuencia.*

2. ¿Qué es lo que hará el programa?

Con dos ciclos for anidados, para el número mayor se comparará el dato en esta variable con todas las posiciones del arreglo, si un número en el arreglo es mayor que la variable mayor, adquirirá su valor.

Con el número menor será un proceso similar, a diferencia que cada vez que encuentre un número más pequeño, adquirirá este valor.

3. ¿Qué espero de salida?

Imprimir -> float mayor, menor; *Se imprimirán ambos números en pantalla.*

Requerimientos Funcionales

El programa permite introducir una secuencia de números de longitud finita decidida por el usuario.

El programa acepta números enteros y flotantes, positivos y negativos.

El programa compara y obtiene qué número es mayor y menor.

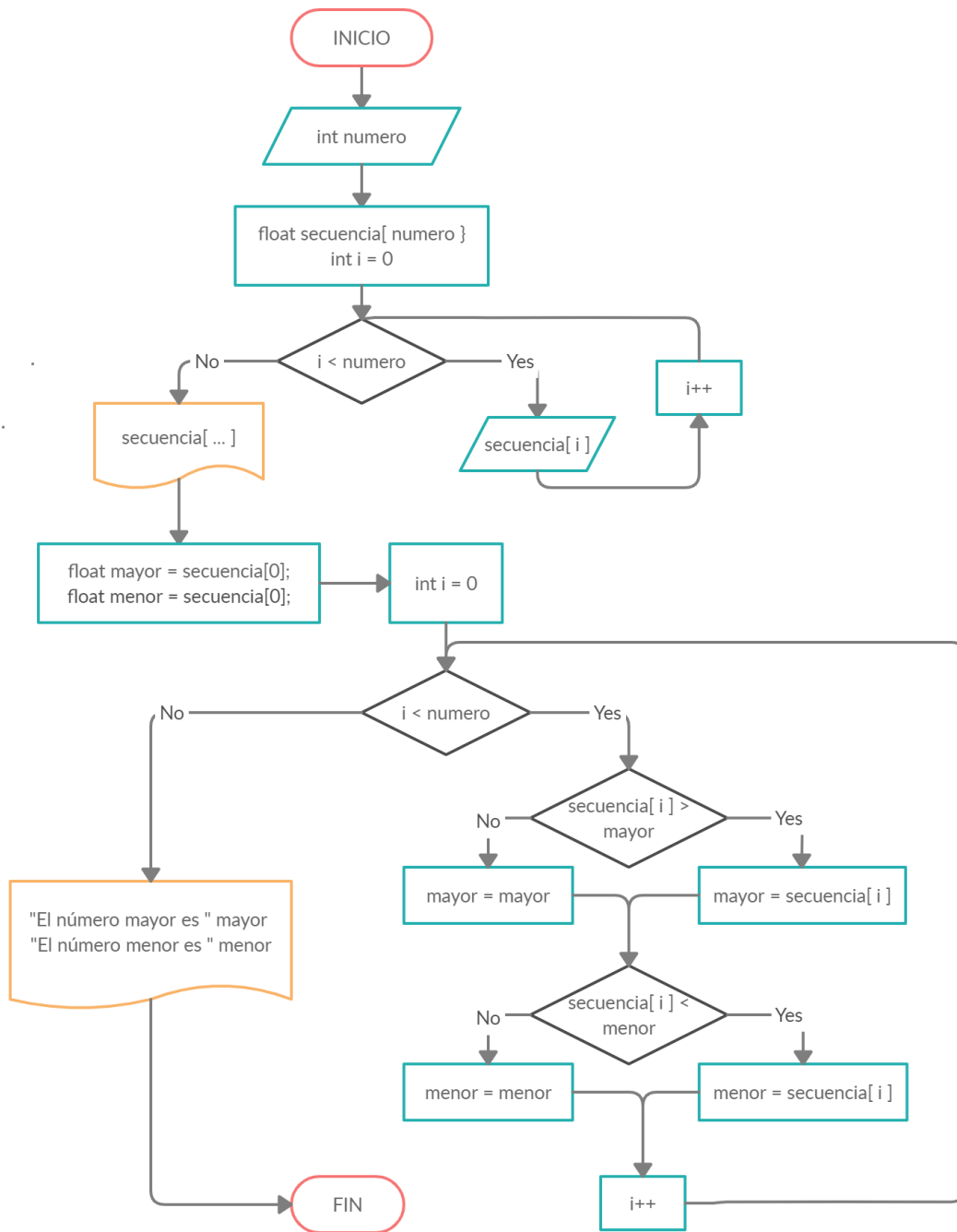
El programa se repite hasta que el usuario de en la opción de salir.

