



Arreglos unidimensionales y multidimensionales

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M. I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No. de Práctica: 11

Integrante(s): Sánchez García Rocío

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista: 47

Semestre: 2021-1

Fecha de entrega: Domingo 10 de enero de 2020

Observaciones:

CALIFICACIÓN: _____

Práctica 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 dos dimensiones, a través de un programa en lenguaje C.
- ❖ Manipular arreglos a través de índices y apuntadores.

Introducción

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

Desarrollo

Arreglos unidimensionales

La sintaxis para definir un arreglo en lenguaje C es el siguiente:

```
tipoDeDato nombre [tamaño]
```

donde nombre se refiere al identificador del arreglo, tamaño es un numero entero y define el número máximo de elementos que puede contener el arreglo. Un arreglo puede ser de los tipos de dato entero, real, carácter o estructura.

Código (arreglo unidimensional while)

```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X
gauss.c factorial.c factorialWhile.c menuSalir.c submenu.c calculadora5.c unidimensionalWhile.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 int main ()
7 {
8     #define TAMANO 5
9     char ao = 162;
10    int lista[TAMANO] = {10, 8, 5, 8, 7};
11    int indice = 0;
12    printf("\tLista\n");
13    while (indice < 5 )
14    {
15        printf("\n\tCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
16        indice += 1; // análogo a indice = indice + 1;
17    }
18    printf("\n");
19    return 0;
20 }

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc unidimensionalWhile.c -o unidimensionalWhile.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>unidimensionalWhile.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\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Código (arreglo unidimensional for)

```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X
gauss.c factorial.c factorialWhile.c menuSalir.c submenu.c calculadora5.c unidimensionalWhile.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 int main ()
7 {
8     #define TAMANO 5
9     char ao = 162;
10    int lista[TAMANO] = {10, 8, 5, 8, 7};
11    int indice = 0;
12    printf("\tLista\n");
13    while (indice < 5 )
14    {
15        printf("\n\tCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
16        indice += 1; // análogo a indice = indice + 1;
17    }
18    printf("\n");
19    return 0;
20 }

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc unidimensionalWhile.c -o unidimensionalWhile.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>unidimensionalWhile.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\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Apuntadores

Hace referencia a la localidad de memoria de otra variable. Debido a que los apuntadores trabajan directamente con la memoria, a través de ellos se accede con rapidez a un dato.

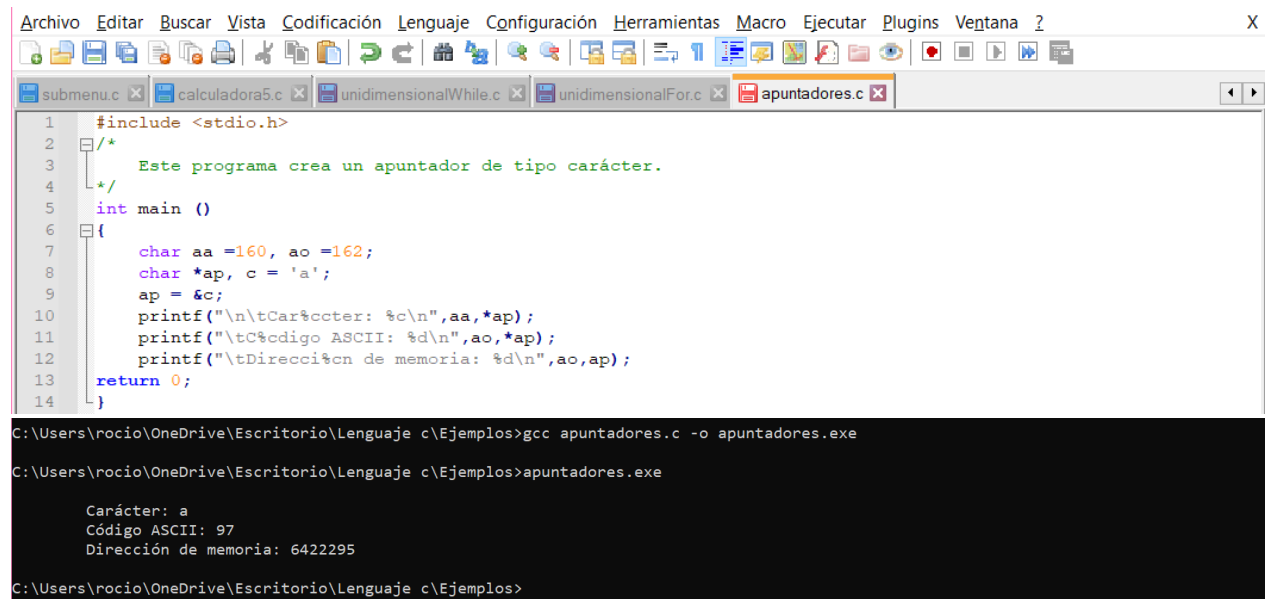
La sintaxis para declarar un apuntador y para asignarle la dirección de memoria de otra variable es, respectivamente:

```
TipoDeDato *apuntador, variable;  
apuntador=&variable;
```

