

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 4

Matrices y arreglos

Ciudad de México; a 29 de octubre de 2020.

Índice

INTRODUCCIÓN	4
DESARROLLO	5
PROGRAMA 1. Intercambio de matrices	5
Descripción.....	5
Análisis.....	5
Requerimientos Funcionales	6
Requerimientos no Funcionales	6
Diagrama de flujo	7
Código en C	8
Resultados (compilación)	10
PROGRAMA 2. Palabras al revés	2
Descripción.....	2
Análisis.....	2
Requerimientos Funcionales	3
Requerimientos no Funcionales	3
Diagrama de flujo	4
Código en C	5
Resultados (compilación)	6
PROGRAMA 3. Triángulo de Pascal	7
Descripción.....	7
Análisis.....	7
Requerimientos Funcionales	8

Requerimientos no Funcionales	8
Diagrama de flujo	9
Código en C	10
Resultados (compilación)	11
CONCLUSIONES	2

INTRODUCCIÓN

Las cadenas o strings en C, son arreglos unidimensionales de tipo char que incluso, cuentan con su propia librería `#include <string.h>` con funciones que facilitan sus operaciones y optimizan el código del programa.

En esta práctica se analiza el comportamiento de las cadenas que pueden operarse tanto como un arreglo que como un string.

Las matrices son arreglos bidimensionales que como ya se había visto en prácticas pasadas, empaquetan grupos de datos acomodados. El intercambio de valores de una matriz a otra de un modo básico y largo, sin uso de apuntadores, se da por medio de variables auxiliares, que sirven para almacenar valores antes de ser modificados.

DESARROLLO

PROGRAMA 1. INTERCAMBIO DE MATRICES

Descripción

Crear un programa que cree dos arreglos. Que se cambie los valores de la matriz A a la Matriz B y los valores de la Matriz B se guarden en la matriz A.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

int filas; cantidad de filas

int columnas; cantidad de columnas

float matrizA[filas][columnas];

float matrizB[filas][columnas];

Salidas:

matrizA y matrizB con datos intercambiados.

2. ¿Qué es lo que hará el programa?

Se solicitará al usuario que introduzca el número de filas y columnas de ambas matrices. Desde teclado, las dos matrices se llenarán para el usuario. Cuando el usuario de en la opción de intercambiar matrices, en una variable auxiliar se guardarán los valores de la matriz A, la matriz A adquirirá los valores de matriz B y matriz B obtendrá los valores del auxiliar. Los datos se irán intercambiando posición por posición mediante dos ciclos for anidados.

3. ¿Qué espero de salida?

Imprimir -> matrizA y matrizB con datos intercambiados.

Requerimientos Funcionales

El programa intercambia todos los valores de una matriz a otra (ambas matrices de las mismas dimensiones), la cantidad de veces que el usuario quiera.

El programa acepta valores enteros y decimales en los datos de llenado de las matrices.

El programa termina cuando el usuario de la opción de salir.