Requerimientos no Funcionales

El programa no admite una cantidad de números mayor o menor a la establecida con anticipación.

El programa no admite introducir una longitud menor o igual a cero.

Diagrama de flujo



Código en C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

    int numero = 0, indice = 0;
    float mayor = 0, menor = 0;

    while(indice != 2){
        printf("Elija una opcion:\n1) Obtener mayor y menor.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);

        if(indice == 1){
            printf("Ingrese cuantos numeros seran la secuencia:\n");
            scanf("%d",&numero);
            fflush(stdin);

            float secuencia[numero] = {};
            printf("Ingrese la secuencia:\n");
            fflush(stdin);

            for(int i=0; i<numero; i++){
                scanf("%f",&secuencia[i]);
            }
            for(int i=0; i<numero; i++){
                printf("[%.2f] ",secuencia[i]);
            }

            //NUMERO MAYOR - MENOR
            mayor = secuencia[0];
            menor = secuencia[0];
            for(int i=0; i<numero; i++){
                (secuencia[i] > mayor) ? mayor = secuencia[i] : mayor = mayor;
                (secuencia[i] < menor) ? menor = secuencia[i] : menor = menor;
            }
            printf("\n\nEl numero MAYOR es %.2f",mayor);
            printf("\nEl numero MENOR es %.2f",menor);
            printf("\n\n\n");
        }
    }
    return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practica 1\Prac...
1) Obtener mayor y menor.
2) Salir.
1
Ingrese cuantos numeros seran la secuencia:
15
Ingrese la secuencia:
2 4 5 -6 8
9 0 -1.2 3.6 10
8.1 2 -1.3 0 12
[2.00] [4.00] [5.00] [-6.00] [8.00] [9.00] [0.00] [-1.20] [3.60] [10.00]
[8.10] [2.00] [-1.30] [0.00] [12.00]