La declaración de una variable apuntador inicia con el carácter *. Cuando a una variable le antecede un *ampersand*, lo que se hace es acceder a la dirección de memoria de la misma (es lo que pasa cuando se lee un dato con *scanf*).

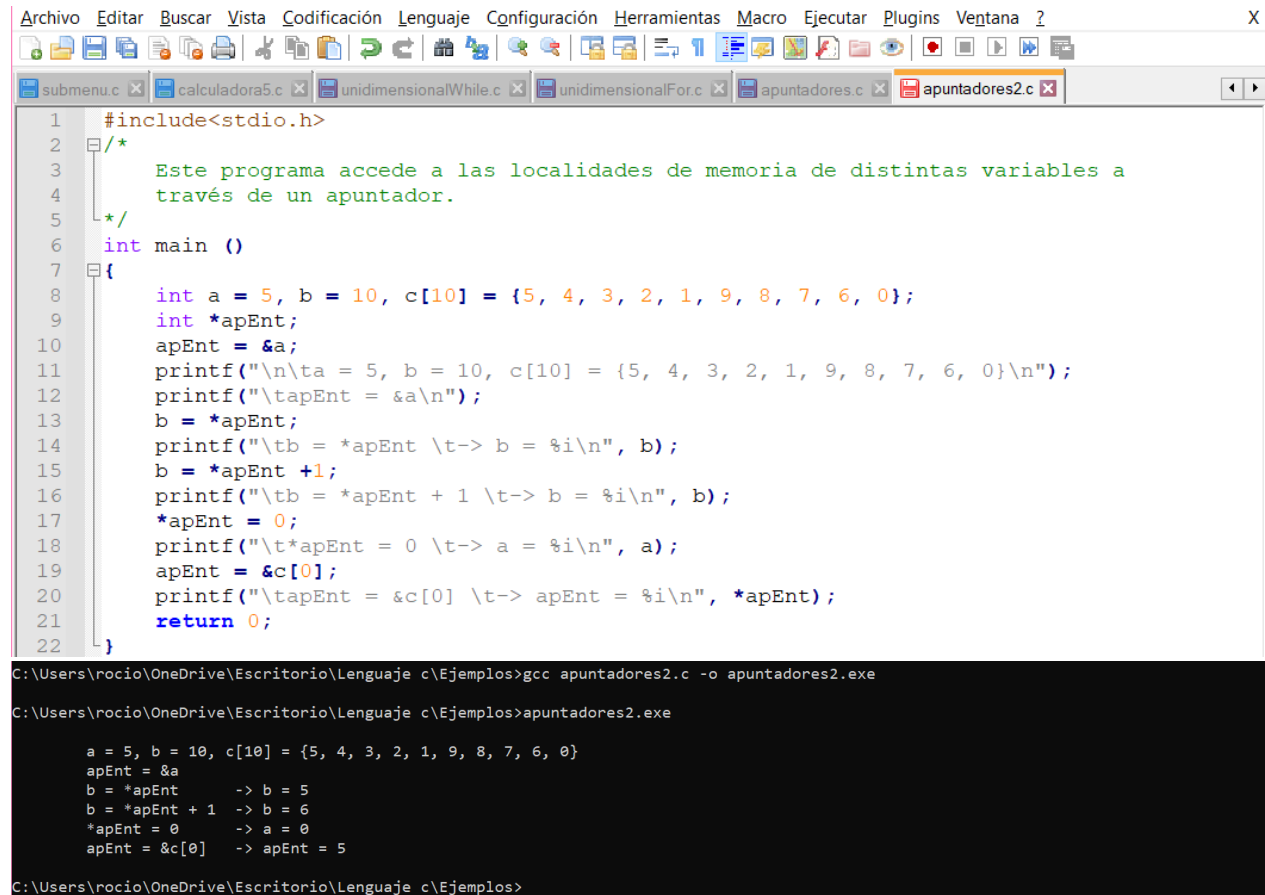
Solo pueden apuntar a direcciones de memoria del mismo tipo de dato con el que fueron declarados; para acceder al contenido de dicha dirección, a la variable apuntador se le antepone *.