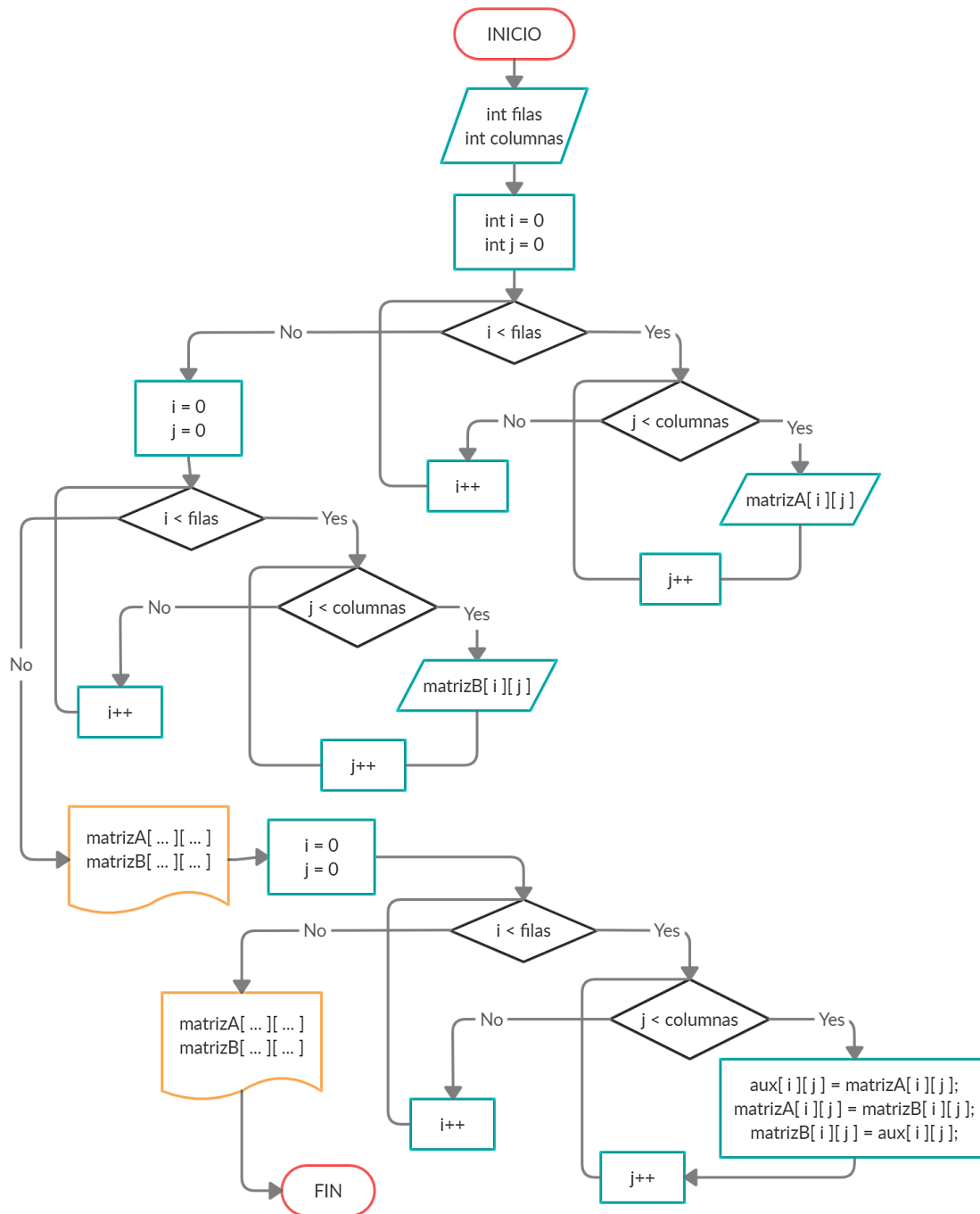
Requerimientos no Funcionales

El programa no intercambia valores entre más de dos matrices a la vez.

Las matrices no reciben valores que no sean numéricos.

El programa no imprime valores flotantes con más dos decimales.

Diagrama de flujo



Código en C

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

int main(){

    int indice = 0, filas, columnas;

    while(indice != 3){
        system("cls");
        indice = 0; filas = 0; columnas = 0;

        printf("Introduzca el numero de filas y columnas de ambas matrices:\nfilas:");
        scanf("%d",&filas);
        fflush(stdin);
        printf("columnas:");
        scanf("%d",&columnas);
        fflush(stdin);

        float matrizA[filas][columnas] = {};
        float matrizB[filas][columnas] = {};
        float aux[filas][columnas] = {};

        //LLENADO DE MATRICES
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("Matriz A: dato en [%d][%d] = ",i,j);
                scanf("%f",&matrizA[i][j]); fflush(stdin);
            }
            printf("\n");
        }
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("Matriz B: dato en [%d][%d] = ",i,j);
                scanf("%f",&matrizB[i][j]); fflush(stdin);
            }
        }

        //IMPRESION DE LAS MATRICES
        printf("\nMatriz A\n");
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("[%2f]\t",matrizA[i][j]);
            }
            printf("\n");
        }

        printf("\nMatriz B\n");
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("[%2f]\t",matrizB[i][j]);
            }
            printf("\n");
        }
    }
}
```