El numero MAYOR es 12.00
El numero MENOR es -6.00

Elija una opcion:
1) Obtener mayor y menor.
2) Salir.
```

numero = 15

secuencia[] = {2, 4, 5, -6, 8, 9, 0, -1.2, 3.6, 10, 8.1, 2, -1.3, 0, 12}

mayor = 12

menor = -6

PROGRAMA 4. TRES NÚMEROS COMPARADOS

Descripción

Diseñar un algoritmo que lea 3 números y encuentre si uno de ellos es la suma de los otros dos.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

float numA;

float numB;

float numC; **números a comparar.**

2. ¿Qué es lo que hará el programa?

Con uso de las ternas, se comparará si $\text{numA} + \text{numB}$ suman numC , si no, se comparará si numA y numC suman numB , en caso de que tampoco, comparará si numB y numC suman numA .

Si ninguno de los casos se cumple, significará que ninguna pareja de números suman al otro.

3. ¿Qué espero de salida?

Imprimir -> Imprimir qué números suman al otro; **Se imprimirán ambos números en pantalla.**

Requerimientos Funcionales

El programa permite introducir tres números enteros y flotantes positivos y negativos.

El programa verifica si la suma de dos números de los introducidos suman al elemento restante.

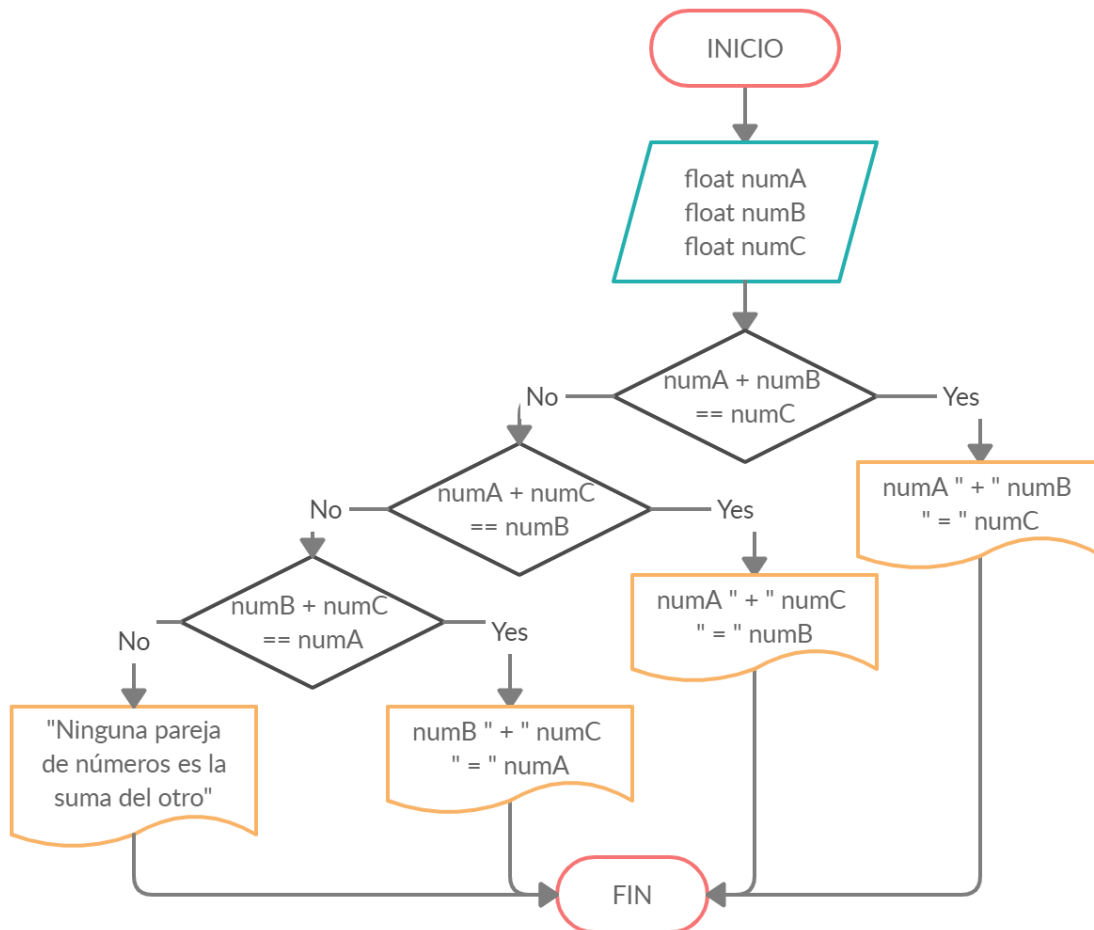
El programa se detiene hasta que el usuario de la opción de salir.

Requerimientos no Funcionales

El programa no admite más o menos de tres números.

El programa sólo imprime números con dos decimales.

Diagrama de flujo



Código en C

```
#include <stdlib.h>
#include <stdio.h>

int main(){
    float numA = 0, numB = 0, numC = 0;
    int indice = 0;

    while(indice != 2){
        printf("Elija una opcion:\n1) Ingresar 3 numeros y compararlos.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);

        if(indice == 1){
            printf("Ingresa 3 numeros enteros:");
            scanf("%f %f %f",&numA, &numB, &numC);
            fflush(stdin);