Código (apuntadores)



```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X  
submenu.c calculadora5.c unidimensionalWhile.c unidimensionalFor.c apuntadores.c  
1 #include <stdio.h>  
2 /*  
3  Este programa crea un apuntador de tipo carácter.  
4  */  
5 int main ()  
6 {  
7     char aa =160, ao =162;  
8     char *ap, c = 'a';  
9     ap = &c;  
10    printf("\n\tCarácter: %c\n",aa,*ap);  
11    printf("\tCódigo ASCII: %d\n",ao,*ap);  
12    printf("\tDirección de memoria: %d\n",ao,ap);  
13    return 0;  
14 }  
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apuntadores.c -o apuntadores.exe  
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apuntadores.exe  
Carácter: a  
Código ASCII: 97  
Dirección de memoria: 6422295  
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Código (apuntadores)



The screenshot shows a C code editor with a menu bar (Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Plugins, Ventana, ?) and a toolbar. The code in the editor is as follows:

```
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 int main ()
7 {
8     int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
9     int *apEnt;
10    apEnt = &a;
11    printf("\n\ta = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
12    printf("\tapEnt = &a\n");
13    b = *apEnt;
14    printf("\tb = *apEnt \t-> b = %i\n", b);
15    b = *apEnt + 1;
16    printf("\tb = *apEnt + 1 \t-> b = %i\n", b);
17    *apEnt = 0;
18    printf("\t*apEnt = 0 \t-> a = %i\n", a);
19    apEnt = &c[0];
20    printf("\tapEnt = &c[0] \t-> apEnt = %i\n", *apEnt);
21    return 0;
22 }
```

Below the code editor, a terminal window shows the execution of the program:

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apuntadores2.c -o apuntadores2.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apuntadores2.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
```

El nombre de un arreglo es un apuntador fijo al primero de sus elementos; por lo que las siguientes instrucciones, para el código de arriba, son equivalentes

```
apEnt = &c[0];
apEnt = c;
```

Código (apuntadores)

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X
submenu.c  calculadora5.c  unidimensionalWhile.c  unidimensionalFor.c  apuntadores.c  apuntadores2.c  apuntadores3.c
1  #include <stdio.h>
2  /*
3   Este programa trabaja con aritmética de apuntadores para acceder a los
4   valores de un arreglo.
5  */
6  int main ()
7  {
8      int arr[] = {5, 4, 3, 2, 1};
9      int *apArr;
10     apArr = arr;
11     printf("\n\tint arr[] = {5, 4, 3, 2, 1};\n");
12     printf("\tapArr = &arr[0]\n");
13     int x = *apArr;
14     printf("\tx = *apArr \t -> x = %d\n", x);
15     x = *(apArr+1);
16     printf("\tx = *(apArr+1) \t -> x = %d\n", x);
17     x = *(apArr+2);
18     printf("\tx = *(apArr+1) \t -> x = %d\n", x);
19     return 0;
20 }