```

while(indice != 2 && indice != 3){ //MATRICES OPERANDO

    printf("\n\nElija una opcion:\n1) Intercambiar matrices.\n2) Cambiar datos de matrices.\n");
    printf("3) Salir\n");
    scanf("%d",&indice);
    fflush(stdin);

    if(indice == 1 || indice == 2){

        //INTERCAMBIO DE VALORES
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                aux[i][j] = matrizA[i][j];
                matrizA[i][j] = matrizB[i][j];
                matrizB[i][j] = aux[i][j];
            }
        }

        printf("\nIntercambiando valores de las matrices...\n\nMatriz A\n");
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("%.2f\t",matrizA[i][j]);
            }
            printf("\n");
        }

        printf("\nMatriz B\n");
        for(int i=0; i<filas; i++){
            for(int j=0; j<columnas; j++){
                printf("%.2f\t",matrizB[i][j]);
            }
            printf("\n");
        }
    }
}

return 0;
}

```

Resultados (compilación)

```
Introduzca el numero de filas y columnas de ambas matrices:
filas:2
columnas:3
Matriz A: dato en [0][0] = 1
Matriz A: dato en [0][1] = 2.6
Matriz A: dato en [0][2] = 7
Matriz A: dato en [1][0] = 9.8
Matriz A: dato en [1][1] = 0.1
Matriz A: dato en [1][2] = 0.02

Matriz B: dato en [0][0] = 15
Matriz B: dato en [0][1] = 2.4
Matriz B: dato en [0][2] = 0
Matriz B: dato en [1][0] = 0
Matriz B: dato en [1][1] = 1.7
Matriz B: dato en [1][2] = 65

Matriz A
[1.00] [2.60] [7.00]
[9.80] [0.10] [0.02]

Matriz B
[15.00] [2.40] [0.00]
[0.00] [1.70] [65.00]

Elija una opcion:
1) Intercambiar matrices.
2) Cambiar datos de matrices.
3) Salir
1

Intercambiando valores de las matrices...

Matriz A
[15.00] [2.40] [0.00]
[0.00] [1.70] [65.00]

