



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Marco Antonio Martínez Quintana

*Profesor:*

Fundamentos de programación

*Asignatura:*

3

*Grupo:*

11

*No de Práctica(s):*

Zurita León Dana Cecilia

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

No aplica

*No. de Lista o Brigada:*

Primero

*Semestre:*

04 de Enero de 2021

*Fecha de entrega:*

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **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.

### **Introducción:**

Los arreglos son conjuntos de datos del mismo tipo, a estos valores se les asigna una posición determinada, iniciando desde el número 0, tienen un tamaño fijo, este se determina como en la parte de asignaciones.

Los arreglos son de tipo multidimensional o unidimensional.

Los apuntadores son variables que contiene la dirección de una variable, esto se refiere a la localidad de memoria de otra variable, ya que trabaja directamente con la memoria, se obtienen datos de manera rápida.

Solo pueden acceder a datos con el mismo tipo de datos con el que fueron declarados.

## Actividad:

The screenshot shows a code editor with a C program. The code defines an array of 5 elements and prints each element using a while loop. The output window shows the execution of the program and its output.

```
#include <stdio.h>
/*
Este programa genera un arreglo unidimensional de 5 elementos y los
accede a cada elemento del arreglo a través de un ciclo while.
*/
int main (){
    //Define el tamaño del arreglo
    #define TAMANO 5
    //Determinación de variables
    int lista[TAMANO] = {10, 8, 5, 8, 7};
    char ao= 162;
    int indice = 0;

    //Imprime el nombre el programa
    printf("\tLista\n");
    //Se realiza el procedimiento siempre y cuando se cumpla la condición
    while (indice < 5 ){
        //Código a ejecutar
        printf("\nCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
        indice += 1; // análogo a indice = indice + 1;
    }

    printf("\n");
    return 0;
}
```

Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac11.c -o prac11.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac11.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\dellR\Desktop\LenguajeC\Ejemplos>
```

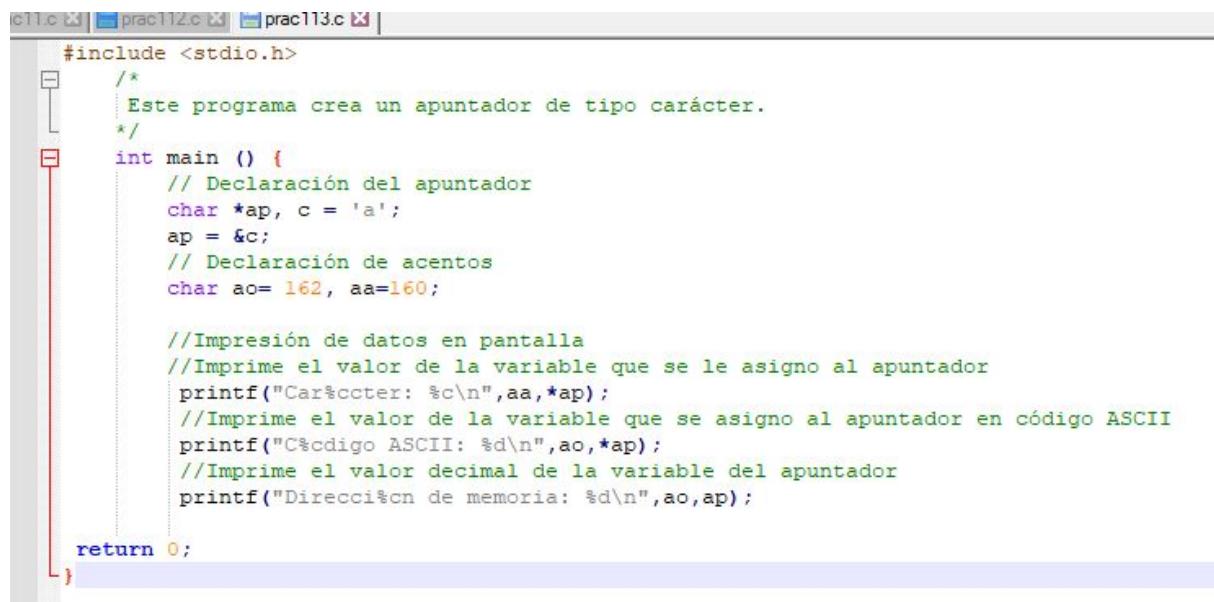
```
lagausswhie.c submenu.c ifelsecalcu.c menusalir.c wcalculadora.c p10.c pp10.c dep1.c prac11.c prac112.c
1 #include <stdio.h>
2 /*
3 | Este programa genera un arreglo unidimensional de 5 elementos y
4 | accede a cada elemento del arreglo a través de un ciclo for.
5 */
6 int main () {
7     //Tamaño del arreglo
8     #define TAMANO 5
9     //Declaracion de variables
10    int lista[TAMANO] = {10, 8, 5, 8, 7};
11    char ao= 162;
12    //Impresion de datos
13    printf("\tlista\n");
14    for (int indice = 0 ; indice < 5 ; indice++){
15        printf("\nCalificación del alumno %d es %d",ao, indice+1, lista[indice]);
16    }
17    //Salto de linea
18    printf("\n");
19
20    return 0;
21
22 }
```

Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac112.c -o prac112.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac112.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\dellR\Desktop\LenguajeC\Ejemplos>
```



```
#include <stdio.h>
/*
Este programa crea un apuntador de tipo carácter.
*/
int main () {
    // Declaración del apuntador
    char *ap, c = 'a';
    ap = &c;
    // Declaración de acentos
    char ao= 162, aa=160;

    //Impresión de datos en pantalla
    //Imprime el valor de la variable que se le asigno al apuntador
    printf("Carácter: %c\n",aa,*ap);
    //Imprime el valor de la variable que se asigno al apuntador en código ASCII
    printf("Código ASCII: %d\n",ao,*ap);
    //Imprime el valor decimal de la variable del apuntador
    printf("Dirección de memoria: %d\n",ao,ap);

    return 0;
}
```