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apuntadores3.c -o apuntadores3.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apuntadores3.exe

int arr[] = {5, 4, 3, 2, 1};
apArr = &arr[0]
x = *apArr      -> x = 5
x = *(apArr+1)  -> x = 4
x = *(apArr+1)  -> x = 3

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Código (apuntadores en ciclo for)

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X
calculadora5.c  unidimensionalWhile.c  unidimensionalFor.c  apuntadores.c  apuntadores2.c  apuntadores3.c  apuntadoresf
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 apuntador
5   utilizando un ciclo for.
6  */
7  int main ()
8  {
9      #define TAMANO 5
10     char ao= 162;
11     int lista[TAMANO] = {10, 8, 5, 8, 7};
12     int *ap = lista;
13     printf("\n\t\tLista\n");
14     for (int indice = 0 ; indice < 5 ; indice++)
15     {
16         printf("\n\tCalificaci%cn del alumno %d es %d",ao, indice+1, *(ap+indice));
17     }
18     printf("\n");
19     return 0;
20 }
```

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apunadoresFor.c -o apunadoresFor.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apunadoresFor.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\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_

```

Código (apuntadores en cadenas)

```

Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
[Icons]
apunadores2.c  apunadores3.c  apunadoresFor.c  apuntadorCadena.c  x.c
1  #include <stdio.h>
2  /*
3   Este programa muestra el manejo de cadenas en lenguaje C.
4  */
5  int main()
6  {
7      char palabra[20];
8      int i=0;
9      printf("\n\tIngrese una palabra: ");
10     scanf("%s", palabra);
11     printf("\n\tLa palabra ingresada es: %s\n", palabra);
12     for(i = 0 ; i < 20 ; i++)
13     {
14         printf("\n\t%c\n", palabra[i]);
15     }
16     return 0;
17 }
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc apuntadorCadena.c -o apuntadorCadena.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>apuntadorCadena.exe

Ingrese una palabra: hola

La palabra ingresada es: hola

h
o
l
a

@
@
P
@
@

