



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:*

Marco Antonio Martínez Quintana

*Asignatura:*

Fundamentos de programación

*Grupo:*

3

*No de Práctica(s):*

11

*Integrante(s):*

Pineda Cruz Tania

*No. de Equipo de  
cómputo empleado:*

No aplica

*No. de Lista o Brigada:*

34

*Semestre:*

1er semestre

*Fecha de entrega:*

04/01/2021

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **Guía práctica de estudio 11: Arreglos unidimensionales y 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.

### **Actividades**

- Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.
- Resolver un problema que requiera el uso de un arreglo de 2 dimensiones, a través de un programa en lenguaje C.
- Manipular arreglos a través de índices y apuntadores.

### **Introducción**

#### Arreglo

Conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. A cada elemento del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico.

#### Arreglos unidimensionales

Puede almacenar a  $n$  elementos del mismo tipo y además permite el acceso a cada uno de estos elementos. Se dice que es unidimensional ya que, solo tienen una dimensión.

#### Apuntadores

Es una variable que contiene la dirección de una variable, es decir, hace referencia a la localidad de memoria de otra variable. La declaración de una variable apuntador inicia con el carácter \*.

#### Arreglos multidimensionales

Es un arreglo con más de dos dimensiones. En una matriz, las dos dimensiones se representan con filas y columnas. De manera práctica se puede considerar que la primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente.

## Resultados

### Código (arreglo unidimensional while)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c gauss.c gausswhile.c menuSalir.c subMenu.c practica10.c practica11.c
1 #include <stdio.h>
2 /*
3  Este programa genera un arreglo unidimensional de 5 elementos y los
4  accede a cada elemento del arreglo a través de un ciclo while.
5 */
6
7 int main ()
8 {
9     char ao=162;
10    #define TAMANO 5
11    int lista[TAMANO] = {10, 8, 5, 8, 7};
12
13    int indice = 0;
14
15    printf("\nlista\n");
16    while(indice < 5)
17    {
18        printf("\nCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
19        indice += 1;
20    }
21
22    printf("\n");
23
24    return 0;
25 }
```

Simbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica11.c -o practica11.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 5
Calificación del alumno 4 es 8
Calificación del alumno 5 es 7
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

### Código (arreglo unidimensional for)


```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c gauss.c gausswhile.c menuSalir.c subMenu.c practica10.c practica11.c
1 #include <stdio.h>
2 /*
3  Este programa genera un arreglo unidimensional de 5 elementos y
4  accede a cada elemento del arreglo a través de un ciclo for
5 */
6
7 int main ()
8 {
9     char ao=162;
10    #define TAMANO 5
11    int lista[TAMANO] = {10, 8, 5, 8, 7};
12
13    printf("\nlista\n");
14    for(int indice = 0 ; indice < 5 ; indice++)
15    {
16        printf("\nCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
17    }
18
19    printf("\n");
20
21    return 0;
22 }
```

Simbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica11.c -o practica11.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 5
Calificación del alumno 4 es 8
Calificación del alumno 5 es 7
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

## Código (apuntadores)


```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c practica10.c practica11.c
1 #include <stdio.h>
2 /*
3  * Este programa crea un apuntador de tipo carácter.
4  */
5
6 int main ()
7 {
8     char aa=160, ao=162, *ap, c = 'a';
9     ap = &c;
10
11     printf("Carácter: %c\n",aa,*ap);
12     printf("Código ASCII: %d\n",ao,*ap);
13     printf("Dirección de memoria: %d\n",ao,ap);
14
15     return 0;
16 }
```

 Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica11.c -o practica11.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
Carácter: a
Código ASCII: 97
Dirección de memoria: 6422295
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

## Código (apuntadores)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c practica10.c practica11.c
1 #include <stdio.h>
2 /*
3  * Este programa accede a las localidades de memoria de distintas variables a
4  * través de un apuntador.
5  */
6
7 int main ()
8 {
9     int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
10     int *apEnt;
11     apEnt = &a;
12
13     printf("a = 5, b = 10; c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
14     printf("apEnt = %a\n");
15
16     b = *apEnt;
17     printf("b = *apEnt \t-> b = %i\n", b);
18
19     b = *apEnt + 1;
20     printf("b = *apEnt + 1 \t-> b = %i\n", b);
21
22     *apEnt = 0;
23     printf("*apEnt = 0 \t-> a = %i\n", a);
24
25     apEnt = &c[0];
26     printf("apEnt = %c[0] \t-> apEnt = %i\n", *apEnt);
27
28     return 0;
29 }
```

 Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
a = 5, b = 10; c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}
apEnt = &a
b = *apEnt      -> b = 5
b = *apEnt + 1  -> b = 6
*apEnt = 0      -> a = 0
apEnt = &c[0]   -> apEnt = 5
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>_
```

## Código (apuntadores)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c practica10.c practica11.c
1 #include <stdio.h>
2
3 /*
4  Este programa trabaja con aritmética de punteros para acceder a los
5  valores de un arreglo.
6 */
7
8 int main ()
9 {
10     int arr[] = {5, 4, 3, 2, 1};
11     int *apArr;
12     apArr = arr;
13
14     printf("int arr[] = {5, 4, 3, 2, 1}\n");
15     printf("apArr = %s\n", arr);
16
17     int x = *apArr;
18     printf("x = *apArr \t -> x = %d\n", x);
19
20     x = *(apArr+1);
21     printf("x = *(apArr+1) \t -> x = %d\n", x);
22
23     x = *(apArr+2);
24     printf("x = *(apArr+2) \t -> x = %d\n", x);
25
26     return 0;
27 }
```

**C:\** Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
int arr[] = {5, 4, 3, 2, 1};
apArr = &arr[0]
x = *apArr          -> x = 5
x = *(apArr+1)      -> x = 4
x = *(apArr+2)      -> x = 3

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>_
```

## Código (apuntadores en ciclo for)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c practica10.c practica11.c
1 #include <stdio.h>
2
3 /*
4  Este programa genera un arreglo unidimensional de 5 elementos y
5  accede a cada elemento del arreglo a través de un puntero
6  utilizando un ciclo for
7 */
8
9 int main ()
10 {
11     char ao="162";
12
13     #define TAMANO 5
14     int lista[TAMANO] = {10, 8, 5, 8, 7};
15     int *ap = lista;
16
17     printf("\nLista\n");
18     for (int indice = 0 ; indice < 5 ; indice++)
19     {
20         printf("\nCalificación del alumno %d es %d",ao, indice+1, *(ap+indice));
21     }
22
23     printf("\n");
24
25     return 0;
26 }
```

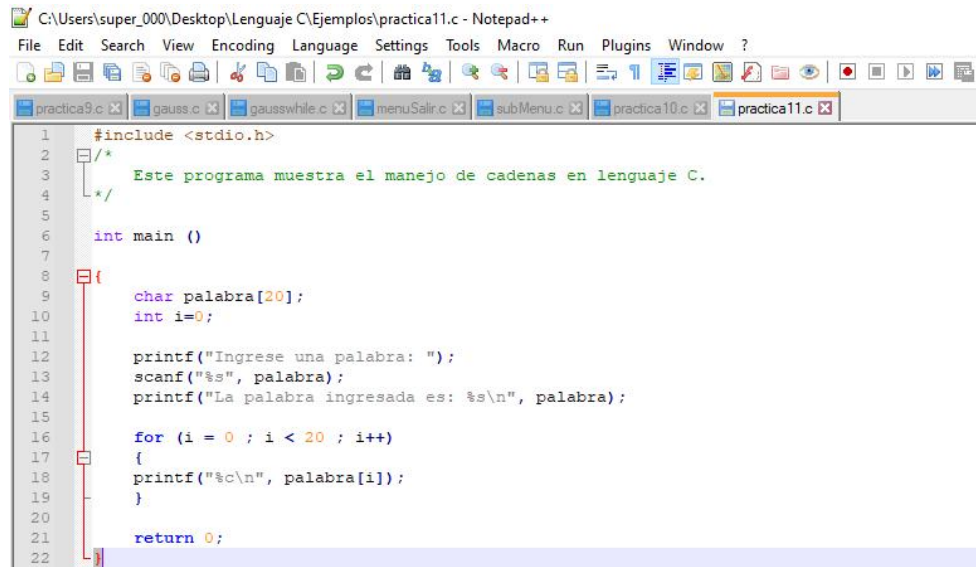
**C:\** Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
Lista

Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 5
Calificación del alumno 4 es 8
Calificación del alumno 5 es 7


