



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Alejandro Esteban Pimentel Alarcon

Asignatura:

Fundamentos de programación

Grupo:

3

No de Práctica(s):

11

Integrante(s):

Nava Corona Nadia Erandeni

*No. de Equipo de cómputo
empleado:*

No. de Lista o Brigada:

6948

Semestre:

2020-1

Fecha de entrega:

Lunes 28 de Octubre

Observaciones:

Bastante bien, pero tuviste una omisión en la actividad 1

CALIFICACIÓN: 9

Arreglos unidimensionales y multidimensionales

Introducción.

En esta práctica aprenderemos a realizar arreglos unidimensionales (en este caso una lista) y arreglos multidimensionales (como lo es el caso de las matrices).

Pero ¿qué es un arreglo unidimensional y qué es un arreglo multidimensional?, un arreglo **unidimensional** es un tipo de datos estructurado que está formado por una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. Los datos que se guarden en los arreglos todos deben ser del mismo tipo y un arreglo **multidimensional** es aquel que consta de varias dimensiones de arreglos en la memoria, y utiliza el mismo método de declaración y creación que los arreglos unidimensionales y bidimensionales, y la forma de mandar a llamar a sus diferentes espacios de memoria es la misma

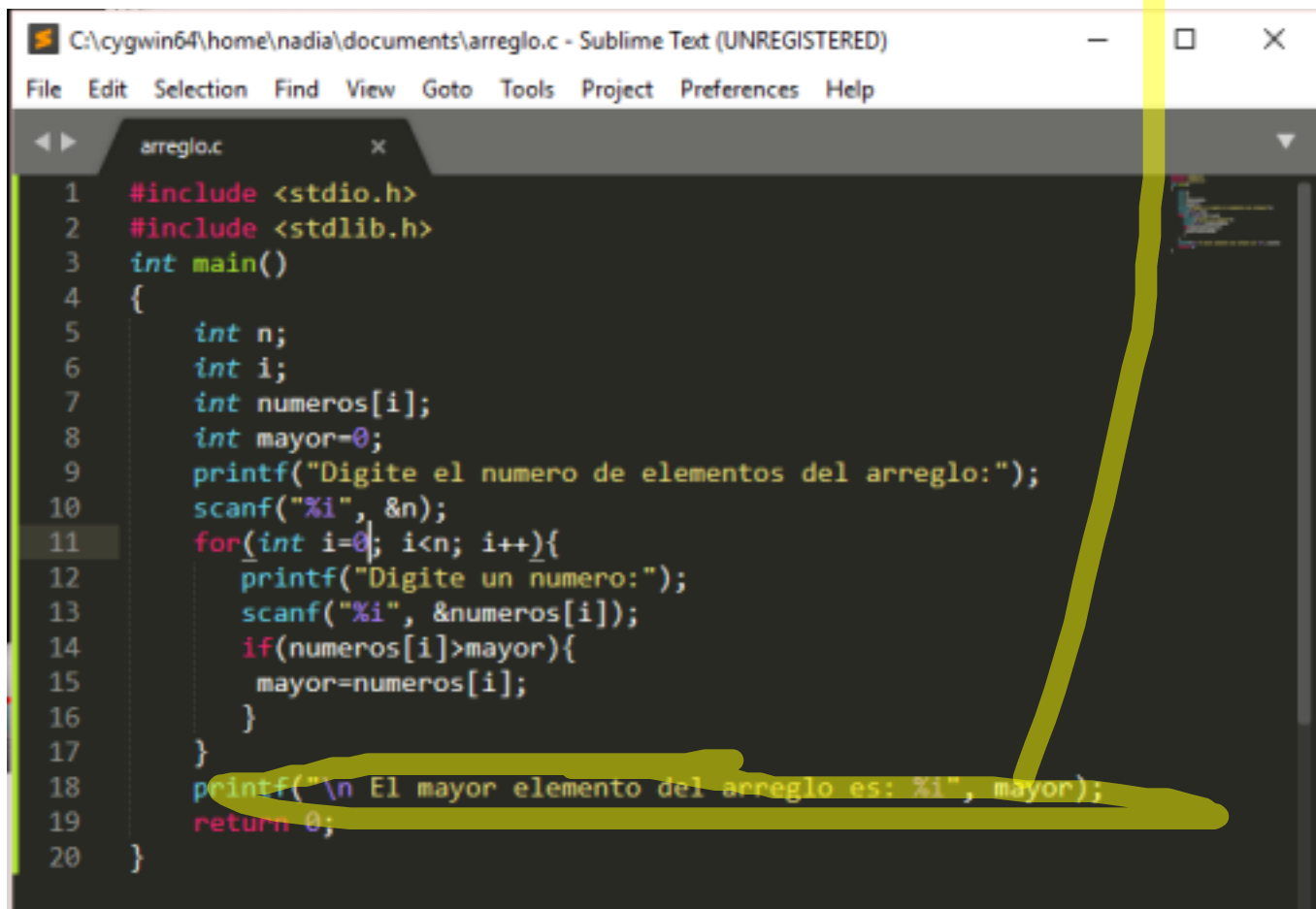
Objetivo.