Matriz B
[1.00] [2.60] [7.00]
[9.80] [0.10] [0.02]
```

MatrizA = { 1, 2.6, 7, 9.8, 0.1, 0.02 } MatrizB = {15, 2.4, 0, 0, 1.7, 65 }

Intercambio:

MatrizA = {15, 2.4, 0, 0, 1.7, 65 } MatrizB = { 1, 2.6, 7, 9.8, 0.1, 0.02 }

PROGRAMA 2. PALABRAS AL REVÉS

Descripción

Crear un programa que cree un arreglo de caracteres y que ingrese la palabra y la muestre al revés HOLA ALOH.

Análisis

1. ¿Cuáles son la entradas y salidas?

Entradas:

char palabra[250]; [palabra introducida desde teclado.](#)

Salidas:

char invertida[250]; [palabra invertida.](#)

2. ¿Qué es lo que hará el programa?

Se solicitará al usuario que introduzca la palabra desde teclado. Dentro de un for que inicie en longitud de la palabra menos uno, y vaya restando un uno en cada ciclo hasta llegar a cero, una variable auxiliar con valor cero irá sumando una unidad por cada ciclo hasta llegar a longitud menos uno. Los valores contados desde inversa de la palabra se irán almacenando en el nuevo arreglo que iniciará desde la posición cero, para guardar los caracteres en orden invertido.

3. ¿Qué espero de salida?

Imprimir -> invertida[250]

Requerimientos Funcionales

El programa guarda una palabra introducida por el usuario y la imprime escrita al revés.

El programa recibe una cadena hasta de 250 caracteres.

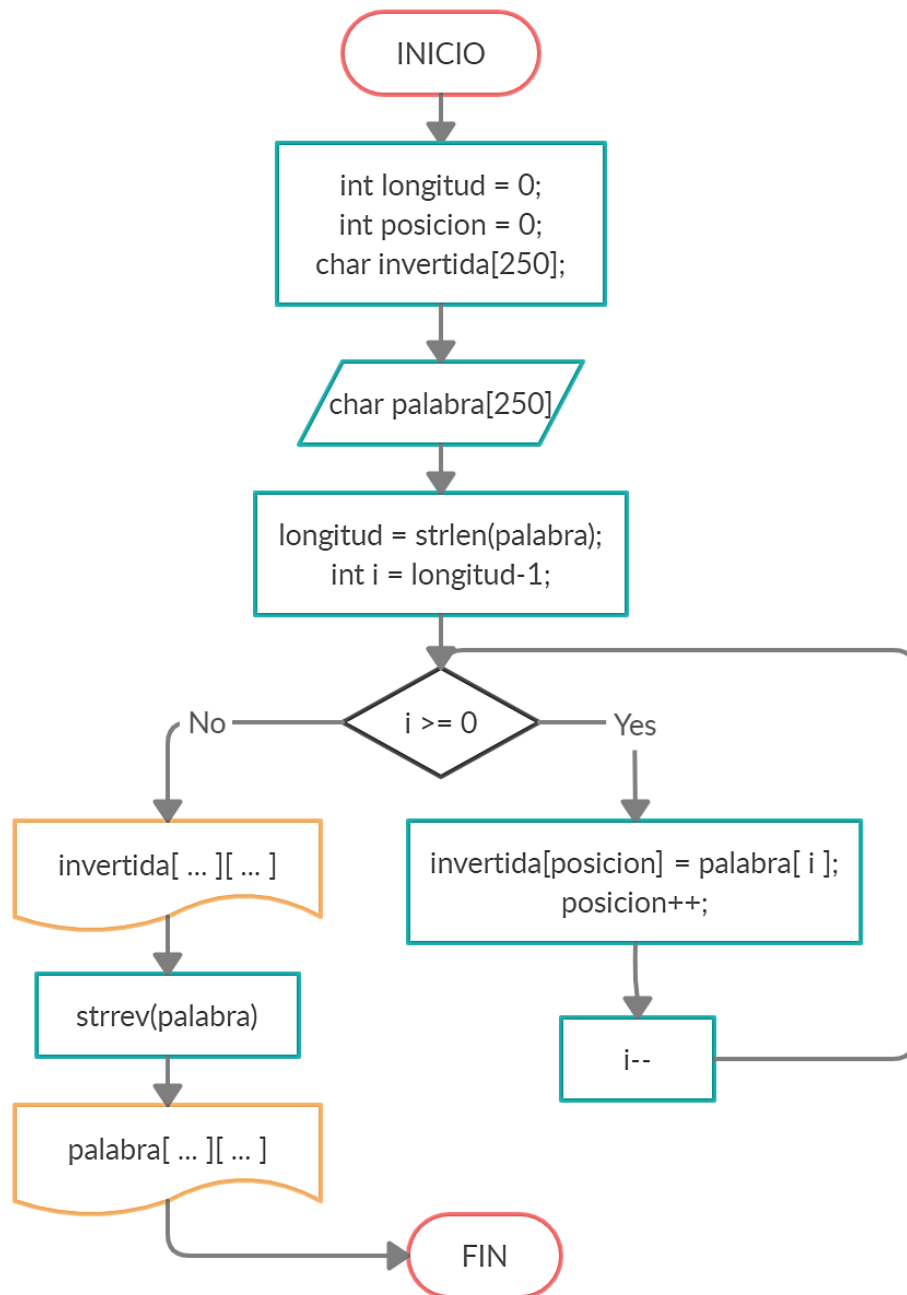
El programa termina hasta que el usuario de la opción de salir.

Requerimientos no Funcionales

El programa no admite más de una palabra a la vez, si se escribiera más de una, el programa únicamente tomaría la primera palabra y la invertiría.

El programa no recibe una palabra que supere los 250 caracteres.

Diagrama de flujo



Código en C

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

int main(){
    char palabra[250];
    char invertida[250];
    int longitud, posicion, indice = 0;

    while(indice != 2){
        indice = 0; longitud = 0; posicion = 0;
        char invertida[250] = {};

        printf("\nElija una opcion:\n1) Invertir una palabra.\n2) Salir.\n");
        scanf("%d",&indice);
        fflush(stdin);

        if(indice == 1){
            printf("Escribe tu palabra:\n");
            scanf("%s",&palabra);
            fflush(stdin);
            longitud = strlen(palabra);

            for(int i=longitud-1; i>=0; i--){
                invertida[posicion] = palabra[i];
                posicion++;
            }

            printf("\n\n%s\n",invertida);

            strrev(palabra);
            printf("\n%s\n\n",palabra);
        }
    }

    return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3n Electiva\Practica 4\29.10.2020
Elija una opcion:
1) Invertir una palabra.
2) Salir.
1
Escribe tu palabra:
PARANGARICUTIRIMICUARO

ORAUCIMIRITUCIRAGNARAP

ORAUCIMIRITUCIRAGNARAP

Elija una opcion:
1) Invertir una palabra.
2) Salir.
1
Escribe tu palabra:
CALAMARINO

ONIRAMALAC

ONIRAMALAC

Elija una opcion:
1) Invertir una palabra.
2) Salir.
```

