



## 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):* MIRANDA GUTIERREZ CELINE  
TORRES MENDOZA ALEXA

*No. de Equipo de  
cómputo empleado:* 46 y 47

*No. de Lista o Brigada:* 30 y 49

*Semestre:* 2020-1

*Fecha de entrega:* 28/10/2019

---

*Observaciones:*

---

---

# CALIFICACIÓN:

---

## PRÁCTICA NO.11 “ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES”

### INTRODUCCIÓN

Un **arreglo** es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria RAM (sirve para almacenar datos en forma temporal).

#### Arreglos unidimensionales

Es un tipo de datos estructurado que está formado de una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. Están formados por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre, y se diferencian por la posición que tiene cada elemento dentro del arreglo de datos. Al declarar un arreglo, se debe inicializar sus elementos antes de utilizarlos. Para declarar un arreglo tiene que indicar su tipo, un nombre único y la cantidad de elementos que va a contener.

#### Arreglos multidimensionales

Es un tipo de dato estructurado, que está compuesto por dimensiones. Para hacer referencia a cada componente del arreglo es necesario utilizar  $n$  índices, uno para cada dimensión. El término dimensión representa el número de índices utilizados para referirse a un elemento particular en el arreglo. Los arreglos de más de una dimensión se llaman arreglos multidimensionales.

### OBJETIVO

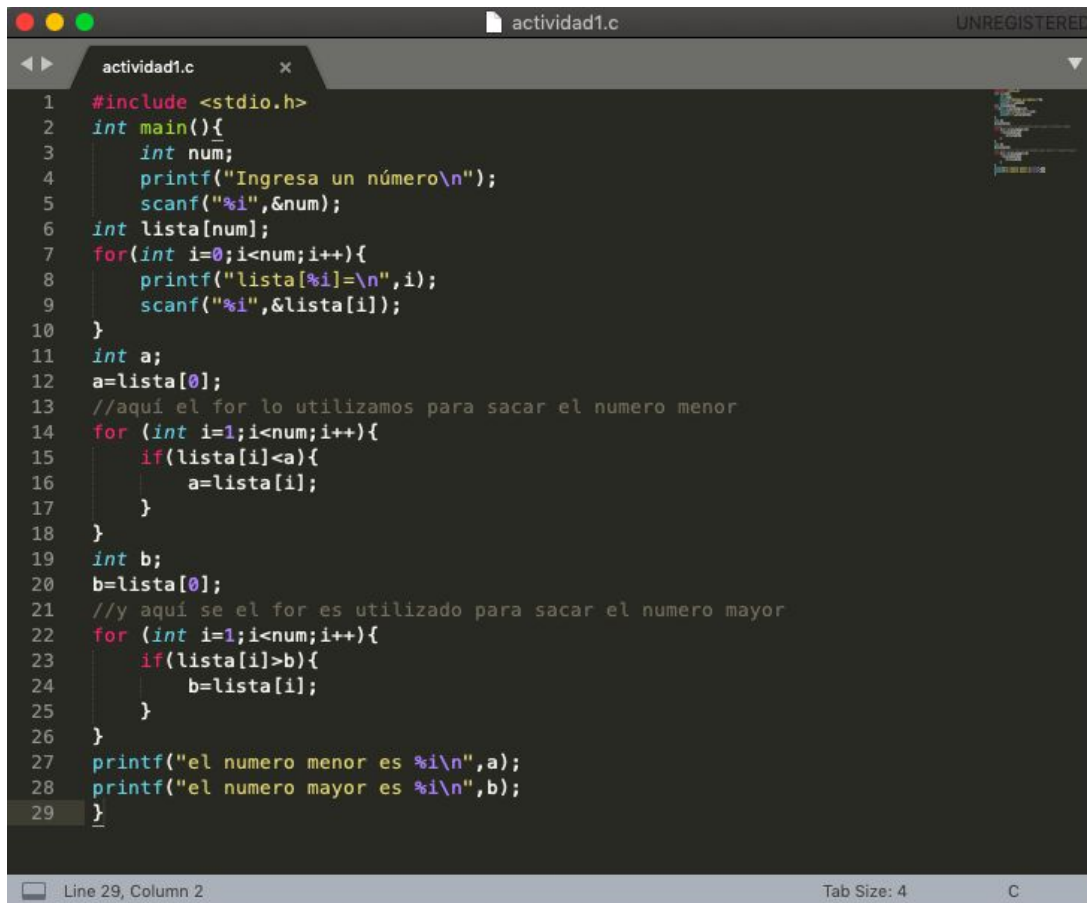
Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

### 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.



```
1  #include <stdio.h>
2  int main(){
3      int num;
4      printf("Ingresa un número\n");
5      scanf("%i",&num);
6      int lista[num];
7      for(int i=0;i<num;i++){
8          printf("lista[%i]=\n",i);
9          scanf("%i",&lista[i]);
10     }
11     int a;
12     a=lista[0];
13     //aquí el for lo utilizamos para sacar el numero menor
14     for (int i=1;i<num;i++){
15         if(lista[i]<a){
16             a=lista[i];
17         }
18     }
19     int b;
20     b=lista[0];
21     //y aquí se el for es utilizado para sacar el numero mayor
22     for (int i=1;i<num;i++){
23         if(lista[i]>b){
24             b=lista[i];
25         }
26     }
27     printf("el numero menor es %i\n",a);
28     printf("el numero mayor es %i\n",b);
29 }
```