```

Arreglos multidimensionales

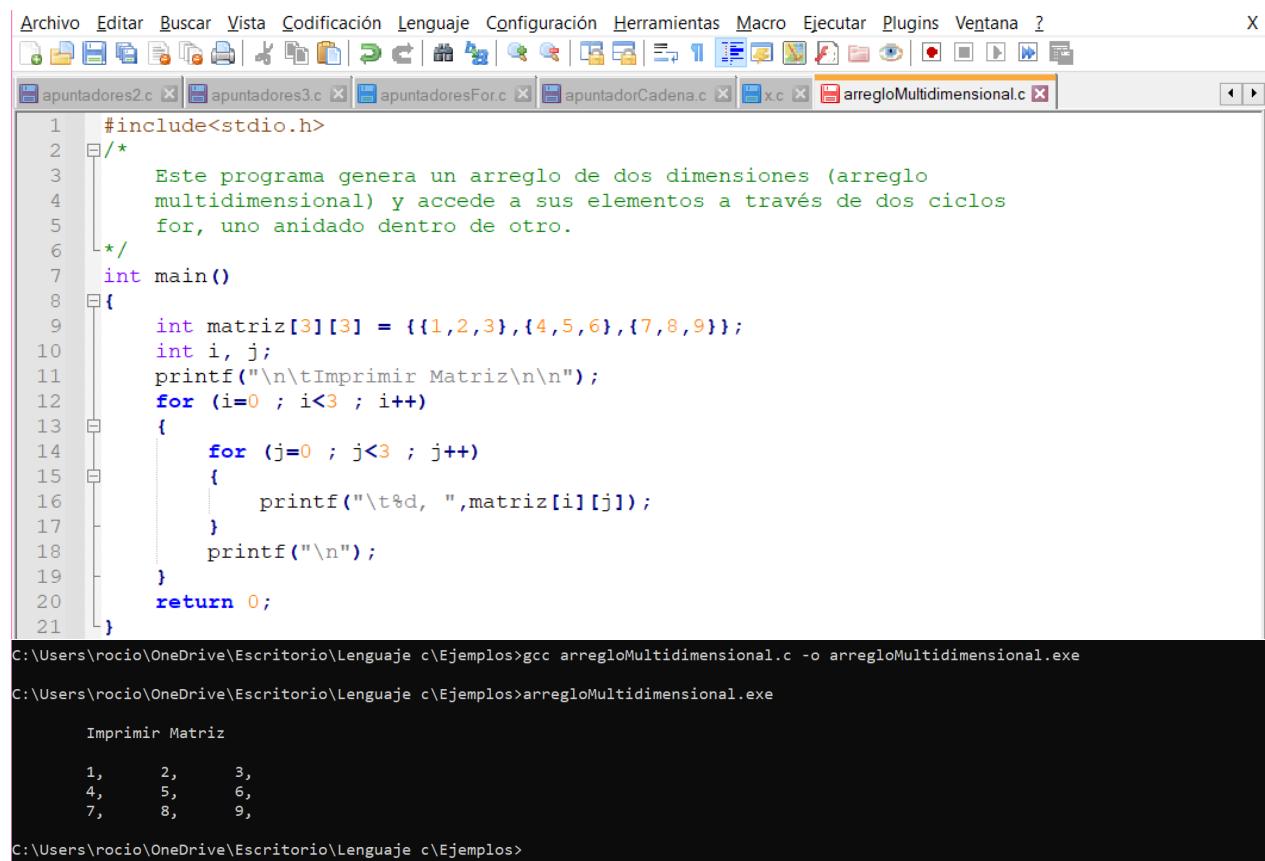
Su sintaxis es la siguiente:

```
tipoDato nombre[tamaño][tamaño]...[tamaño];
```

nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión (el número de dimensiones está determinado por el número de corchetes). Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o estructura.

La primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente. Sin embargo, en la memoria cada elemento del arreglo se guarda de forma contigua, por lo tanto, se puede recorrer un arreglo multidimensional con punteros.

Código (arreglos multidimensionales)



```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X
apuntadores2.c apuntadores3.c apuntadoresFor.c apuntadorCadena.c x.c arregloMultidimensional.c
1 #include<stdio.h>
2 /*
3  Este programa genera un arreglo de dos dimensiones (arreglo
4  multidimensional) y accede a sus elementos a través de dos ciclos
5  for, uno anidado dentro de otro.
6  */
7 int main()
8 {
9     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
10    int i, j;
11    printf("\n\tImprimir Matriz\n\n");
12    for (i=0 ; i<3 ; i++)
13    {
14        for (j=0 ; j<3 ; j++)
15        {
16            printf("\t%d, ",matriz[i][j]);
17        }
18        printf("\n");
19    }
20    return 0;
21 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arregloMultidimensional.c -o arregloMultidimensional.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arregloMultidimensional.exe

Imprimir Matriz

1,    2,    3,
4,    5,    6,
7,    8,    9,

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```


➤ **Actividades asignadas por el profesor**

Arreglos

Es un conjunto de datos del mismo tipo asociados por el nombre de una variable

Arreglos en C

Estos necesitan reservar la memoria a utilizar a través de una inicialización o reservación del espacio en memoria

Arreglos unidimensionales

Sintaxis:

```
int a[10]; //Reserva 10 espacios  
int b[]={1,2,3,4,5,6}; Reserva 6 espacios
```

Arreglos bidimensionales o multidimensionales

Sintaxis:

```
double c[10][10];// Reserva 100 espacios  
int datos[3][3]={1,2,3,4,5,6,7,8,9}// Reserva 9 espacios
```

Arreglos de tipo char

Sintaxis:

