



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

ALEJANDRO ESTEBAN PIMENTEL ALARCON

Profesor:

FUNDAMENTOS DE PROGRAMACION

Asignatura:

03

Grupo:

11

No de Práctica(s):

HINOJOSA RUIZ DIANA LAURA

Integrante(s):

*No. de Equipo de
cómputo empleado:*

22

6740

No. de Lista o Brigada:

PRIMER SEMESTRE

Semestre:

28 OCTUBRE 2019

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES

Objetivo

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

Introducción

Para la presente práctica solo realizamos dos actividades que se van a mostrar más adelante, pero ¿a que se refieren con “arreglo”? Un arreglo es un conjunto o una estructura de datos homogéneos que se encuentra ubicados en forma consecutiva, y normalmente se le conoce como una lista.

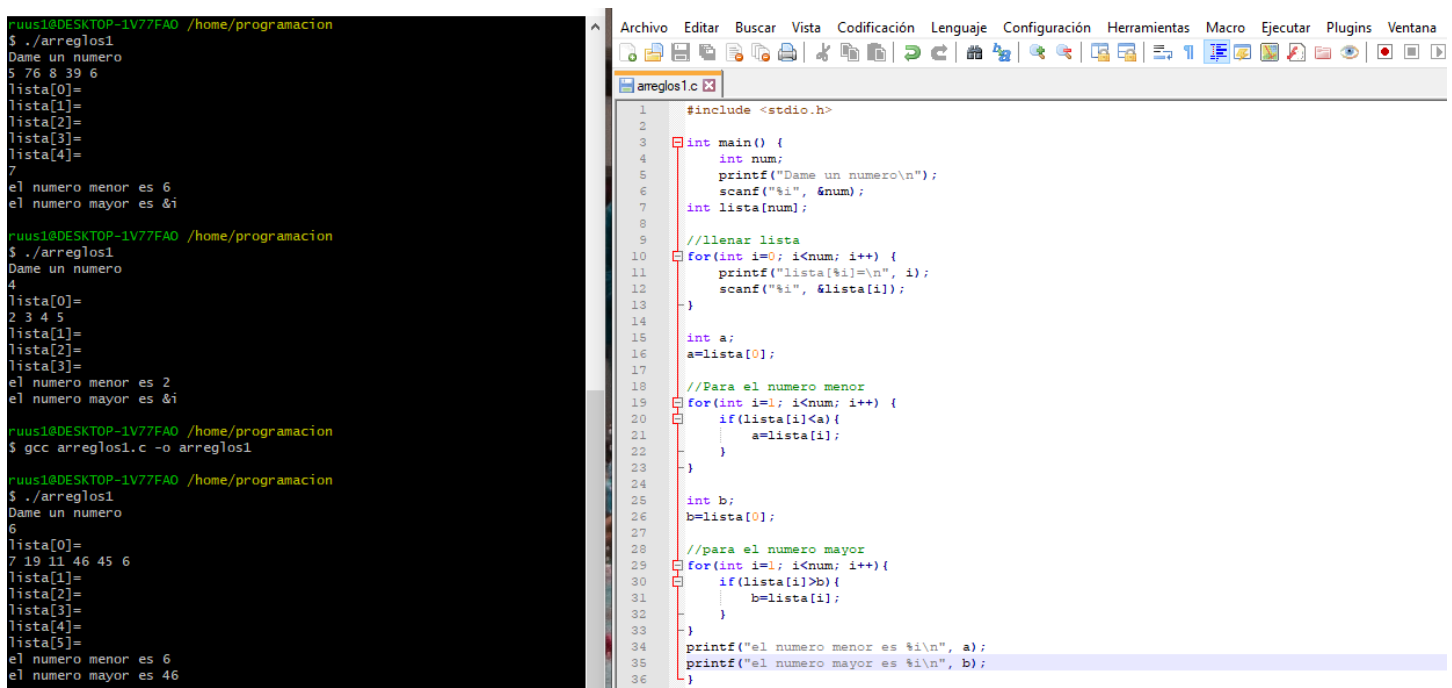
Desarrollo

Durante la practica el profesor no explico dos tipos de estructuras para un arreglo, la primera fue el unidimensional los cuales tienen una estructura normal y presentan listas de elementos iguales, la segunda fue el multidimensional, la cual se refiere a mas cadenas de datos. Y todos los datos que se guardan en los arreglos deben ser del mismo tipo.

Actividad 1:

Este programa pide un número y con este va a generar un arreglo de esa misma longitud, después de eso va a pedir al usuario números y va a parar hasta que llene el arreglo, finalmente va a tener que mostrar el mayor y el menor de todos.

