	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor : Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 11

Integrante(s): Meraz Dionicio Israel

No. de Equipo de cómputo empleado: 3

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 28 de Octubre de 2019

Observaciones:

CALIFICACIÓN: _____

OBJETIVO:
Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

INTRODUCCION:
Los arreglos (arrays) permiten almacenar vectores y matrices. Los arreglos unidimensionales sirven para manejar vectores y los arreglos bidimensionales para matrices. Sin embargo, las matrices también se pueden almacenar mediante arreglos unidimensionales y por medio de apuntadores a apuntadores, temas que se verán en el capıtulo siguiente.
La palabra unidimensional no indica que se trata de vectores en espacios de dimension uno; indica que su manejo se hace mediante un subíndice. El manejo de los arreglos bidimensionales se hace mediante dos subíndices.

El siguiente ejemplo muestra la definición de tres arreglos,

- double x[80];
- int factores[30];
- char cod [20];

La declaracion de los arreglos bidimensionales, caso particular de los arreglos multidimensionales, se hace como en el siguiente ejemplo:

- double a[3][4];
- int pos[10][40];
- char list[25][25];

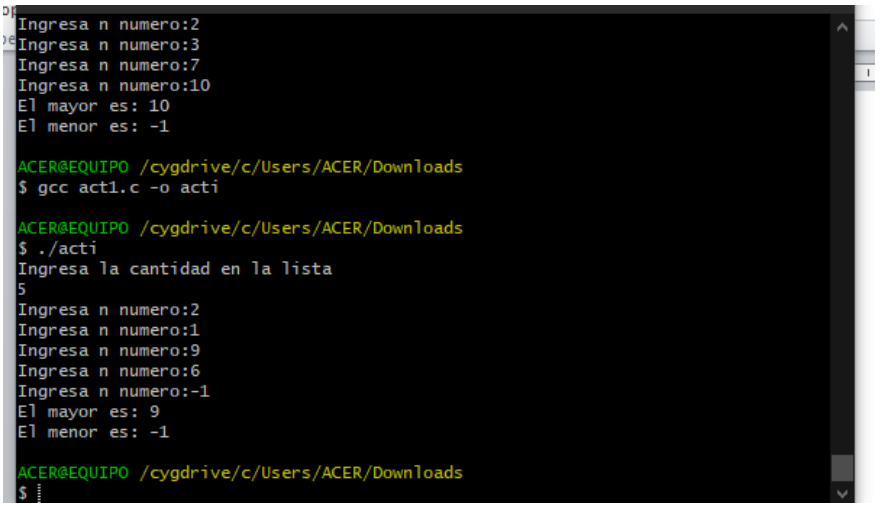
- Hacer un programa que:

Pida al usuario un número, Genere un arreglo de esa longitud. Pida al usuario números suficientes para llenarlo y Muestre al usuario el número menor y mayor de dicho arreglo.

```c
#include <stdio.h>
int main(){
    int n,i;
    int lista[n];
printf("Ingresa la cantidad en la lista\n");
scanf("%i", &n);

    for (int i=0; i<n; i++){
        printf("Ingresa n numero:");
        scanf("%i", &lista[i]);
        }
        int valor1=lista[0];
        int valor2=lista[0];

    for(int i=0; i<n; i++){
        if (lista[i]>valor1){
            valor1=lista[i];
        }
        if (lista[i]<valor2){
            valor2=lista[i];
        }
    }
    printf("El mayor es: %i\n", valor1);
    printf("El menor es: %i\n", valor2);
return 0;
}
```

Lo primero que hice es declarar los valores que voy a utilizar.

Depues de que haya pedido el número de espacios que va a ocupar la lista, con un for hice que llenaran primero la lista.
Enseguida los declarar valores pero inicializando en lista[0] que son el primer valor.
Con otro for hice que evaluara el primero valor con el primer valor y solo la variable va a cambiar si un valor de la lista es mayor o en el otro caso menor.



- Hacer un programa que:

Pida al usuario dos números N y M, genere dos matrices de N x M , Pida al usuario numeros suficientes para llenar amas matrices y Muestre al usuario la matriz resultado de sumar las dos de entrada.