```
char nom[20]= "Hola"
```

Implementación:

Realizar un sistema de captura de calificaciones utilizando tanto arreglos unidimensionales como multidimensionales.

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos.c -o arreglos.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos.exe

Calificación 1: 10
Calificación 2: 9
Calificación 3: 10

Calificaciones capturadas
10    9    10

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Captura de calificaciones y promedio

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos.c -o arreglos.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos.exe

Calificación 1: 10
Calificación 2: 9
Calificación 3: 10

Calificaciones capturadas
10    9    10

El promedio es: 9.666667

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos.c -o arreglos.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos.exe

Calificación 1: 10

Calificación 2: 9

Calificación 3: 10

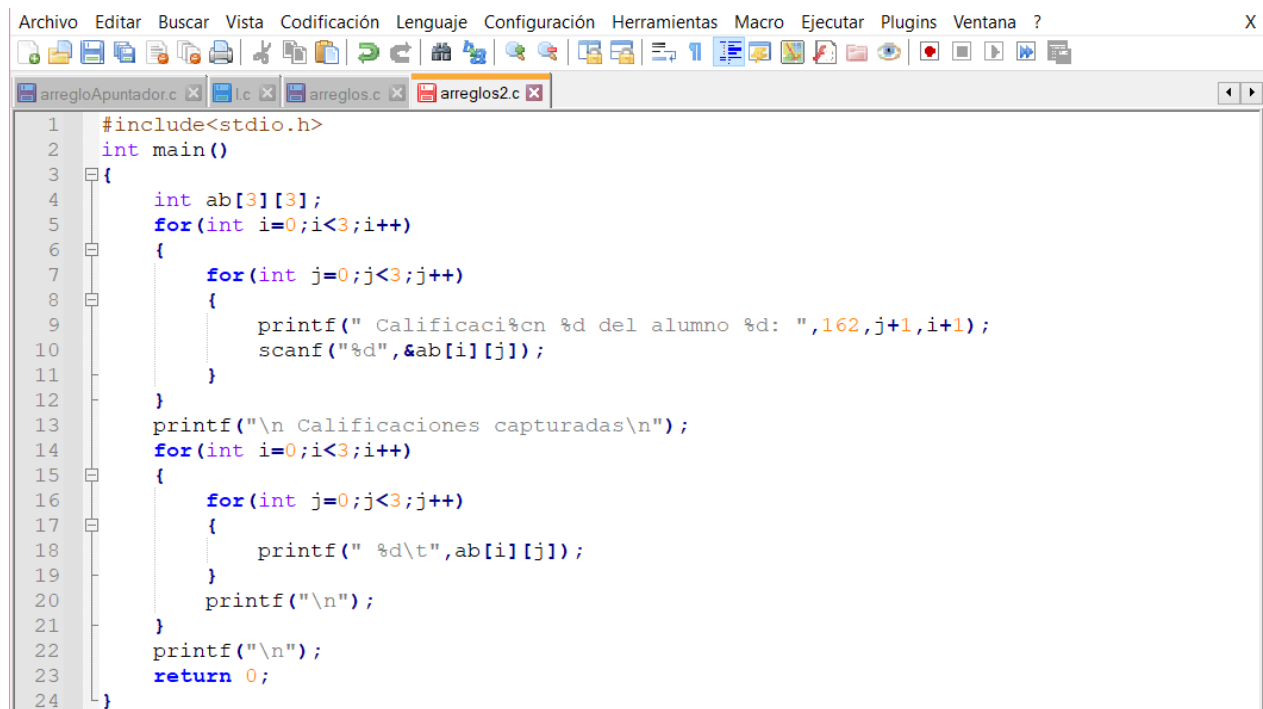
Calificaciones capturadas
10      9      10

El promedio es: 9.666667

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_