Para esta actividad tuve que utilizar varios For, porque con eso pude trabajar mejor con las listas utilizadas.



```
ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./arreglos1
Dame un numero
5 76 8 39 6
lista[0]=
lista[1]=
lista[2]=
lista[3]=
lista[4]=
7
el numero menor es 6
el numero mayor es 8

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./arreglos1
Dame un numero
4
lista[0]=
2 3 4 5
lista[1]=
lista[2]=
lista[3]=
el numero menor es 2
el numero mayor es 5

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc arreglos1.c -o arreglos1

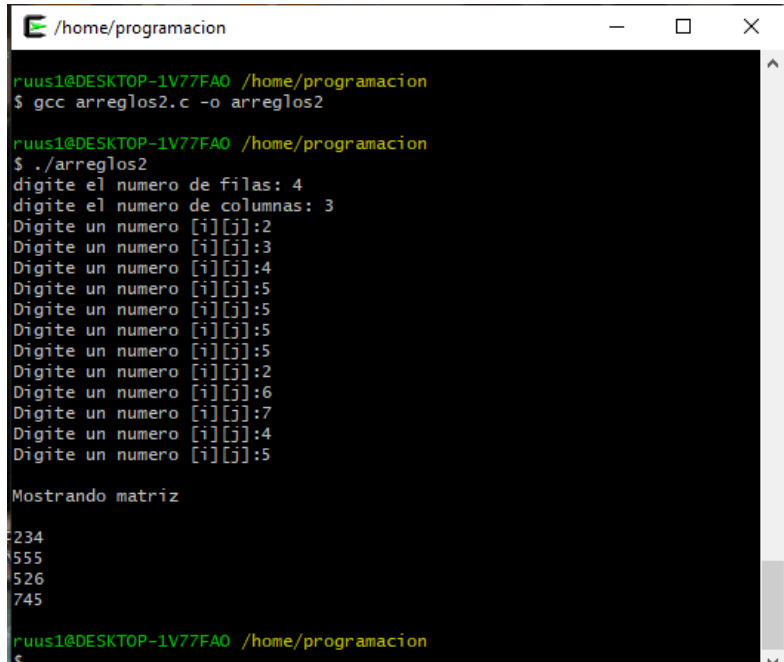
ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./arreglos1
Dame un numero
6
lista[0]=
7 19 11 46 45 6
lista[1]=
lista[2]=
lista[3]=
lista[4]=
lista[5]=
el numero menor es 6
el numero mayor es 46
```

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana
arreglos1.c
1  #include <stdio.h>
2
3  int main() {
4      int num;
5      printf("Dame un numero\n");
6      scanf("%i", &num);
7      int lista[num];
8
9      //llenar lista
10     for(int i=0; i<num; i++) {
11         printf("lista[%i]=\n", i);
12         scanf("%i", &lista[i]);
13     }
14
15     int a;
16     a=lista[0];
17
18     //Para el numero menor
19     for(int i=1; i<num; i++) {
20         if(lista[i]<a){
21             a=lista[i];
22         }
23     }
24
25     int b;
26     b=lista[0];
27
28     //para el numero mayor
29     for(int i=1; i<num; i++){
30         if(lista[i]>b){
31             b=lista[i];
32         }
33     }
34     printf("el numero menor es %i\n", a);
35     printf("el numero mayor es %i\n", b);
36 }
```

Actividad 2:

Este programa va a pedir dos números (los cuales uno corresponde a filas y otro a columnas) y va a generar dos matrices, las cuales ya hechas se van a sumar y se va a mostrar el resultado de la suma.

Para mí fue la actividad más difícil porque no encontraba la forma de poder sumar las matrices y de saber cuántos parámetros se podrían utilizar para cada una, utilicé lo de meter una for dentro de otro for y al parecer si funciona.



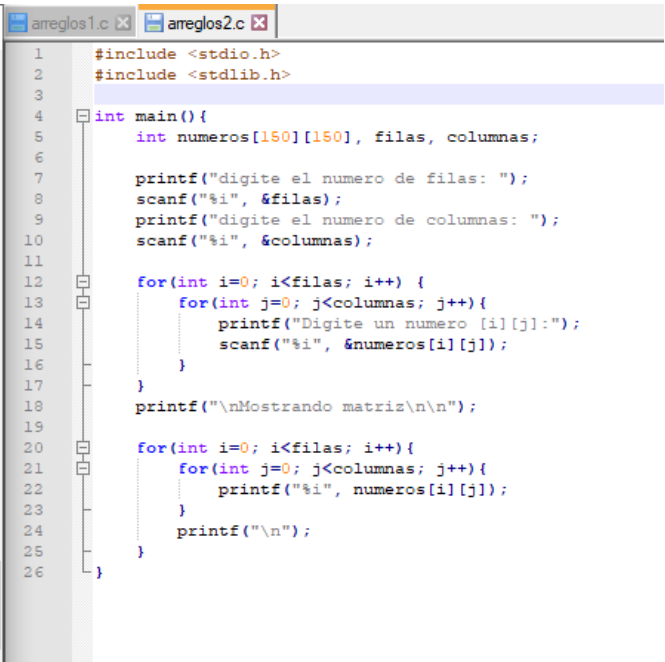
```
ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc arreglos2.c -o arreglos2

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./arreglos2
digite el numero de filas: 4
digite el numero de columnas: 3
Digite un numero [i][j]:2
Digite un numero [i][j]:3
Digite un numero [i][j]:4
Digite un numero [i][j]:5
Digite un numero [i][j]:5
Digite un numero [i][j]:5
Digite un numero [i][j]:2
Digite un numero [i][j]:6
Digite un numero [i][j]:7
Digite un numero [i][j]:4
Digite un numero [i][j]:5

Mostrando matriz

234
555
526
745

ruus1@DESKTOP-1V77FA0 /home/programacion
$
```



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int numeros[150][150], filas, columnas;
6
7      printf("digite el numero de filas: ");
8      scanf("%i", &filas);
9      printf("digite el numero de columnas: ");
10     scanf("%i", &columnas);
11
12     for(int i=0; i<filas; i++) {
13         for(int j=0; j<columnas; j++){
14             printf("Digite un numero [i][j]:");
15             scanf("%i", &numeros[i][j]);
16         }
17     }
18     printf("\nMostrando matriz\n\n");
19
20     for(int i=0; i<filas; i++){
21         for(int j=0; j<columnas; j++){
22             printf("%i", numeros[i][j]);
23         }
24         printf("\n");
25     }
26 }
```

Conclusión

Para mí fue interesante como siempre vamos a poder meter y meter más cosas a nuestros programas y de magnitudes que no nos imaginábamos, sobre todo cuando ya podemos resolver problemas tan básicos como una suma y unos más elaborados como una suma, pero de matrices.