Line 29, Column 2      Tab Size: 4      C

```

Last login: Mon Oct 28 22:46:14 on ttys000
MacBook-Air-de-Aline:~ alinemiranda$ ls
Creative Cloud Files  Library          actividad1
Desktop              Movies           actividad1.c
Documents            Music            main
Downloads            Pictures
FP_2020-1_9144       Public
MacBook-Air-de-Aline:~ alinemiranda$ gcc actividad1.c -o actividad1
MacBook-Air-de-Aline:~ alinemiranda$ ./actividad1
Ingresa un número
32
lista[0]=
434
lista[1]=
545
lista[2]=
232
lista[3]=
77
lista[4]=
799
lista[5]=
45
lista[6]=
1
lista[7]=
2
lista[8]=
987
lista[9]=
2233
lista[10]=
2321
lista[11]=
11
lista[12]=
1321

```

```

lista[10]=
2321
lista[11]=
11
lista[12]=
1321
lista[13]=
545
lista[14]=
22
lista[15]=
77
lista[16]=
43
lista[17]=
99
lista[18]=
1000
lista[19]=
66674
lista[20]=
22231
lista[21]=
14
lista[22]=
64
lista[23]=
143
lista[24]=
656
lista[25]=
33
lista[26]=
66
lista[27]=
945
lista[28]=

```

```
alinemiranda — -bash — 80x37
lista[15]=
77
lista[16]=
43
lista[17]=
99
lista[18]=
1000
lista[19]=
66674
lista[20]=
22231
lista[21]=
14
lista[22]=
64
lista[23]=
143
lista[24]=
656
lista[25]=
33
lista[26]=
66
lista[27]=
945
lista[28]=
88
lista[29]=
663
lista[30]=
446
lista[31]=
745
el numero menor es 1
el numero mayor es 66674
MacBook-Air-de-Aline:~ alinemiranda$
```

## 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.

```
actividad2.c  UNREGISTERED

1  #include <stdio.h>
2  int main(){
3      int x, y;
4      printf("Ingrese columnas de la matriz\n");
5      scanf("%i",&x);
6      printf("Ingrese filas de la matriz\n");
7      scanf("%i",&y);
8
9      int m1[x][y];
10     int m2[x][y];
11     int m3[x][y];
12     printf("Llenar la matriz 1\n");
13     printf("\n");
14     for (int a=0;a<x; a++)
15     {
16         for(int b=0; b<y; b++)
17         {
18             printf("lugar[%i][%i]\n",a+1,b+1);
19             scanf("%i",&m1[a][b]);
20         }
21     }
22     printf("\n");
23     printf("Llenar la matriz 2\n");
24     printf("\n");
25     for (int a=0; a < x; a++)
26     {
27         for(int b=0; b < y; b++)
28         {
29             printf("lugar[%i][%i]\n",a+1,b+1);
30             scanf("%i",&m2[a][b]);
31         }
    }
```

Line 24, Column 18 Tab Size: 4 C

```
alinemiranda — -bash — 80x24

Last login: Mon Oct 28 23:29:25 on ttys000
MacBook-Air-de-Aline:~ alinemiranda$ ls
Creative Cloud Files  Library          actividad1
Desktop              Movies           actividad1.c
Documents            Music            actividad2
Downloads            Pictures         actividad2.c
FP_2020-1_9144       Public           main
MacBook-Air-de-Aline:~ alinemiranda$ gcc actividad2.c -o actividad2
MacBook-Air-de-Aline:~ alinemiranda$ ./actividad2
Ingrese columnas de la matriz
3
Ingrese filas de la matriz
3
Llenar la matriz 1

lugar[1][1]
1
lugar[1][2]
2
lugar[1][3]
3
lugar[2][1]
4
lugar[2][2]
```

```
alinemiranda — -bash — 80x24
4
lugar[2][2]
4
lugar[2][3]
5
lugar[3][1]
6
lugar[3][2]
7
lugar[3][3]
8

Llenar la matriz 2

lugar[1][1]
9
lugar[1][2]
8
lugar[1][3]
7
lugar[2][1]
6
lugar[2][2]
5

alinemiranda — -bash — 80x24
lugar[1][1]
9
lugar[1][2]
8
lugar[1][3]
7
lugar[2][1]
6
lugar[2][2]
5
lugar[2][3]
4
lugar[3][1]
3
lugar[3][2]
2
lugar[3][3]
1

La suma de las dos matrices es:
10      10      10
10      9       9
9       9       9
MacBook-Air-de-Aline:~ alinemiranda$
```

## CONCLUSIONES

Con esta práctica pudimos concluir que el uso de los arreglos unidimensionales y multidimensionales son de mucha importancia debido a que gracias a estos se puede obtener datos de cualquier tipo de una forma ordenada. Es decir, este tipo de herramienta puede aplicarse hasta en una empresa, como por ejemplo saber el salario del total de sus empleados. Al utilizar esta herramienta se puede ahorrar tiempo y aumentaría la exactitud de los datos que se desean saber de una manera mucho más

rápida. Se puede utilizar en todo tipo de área y con comandos básicos puede realizarse de manera exitosa nuestros arreglos.