```

Arreglo bidimensional



```

Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
[Iconos de herramientas]

arregloApuntador.c  l.c  arreglos.c  arreglos2.c

1  #include<stdio.h>
2  int main()
3  {
4      int ab[3][3];
5      for(int i=0;i<3;i++)
6      {
7          for(int j=0;j<3;j++)
8          {
9              printf(" Calificaci%cn %d del alumno %d: ",162,j+1,i+1);
10             scanf("%d",&ab[i][j]);
11         }
12     }
13     printf("\n Calificaciones capturadas\n");
14     for(int i=0;i<3;i++)
15     {
16         for(int j=0;j<3;j++)
17         {
18             printf(" %d\t",ab[i][j]);
19         }
20         printf("\n");
21     }
22     printf("\n");
23     return 0;
24 }

```

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos2.c -o arreglos2.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos2.exe

Calificación 1 del alumno 1: 10
Calificación 2 del alumno 1: 10
Calificación 3 del alumno 1: 9
Calificación 1 del alumno 2: 8
Calificación 2 del alumno 2: 10
Calificación 3 del alumno 2: 8
Calificación 1 del alumno 3: 7
Calificación 2 del alumno 3: 7
Calificación 3 del alumno 3: 9

Calificaciones capturadas
10      10      9
8        10      8
7         7      9

```

```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X
arregloApuntador.c x l.c x arreglos.c x arreglos2.c x
1 #include<stdio.h>
2 int main()
3 {
4     int ab[3][3];
5     for(int i=0;i<3;i++)
6     {
7         for(int j=0;j<3;j++)
8         {
9             printf(" Calificaci%cn %d del alumno %d: ",162,j+1,i+1);
10             scanf("%d",&ab[i][j]);
11         }
12     }
13     printf("\n Calificaciones capturadas\n");
14     for(int i=0;i<3;i++)
15     {
16         for(int j=0;j<3;j++)
17         {
18             printf(" %d\t",ab[i][j]);
19         }
20         printf("\n");
21     }
22     printf("\n");
23     float p[3];
24     int a=0;
25     for(int i=0;i<3;i++)
26     {
27         a=0;
28         for(int j=0;j<3;j++)
29         {
30             a=a+ab[i][j];
31         }
32         p[i]=a/3.0;
33         printf("\nEl promedio del alumno %d es: %f",i+1,p[i]);
34     }
35     return 0;
36 }
```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos2.c -o arreglos2.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos2.exe

```
Calificación 1 del alumno 1: 10
Calificación 2 del alumno 1: 10
Calificación 3 del alumno 1: 9
Calificación 1 del alumno 2: 7
Calificación 2 del alumno 2: 8
Calificación 3 del alumno 2: 8
Calificación 1 del alumno 3: 6
Calificación 2 del alumno 3: 9
Calificación 3 del alumno 3: 9
```

Calificaciones capturadas

```
10    10    9
7      8    8
6      9    9
```

El promedio del alumno 1 es: 9.666667

El promedio del alumno 2 es: 7.666667

El promedio del alumno 3 es: 8.000000

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X
[arregloApuntador.c] [l.c] [arreglos.c] [arreglos2.c]
1  #include<stdio.h>
2  int main()
3  {
4      int ab[3][3];
5      for(int i=0;i<3;i++)
6      {
7          for(int j=0;j<3;j++)
8          {
9              printf(" Calificaci%cn %d del alumno %d: ",162,j+1,i+1);
10             scanf("%d",&ab[i][j]);
11         }
12     }
13     printf("\n Calificaciones capturadas\n");
14     for(int i=0;i<3;i++)
15     {
16         for(int j=0;j<3;j++)
17         {
18             printf(" %d\t",ab[i][j]);
19         }
20         printf("\n");
21     }
22     printf("\n");
23     float p[3];
24     int a=0;
25     for(int i=0;i<3;i++)
26     {
27         a=0;
28         for(int j=0;j<3;j++)
29         {
30             a=a+ab[i][j];
31         }
32         p[i]=a/3.0;
33         printf("\n El promedio del alumno %d es: %f",i+1,p[i]);
34     }
35     a=0;
36     for(int i=0;i<3;i++)
37     {
38         a=a+p[i];
39     }
40     printf("\n\n El promedio general del grupo es: %f\n", (a/3.0));
41     return 0;
42 }
```