Reconocer la importancia y 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.

Actividad 1. Hacer un programa que:

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

Falta el menor

A screenshot of a code editor window titled 'C:\cygwin64\home\nadia\documents\arreglo.c - Sublime Text (UNREGISTERED)'. The code is in C and defines a function 'main' that asks the user for the number of elements in an array, creates the array, prompts the user to enter values, and then finds the maximum value. The code is as follows:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int n;
6      int i;
7      int numeros[i];
8      int mayor=0;
9      printf("Digite el numero de elementos del arreglo:");
10     scanf("%i", &n);
11     for(int i=0; i<n; i++){
12         printf("Digite un numero:");
13         scanf("%i", &numeros[i]);
14         if(numeros[i]>mayor){
15             mayor=numeros[i];
16         }
17     }
18     printf("\n El mayor elemento del arreglo es: %i", mayor);
19     return 0;
20 }
```

A yellow highlight is drawn under the final printf statement on line 18. A yellow arrow points from the text 'Falta el menor' to the code, indicating that the program only finds the maximum but not the minimum.

```

nadia@LAPTOP-VR22P14N ~/documents
$ ./arreglo
Digite el numero de elementos del arreglo:3
Digite un numero:3
Digite un numero:7
Digite un numero:6

El mayor elemento del arreglo es: 7
nadia@LAPTOP-VR22P14N ~/documents
$ ./arreglo
Digite el numero de elementos del arreglo:4
Digite un numero:2
Digite un numero:5
Digite un numero:6
Digite un numero:9

El mayor elemento del arreglo es: 9
nadia@LAPTOP-VR22P14N ~/documents
$ |

```

Actividad 2. Hacer un programa que:

- Pida al usuario dos números N y M.
- Genere dos matrices de $N \times M$.
- Pida al usuario números suficientes para llenar ambas matrices.
- Muestre al usuario la matriz resultado de sumar las dos de entrada.

C:\cygwin64\home\nadia\documents\matriz.c - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```

matriz.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int suma;
6      int m,n; //m son las filas y n son las columnas
7      printf("Ingrese un numero de filas m y columnas n:\n");
8      scanf("%i", &m);
9      scanf("%i", &n);
10     int mat1 [m][n];
11     int mat2 [m][n];
12     int j;
13     int i;
14     printf("LLENADO DE MATRIZ 1\n");
15     for(int i=0; i<m; i++){
16         for (int j=0; j<n; j++){
17             printf("Ingrese numeros:");
18             scanf("%i", &mat1[i][j]);
19         }
20     }
21     printf("LLENADO DE MATRIZ 2\n");
22     for(int i=0; i<m; i++){
23         for (int j=0; j<n; j++){
24             printf("Ingrese numeros:");
25             scanf("%i", &mat2[i][j]);
26         }
27     }
28     for(int i=0; i<m; i++){
29         for (int j=0; j<n; j++){
30             printf("La matriz 1 es: %i \n", mat1[i][j]);
31         }
32     }
33     for(int i=0; i<m; i++){
34         for (int j=0; j<n; j++){
35             printf("la matriz 2 es: %i \n", mat2[i][j]);

```

```

34         for (int j=0; j<n; j++){
35             printf("la matriz 2 es: %i \n", mat2[i][j]);
36         }
37     }
38     for (int i = 0; i < m; i++){
39         for (int j = 0; j < n; j++){
40             suma=mat1[i][j]+mat2[i][j];
41             printf("La suma es: %i\n", suma);
42         }
43     }
44     return 0;
45 }

```

```

~/documents
$ ./matriz
Ingrese un numero de filas m y columnas n:
2 2
LLENADO DE MATRIZ 1
Ingrese numeros:1
Ingrese numeros:2
Ingrese numeros:3
Ingrese numeros:4
LLENADO DE MATRIZ 2
Ingrese numeros:1
Ingrese numeros:2
Ingrese numeros:3
Ingrese numeros:4
La matriz 1 es: 1
La matriz 1 es: 2
La matriz 1 es: 3
La matriz 1 es: 4
La matriz 2 es: 1
La matriz 2 es: 2
La matriz 2 es: 3
La matriz 2 es: 4
La suma es: 2
La suma es: 4
La suma es: 6
La suma es: 8

nadia@LAPTOP-VR22P14N ~/documents
$

```

Conclusiones.

Los arreglos nos permiten hacer un conjunto de operaciones para manipular los datos guardados en ellos, estas operaciones son: ordenar, buscar, insertar, eliminar, modificar entre otras. Los arreglos son de mucha utilidad cuando manejamos una lista de información y así elaborarla de manera rápida y óptima para nuestros programas.