C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>_
```

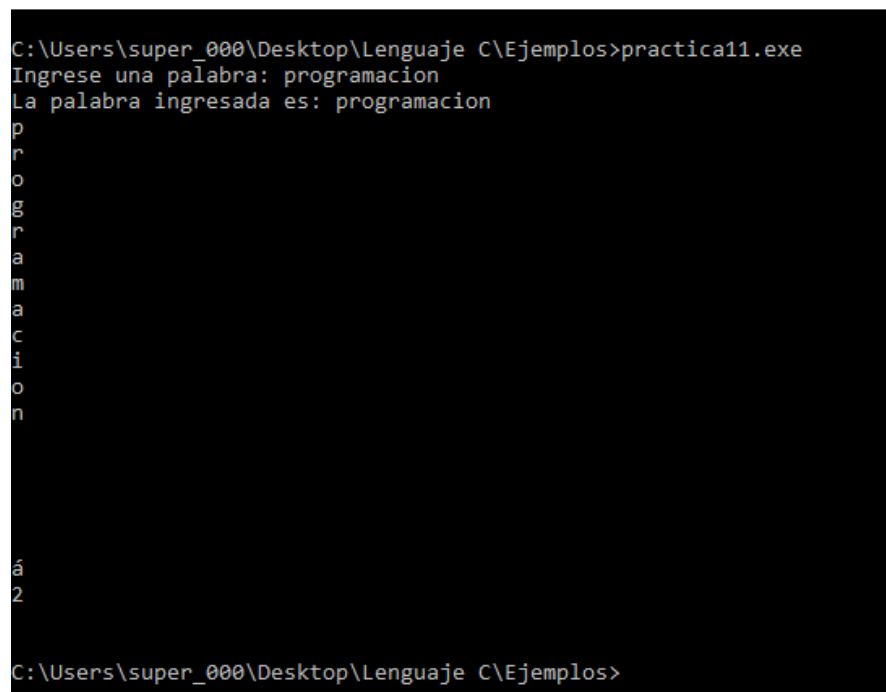
## Código (apuntadores en cadenas)



The screenshot shows a Notepad++ window titled "C:\Users\super\_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and development. The tab bar shows several open files: practica9.c, gauss.c, gausswhile.c, menuSalir.c, subMenu.c, practica10.c, and practica11.c. The main editor displays the following C code:

```
1  #include <stdio.h>
2
3  /*
4   * Este programa muestra el manejo de cadenas en lenguaje C.
5   */
6
7  int main ()
8  {
9      char palabra[20];
10     int i=0;
11
12     printf("Ingrese una palabra: ");
13     scanf("%s", palabra);
14     printf("La palabra ingresada es: %s\n", palabra);
15
16     for (i = 0 ; i < 20 ; i++)
17     {
18         printf("%c\n", palabra[i]);
19     }
20
21     return 0;
22 }
```

 Símbolo del sistema



The screenshot shows a Windows command prompt window with the following text:

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
Ingrese una palabra: programacion
La palabra ingresada es: programacion
p
r
o
g
r
a
m
a
c
i
o
n

á
2

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

## Código (arreglos multidimensionales)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c gauss.c gausswhile.c menuSalir.c subMenu.c practica10.c practica11.c
1 #include <stdio.h>
2
3 /*
4  Este programa genar un arreglo de dos dimensiones (arreglo
5  multidimensional) y accede a sus elementos a través de dos ciclos
6  for, uno anidado dentro de otro.
7  */
8
9 int main ()
10 {
11     char matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, j;
13
14     printf("Imprimir Matriz\n");
15
16     for (i = 0 ; i < 3 ; i++)
17     {
18         for (j=0 ; j<3 ; j++)
19         {
20             printf("%d, ", matriz[i][j]);
21         }
22         printf("\n");
23     }
24
25     return 0;
26 }
```

Simbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica11.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
practica9.c gauss.c gausswhile.c menuSalir.c subMenu.c practica10.c practica11.c
1 #include <stdio.h>
2
3 /*
4  Este programa genar un arreglo de dos dimensiones (arreglo
5  multidimensional) y accede a sus elementos a través de dos ciclos
6  for, uno anidado dentro de otro.
7  */
8
9 int main ()
10 {
11     char matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, j;
13
14     printf("Imprimir Matriz\n");
15
16     for (i = 0 ; i < 3 ; i++)
17     {
18         for (j=0 ; j<3 ; j++)
19         {
20             printf("%d, ", matriz[2][1]);
21         }
22         printf("\n");
23     }
24
25     return 0;
26 }
```

Simbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica11.exe
Imprimir Matriz
8, 8, 8,
8, 8, 8,
8, 8, 8,

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

## **Conclusiones**

Los arreglos nos permiten manejar de forma sencilla y directa conjuntos de datos del mismo tipo, de los cuales conocemos su cantidad y con los cuales se realizarán operaciones similares.

La realización de esta practica en un principio fue algo confuso puesto que, me costó trabajo comprender la funcionalidad que tenía cada uno, sin embargo, al ir avanzando cada vez más en el desarrollo de este y leyendo las descripciones, pude comprender el tema y finalizarlo de manera exitosa.

## **Referencias**

Manual de las prácticas de Fundamentos de Programación, UNAM.