Caso 1:

palabra = PARANGARICUTIRIMICUARO

invertida = ORAUCIMIRITUCIRAGNARAP

Caso 2:

palabra = CALAMARINO

invertida = ONIRAMALAC

PROGRAMA 3. TRIÁNGULO DE PASCAL

Descripción

Realizar el triángulo de pascal.

Análisis

4. ¿Cuáles son la entradas y salidas?

Entradas:

int longitud; número de filas del triángulo de Pascal.

Salidas:

int triangulo[...][...]

5. ¿Qué es lo que hará el programa?

Se solicitará al usuario que introduzca la cantidad de filas que quiera visualizar del triángulo de Pascal. Para representar el triángulo, se utilizará una matriz bidimensional donde primeramente se formará el contorno del triángulo hecho de unos. Para el llenado de los datos internos, para cada posición su valor se obtendrá de la suma de los dos números que se ubican sobre la fila anterior.

6. ¿Qué espero de salida?

Imprimir -> int triangulo[...][...]

Requerimientos Funcionales

El programa permite ver el número de filas que el usuario quiera visualizar del triángulo de pascal (hasta el número 34).

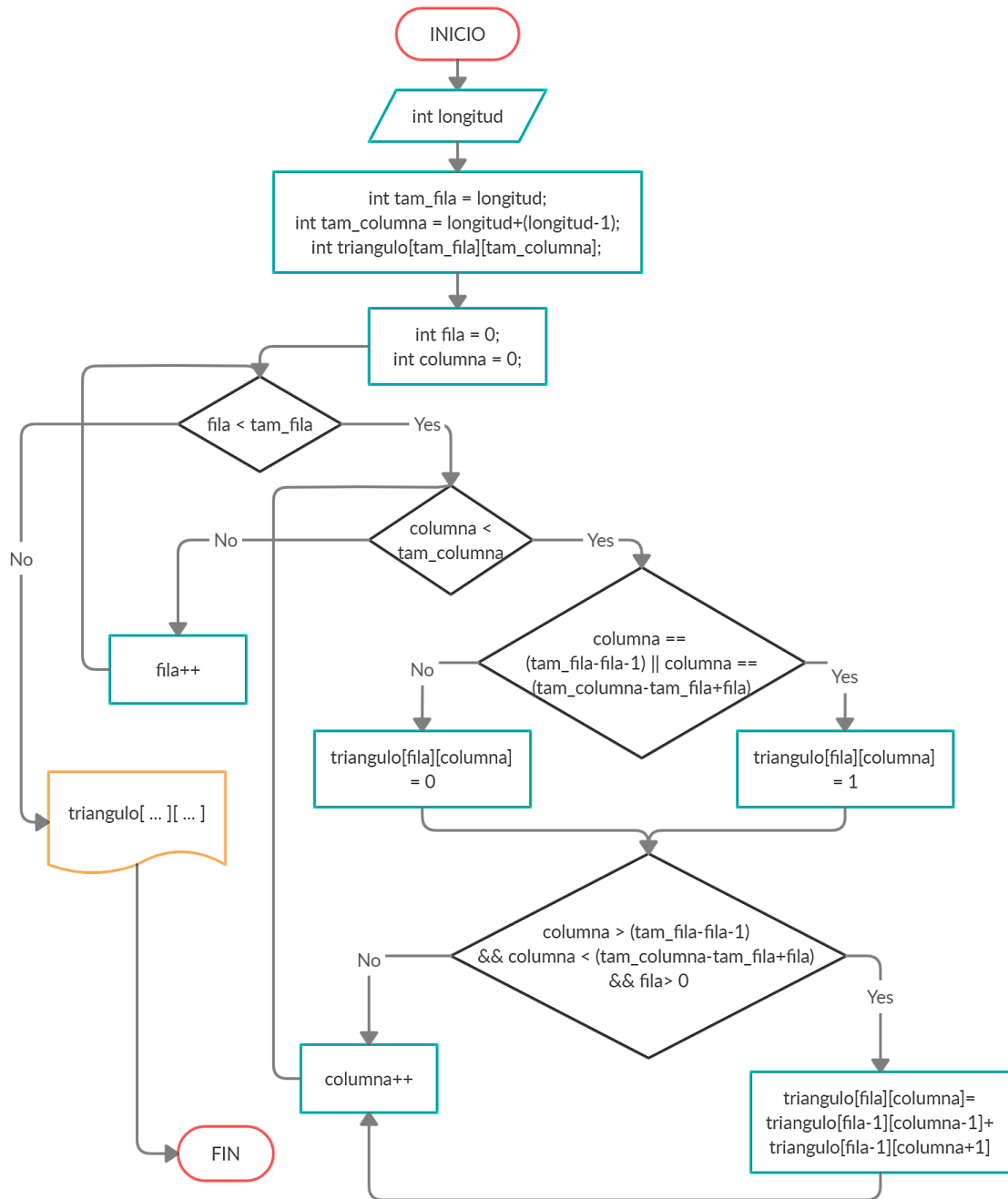
El programa termina hasta que el usuario de la opción de salir.

