



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de programación

## **Práctica 11**

Alumna: Rivera González Frida Alison

M.I. Marco Antonio Martínez Quintana

No. lista: 40

No. De equipo empleado: No aplica.

Semestre: 2021-1

Fecha de entrega: domingo 10 de enero de 2021

Grupo: 3

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.

## Actividad 1.-

```
1  #include<stdio.h>
2  /*
3   Este programa genera un arreglo unidimensional de 5 elementos y los accede
4   a cada elemento del arreglo a través de un ciclo while.
5  */
6
7  int main()
8  {
9      #define TAMANO 5
10     int lista[TAMANO] = {10, 8, 5, 8, 7}; //enlista las calificaciones de los 5 alumnos
11
12     int indice = 0;
13     printf("\nlista\n");
14     while(indice < 5){
15         printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);
16         //enlista cada alumno con la calificación que le corresponde, se anota el acento,
17         //como indice comienza en 0, se le suma un uno para comenzar
18         //por el alumno uno e ir sumando ul cada vez hasta llegar a 5
19         indice += 1; //análogo a indice = indice +1;
20     }
21
22     printf("\n");
23     return 0;
24 }
25
26
```

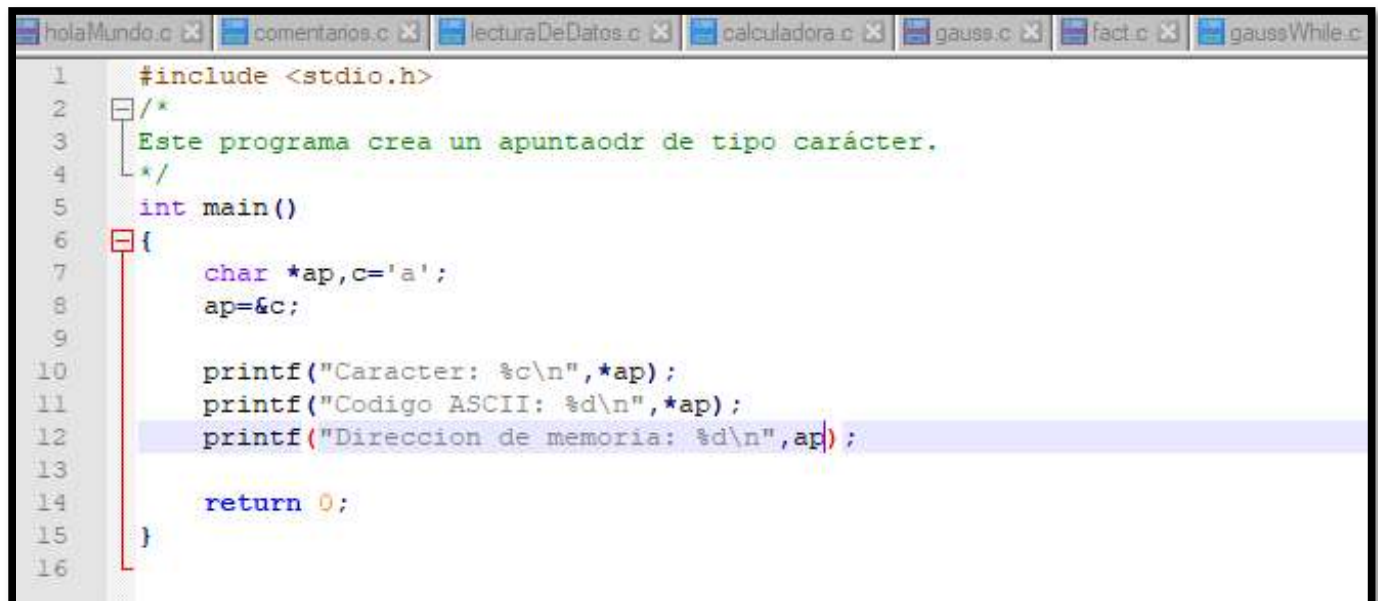
```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc unidimensional.c -o unidimensional.exe
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>unidimensional.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\aidee\Desktop\Lenguaje C\Ejemplos>_
```

## Actividad 2.-



```
1  #include <stdio.h>
2  /*
3   Este programa crea un apuntaodr de tipo carácter.
4   */
5   int main()
6   {
7       char *ap, c='a';
8       ap=&c;
9
10      printf("Caracter: %c\n", *ap);
11      printf("Codigo ASCII: %d\n", *ap);
12      printf("Direccion de memoria: %d\n", ap);
13
14      return 0;
15  }
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc apuntador.c -o apuntador.exe
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>apuntador.exe
```

```
Caracter: a
```

```
Codigo ASCII: 97
```

```
Direccion de memoria: 6422299
```