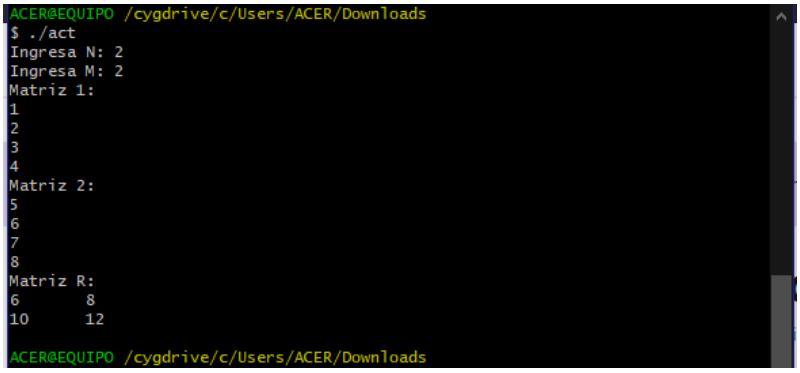
```c
#include <stdio.h>
int main(){
    int N, M;
    printf("Ingresa N: "); scanf("%i", &N);
    printf("Ingresa M: "); scanf("%i", &M);

int matriz[N][M];
printf("Matriz 1:\n");
for(int i=0; i<N; i++){
    for(int j=0; j<M; j++){
        scanf("%i", &matriz[i][j]);
    }
}
int matriz2[N][M];
printf("Matriz 2:\n");
for(int i=0; i<N; i++){
    for(int j=0; j<M; j++){
        scanf("%i", &matriz2[i][j]);
    }
}
int matrizR[N][M];
printf("Matriz R:\n");
for(int i=0; i<N; i++){
    for(int j=0; j<M;j++){
        matrizR[i][j]=matriz[i][j]+matriz2[i][j];
    }
}
for(int i=0; i<N; i++){
    for(int j=0; j<M; j++){

        printf("%i\t", matrizR[i][j]);
        }
        printf("\n");
    }
return 0;
}
```

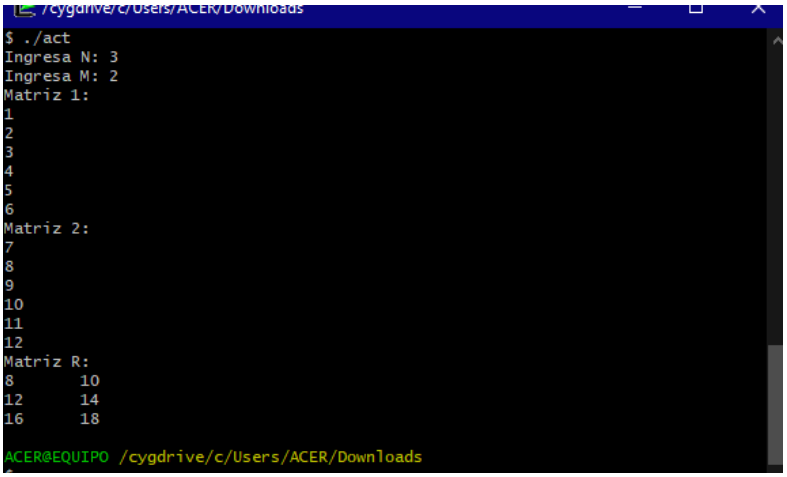
Primero se declaran las variables que corresponden a la cantidad de filas y columas en la matriz.

Enseguida hay que hacer que la matriz 1 y la matriz 2 se llenen de valores y para eso se utiliza dos for para que de alguna manera se multipliquen los dos valores que ingrese anteriormente.

Para hacer la suma se necesita que los valores se sumen son su respectivo casilla y esos valores asignarles en el mismo orden de casilla

Para que imprima se utiliza dos for para que muestre el valor





CONCLUSION:

Hay que tomarle la importancia que tiene ya que nos ayuda en agregar valores del tamaño que sea sin la necesidad de que te lo este pidiendo una por una y con la seguridad de que la vas guardando y con ello realizar cualquier operación como promedio o saber el menor y el mayor de esa lista.

Igual funciona para las matrices son de mucha utilidad para programación de videojuegos o incluso para resolver problemas relacionadas a las matrices.