Requerimientos no Funcionales

El programa no recibe números flotantes ni caracteres como cantidad de filas en el triángulo.

El programa no visualiza más de 34 filas porque el tipo de dato no soporta más bytes.

Diagrama de flujo



Código en C

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

int main(){

int indice, longitud, tam_filas = 0, tam_columna = 0, posicion = 0;

while(indice != 2){
    indice = 0;
    printf("\nElija una opcion:\n1) Crear Triangulo.\n2) Salir.\n");
    scanf("%d",&indice);
    fflush(stdin);

    if(indice == 1){
        longitud = 0;
        printf("Introduzca el tamaño de la base: ");
        scanf("%d",&longitud);
        fflush(stdin);

        tam_filas = longitud;
        tam_columna = longitud+(longitud-1);
        int triangulo[tam_filas][tam_columna] = {};

        for(int fila=0; fila<tam_filas; fila++){
            for(int columna=0; columna<tam_columna; columna++){

                if(columna == (tam_filas-fila-1) ||
                columna == (tam_columna - tam_filas + fila))
                    triangulo[fila][columna] = 1;
                else
                    triangulo[fila][columna] = 0;

                if(columna > (tam_filas-fila-1) && columna <
                (tam_columna - tam_filas + fila) && fila>0)
                    triangulo[fila][columna] =
                    triangulo[fila-1][columna-1]+triangulo[fila-1][columna+1];
            }
        }

        for(int fila=0; fila<tam_filas; fila++){
            printf("\n");
            for(int columna=0; columna<tam_columna; columna++){
                (triangulo[fila][columna] == 0) ? printf(" ") :
                printf("%d",triangulo[fila][columna]);
            }
            printf("\n\n");
        }
    }
return 0;
}
```

Resultados (compilación)

```
C:\Users\kardo\Desktop\SEMESTRE 21-1\Programaci3/n Electiva\Practica 4\Pra
2) Salir.
1
Introduzca numero de filas que quiere visualizar: 5

    1
  1 1
1 2 1
1 3 3 1
1 4 6 4 1

Elija una opcion:
1) Crear Triangulo de Pascal.
2) Salir.
1
Introduzca numero de filas que quiere visualizar: 7

    1
  1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

Elija una opcion:
1) Crear Triangulo de Pascal.
2) Salir.
```

CONCLUSIONES

Para el primer ejercicio, el cambio de valores de una matriz a otra, se utilizó una variable auxiliar de tipo arreglo bidimensional, que con ayuda de dos for anidados extrajo los datos de la primera matriz. Dentro de los mismos ciclos, se realizó el traspaso de una matriz y otra y por último, la segunda matriz obtuvo sus valores de la variable auxiliar.

Para el segundo ejercicio se comparó la funcionalidad de un método preestablecido de la biblioteca `#include <string.h>` para revertir una cadena, con un método utilizando ciclos for guardando la cadena en otra variable pero de manera inversa, obteniendo los mismos resultados.

Para el último ejercicio, se reutilizó el código del triángulo equilátero con la variante de que las demás posiciones del triángulo se rellenaron sumando los dos números colocados encima a las posiciones.