            ((numA + numB) == numC) ? printf("[%.2f + %.2f = %.2f]\n\n",numA,numB,numC):
            ((numA + numC) == numB) ? printf("[%.2f + %.2f = %.2f]\n\n",numA,numC,numB):
            ((numB + numC) == numA) ? printf("[%.2f + %.2f = %.2f]\n\n",numB,numC,numA):
            printf("Ninguna pareja de numeros es la suma del otro.\n\n\n");
        }
    }

    return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practi
Elija una opcion:
1) Ingresar 3 numeros y compararlos.
2) Salir.
1
Ingresa 3 numeros enteros:2.5 13 10.5
[2.50 + 10.50 = 13.00]

Elija una opcion:
1) Ingresar 3 numeros y compararlos.
2) Salir.
1
Ingresa 3 numeros enteros:5 0.2 48.23
Ninguna pareja de numeros es la suma del otro.
```

Caso 1:

numA = 2.5

numB = 13

numC = 10.5

Salida-> "[2.5 + 10.5 = 13]"

Caso 2:

numA = 5

numB = 0.2

numC = 48.23

Salida-> "Ninguna pareja de números es la suma del otro"

PROGRAMA 5 DIVISIÓN CON RESTAS

Descripción

Realizar un programa que divida dos números dados, resolverlo mediante el método de restas sucesivas.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

int dividendo;

int divisor;

Salidas:

int resultado;

int residuo;

2. ¿Qué es lo que hará el programa?

Con el método de las sumas, al dividendo se le restará el divisor. La división terminará cuando al residuo de la resta ya no se le puede restar el divisor sin crear valores negativos.

3. ¿Qué espero de salida?

Imprimir -> resultado, residuo

Requerimientos Funcionales

El programa acepta números enteros sólo positivos.

El programa realiza la división por el método de las sumas decisivas y el resultado es un entero, con el residuo como otro entero.

Requerimientos no Funcionales

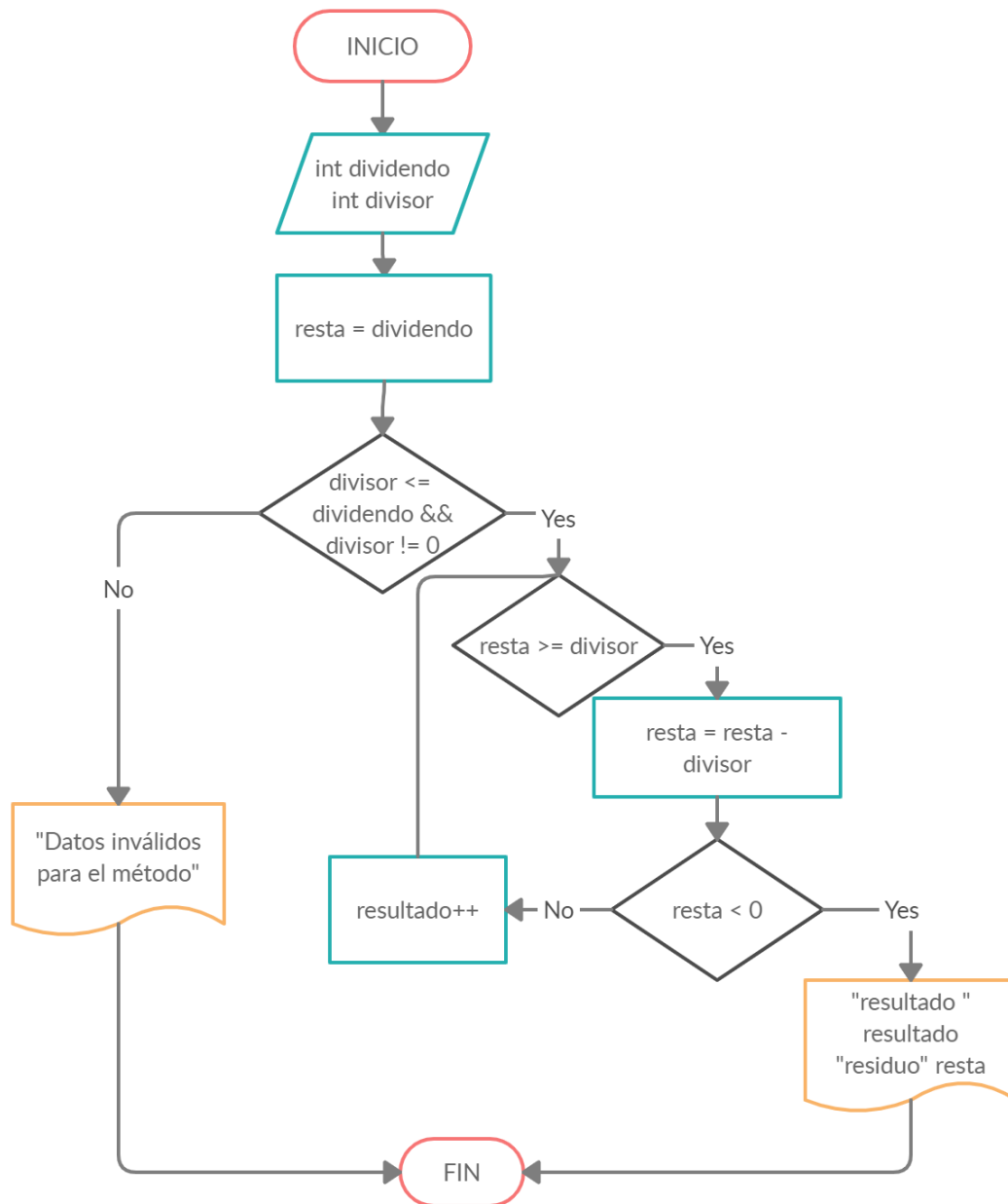
El programa recibe únicamente dos valores, el divisor y el dividendo.

El programa no acepta números negativos, ni decimales o flotantes.