C:\Users\roci0\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc arreglos2.c -o arreglos2.exe

C:\Users\roci0\OneDrive\Escritorio\Lenguaje c\Ejemplos>arreglos2.exe

```
Calificación 1 del alumno 1: 10
Calificación 2 del alumno 1: 10
Calificación 3 del alumno 1: 9
Calificación 1 del alumno 2: 8
Calificación 2 del alumno 2: 7
Calificación 3 del alumno 2: 7
Calificación 1 del alumno 3: 6
Calificación 2 del alumno 3: 9
Calificación 3 del alumno 3: 8
```

Calificaciones capturadas

```
10    10    9
8      7    7
6      9    8
```

El promedio del alumno 1 es: 9.666667

El promedio del alumno 2 es: 7.333333

El promedio del alumno 3 es: 7.666667

El promedio general del grupo es: 7.666667

C:\Users\roci0\OneDrive\Escritorio\Lenguaje c\Ejemplos>

Ejercicio. Sistema de inventarios

Crear un sistema que almacene el inventario de una tienda en un arreglo y al final nos de la cantidad total de artículos que se tienen en existencia.

```
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
[Iconos de herramientas]

arregloMultidimensional.c  arregloApuntador.c  i.c  arreglos.c  arreglos2.c  inventario.c  inventario2.c

1  #include<stdio.h>
2  int main()
3  {
4      int a[8];
5      //Mensaje
6      printf("\n --->INVENTARIO<---\n");
7      //Mostrar menú
8      printf("\n Ingrese la cantidad de productos según su clasificación\n",163,162);
9      printf("\n\t Sección 1: Abarrotes \n\t Sección 2: Enlatados\n\t Sección 3: Lácteos\n\t Sección 4: Botanas\n\t Sección 5: Confeitería\n\t Sección 6: Frutas y Verduras\n\t Sección 7: Bebidas\n\t Sección 8: Higiene personal\n");
10     for(int i = 0; i < 8; i++)
11     {
12         printf("\n Cantidad de articulos en la Sección %d: ",162,i+1);
13         scanf("%d",&a[i]);
14     }
15     printf("\n");
16     int t = 0;
17     for(int i = 0; i < 8; i++)
18     {
19         t = t+a[i];
20     }
21     printf("\n La cantidad de articulos en la tienda es de: %d\n",t);
22     return 0;
23 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc inventario.c -o inventario.exe
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>inventario.exe
```

```
--->INVENTARIO<---
```

```
Ingrese la cantidad de productos según su clasificación
```

```
Sección 1: Abarrotes
Sección 2: Enlatados
Sección 3: Lácteos
Sección 4: Botanas
Sección 5: Confeitería
Sección 6: Frutas y Verduras
Sección 7: Bebidas
Sección 8: Higiene personal
```

```
Cantidad de articulos en la Sección 1: 45
```

```
Cantidad de articulos en la Sección 2: 67
```

```
Cantidad de articulos en la Sección 3: 800
```

```
Cantidad de articulos en la Sección 4: 34
```

```
Cantidad de articulos en la Sección 5: 56
```

```
Cantidad de articulos en la Sección 6: 12
```

```
Cantidad de articulos en la Sección 7: 21
```

```
Cantidad de articulos en la Sección 8: 65
```

```
La cantidad de articulos en la tienda es de: 1100
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Conclusiones

- ❖ Existen muchas aplicaciones de los arreglos que son de mucha utilidad al momento de programar ya que se tiene la oportunidad de manipular datos, tienen la capacidad de realizar varias operaciones.
- ❖ Es posible almacenar grandes cifras ya que los arreglos pueden tener una o más dimensiones.
- ❖ Es de suma importancia poner saber para que sirven ya que de esto dependerá que la ejecución de un programa sea la correcta.

Bibliografía

Manual de prácticas del Laboratorio de Fundamentos de Programación, Facultad de ingeniería UNAM, recuperada el 30 de noviembre, en <http://lcp02.fi-b.unam.mx/>