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

### Actividad 3.-

```
comentarios.c | lecturaDeDatos.c | calculadora.c | gauss.c | fact.c | gaussWhile.c | menuSalir.c | factorial.c | submei
1  #include<stdio.h>
2  /*
3  Este programa accede a las localidades de memoria de distintas variables
4  a través de un apuntador.
5  */
6  int main(){
7      int a=5, b=10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
8      int *apEnt;
9      apEnt=&a;
10
11      printf("a=5, b=10, c[10]={5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
12      printf("apEnt=&a\n");
13
14      b=*apEnt;
15      printf("b=*apEnt \t-> b=%i\n",b);
16
17      b=*apEnt+1;
18      printf("b=*apEnt+1 \t-> b=%i\n",b);
19
20      *apEnt=0 ;
21      printf("*apEnt=0 \t-> a=%i\n",a);
22
23      apEnt=&c[0];
24      printf("apEnt=&c[0] \t-> apEnt=%i\n",*apEnt);
25
26      return 0;
27 }
```

#### C:\. Símbolo del sistema

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc apunadores.c -o apunadores.exe
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>apunadores.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\aidee\Desktop\Lenguaje C\Ejemplos>_
```

#### Actividad 4.-

```
1  #include<stdio.h>
2  /*
3   Este programa trabaja con aritmética de apuntadores para acceder a los valores de un arreglo.
4   */
5
6  int main()
7  {
8      int arr[]={5, 4, 3, 2, 1};
9      int *apArr;
10     apArr=arr;
11
12     printf("int arr[]={5,4,3,2,1};\n");
13     printf("apArr=&arr[0]\n");
14
15     int x=*apArr;
16     printf("x=*apArr \t-> x=%d\n",x);
17
18     x=*(apArr+1);
19     printf("x=*(apArr+1) \t-> x=%d\n",x);
20
21     x=*(apArr+2);
22     printf("x=*(apArr+1) \t-> x=%d\n", x);
23
24     return 0;
25 }
```

C:\. Símbolo del sistema

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc apuntadorarit.c -o apuntadorarit.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>apuntadorarit.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\aidee\Desktop\Lenguaje C\Ejemplos>\_

## Actividad 5.-

```
1  #include<stdio.h>
2  /*
3   Este programa genera un arreglo unidimensional de 5 elemntos y
4   accede a cada elemento del arreglo a través de un apuntador
5   utilizando un ciclo for.
6  */
7
8  int main()
9  {
10     #define TAMANO 5
11     int lista[TAMANO] = {10, 8, 5, 8, 7};
12     int *ap=lista;
13     printf("\tLista\n");
14     for(int indice=0; indice<5; indice++){
15         printf("\nCalificacion del alumno %d es %d", indice+1, *(ap+indice));
16     }
17
18     printf("\n");
19
20     return 0;
21 }
```

```
ca. Símbolo del sistema

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc ciclofor.c -o ciclofor.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>ciclofor.exe
Lista

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

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



## Actividad 6.-

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

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc apcadena.c -o apcadena.exe
```

```
C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>apcadena.exe
```

```
Ingrese una palabra:luna
```

```
La palabra ingresada es: luna
```

```
l
```

```
u
```

```
n
```

```
a
```

```
↓
```

```
@
```

```
p
```

```
↓
```

```
@
```



## Actividad 7.-

```
gene.c x unidimensional.c x apuntador.c x apuntadores.c x apuntadorant.c x ciclofor.c x apcadena.c x amultidi
1  # include<stdio.h>
2  /*
3  Este programa genera un arreglo de dos dimensiones (arreglo multidimensional)
4  y accede a sus elementos a través de dos ciclos for, uno anidado dentro
5  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("Imprimir Matriz\n");
12     for(i=0; i<3; i++){
13         for(j=0; j<3; j++){
14             printf("%d,", matriz[i][j]);
15         }
16         printf("\n");
17     }
18     return 0;
19 }
```

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>gcc armultidi.c -o armultidi.exe

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>armultidi.exe

Imprimir Matriz

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

C:\Users\aidee\Desktop\Lenguaje C\Ejemplos>

## Actividad 8.-

```
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 un apuntador utilizando
5   un ciclo for.
6   */
7  int main(){
8
9      int matriz[3][3]={{1,2,3},{4,5,6},{7,8,9}};
10     int i, cont=0, *ap;
11     ap = matriz;
12     printf("Imprimir Matriz\n");
13     for(i=0; i<9; i++){
14         if(cont==3){
15             printf("\n");
16             cont=0;
17         }
18         printf("%d\t",*(ap+i));
19         cont++;
20     }
21     printf("\n");
22
23     return 0;
24 }
```

## **Conclusiones**

Me costó un poco entender y realizar esta práctica, tuve que investigar en varias páginas para entender un poco mejor los arreglos. Era algo que no conocía y ahora ya que lo sé puedo visualizar en qué puedo llegar a utilizar los arreglos, como en alguna calculadora de matrices o algo por estilo.

No me encantó la práctica fue un poco difícil y uno de los ejercicios, el último no me compila me marca error y o pude encontrar cómo corregirlo.

## **Bibliografía**

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- Manual de prácticas de laboratorio.
- Aplicación Notepad