El programa no realiza la división si el divisor es mayor que el dividendo.

El programa no devuelve valores decimales ni flotantes.

Diagrama de flujo



Código en C

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int indice = 0;

    while(indice != 2){
        printf("Elija una opcion:\n1) Division con restas.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);

        if(indice == 1){
            int dividendo = 0, divisor = 0, resta = 0, resultado = 0;
            printf("Introduce 2 numeros (dividendo) (divisor):");
            scanf("%d %d",&dividendo,&divisor);
            fflush(stdin);
            resta = dividendo;

            if(divisor <= dividendo && divisor != 0){
                while(resta >= divisor){
                    resta = resta - divisor;
                    resultado++;
                }
                printf("Resultado: %d con residuo: %d\n\n", resultado, resta);
            }
            else
                printf("Datos invalidos para el metodo de restas sucesivas\n\n\n");
        }
    }
    return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practica 1\Practica1_5_Division restas sucesivas.exe
Elija una opcion:
1) Division con restas.
2) Salir.
1
Introduce 2 numeros (dividendo) (divisor):7 4
Resultado: 1 con residuo: 3

Elija una opcion:
1) Division con restas.
2) Salir.
1
Introduce 2 numeros (dividendo) (divisor):2 6
Datos invalidos para el metodo de restas sucesivas

Elija una opcion:
1) Division con restas.
2) Salir.
```

Caso 1:

dividendo = 7

divisor = 4

resultado = 1

residuo = 3

Caso 2:

dividendo = 2

divisor = 6

No se puede realizar la división

CONCLUSIONES

El uso del scanf en esta práctica fue fácil, sólo que, si la introducción del tipo de dato era incorrecta, los programas fallaban o entraban en un ciclo infinito. La función fflush(stdin) solucionó esos problemas ya que después de un scanf limpia el buffer de entrada.