Este Estándar para el Intercambio de Información )

Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac113.c -o prac113.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac113.exe
Carácter: a
Código ASCII: 97
Dirección de memoria: 6422295
```

```
prac11.c | prac112.c | prac113.c | prac114.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 int main () {  
7 //Declaración de variables  
8 int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};  
9 //Apuntador  
0 int *apEnt;  
1 apEnt = &a;  
2 //Impresión en pantalla  
3 //Indica los valores del arreglo  
4 printf("a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");  
5 printf("apEnt = &a\n");  
6  
7 //Asigna que b es igual al valor asignado al apuntador  
8 b = *apEnt;  
9 printf("b = *apEnt \t-> b = %i\n", b);  
0  
1 //Asigna que b es igual al valor asignado al apuntador más uno  
2 b = *apEnt +1;  
3 printf("b = *apEnt + 1 \t-> b = %i\n", b);  
4  
5 //Asigna que ahora el apuntador es igual a 0  
6 *apEnt = 0;  
7 printf("*apEnt = 0 \t-> a = %i\n", a);  
8  
9 //Asigna que apEnt ahora es igual al valor del arreglo en la posición 0  
0 apEnt = &c[0];  
1 printf("apEnt = &c[0] \t-> apEnt = %i\n", *apEnt);  
2  
3 return 0;  
4 }|
```

Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac114.c -o prac114.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac114.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\dellR\Desktop\LenguajeC\Ejemplos>
```

```
#include <stdio.h>
/*
Este programa trabaja con aritmética de apuntadores para acceder a los
valores de un arreglo.
*/
int main () {
    //Declaración del arreglo
    int arr[] = {5, 4, 3, 2, 1};
    int *apArr;
    apArr = arr;

    //Impresión en pantalla
    //Imprime valores del arreglo
    printf("int arr[] = {5, 4, 3, 2, 1};\n");
    printf("apArr = &arr[0]\n");

    //La variable x es igual al apuntador
    int x = *apArr;
    //Imprime primer valor del arreglo
    printf("x = *apArr \t -> x = %d\n", x);

    //El apuntador toma el valor del segundo valor del arreglo ya que se le suma 1
    x = *(apArr+1);
    printf("x = *(apArr+1) \t -> x = %d\n", x);

    //El apuntador toma el tercer valor del arreglo
    x = *(apArr+2);
    printf("x = *(apArr+2) \t -> x = %d\n", x);

    return 0;
}
```

```
107 Símbolo del sistema  
1 C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac115.c -o prac115.exe  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac115.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\dellR\Desktop\LenguajeC\Ejemplos>
```

```
#include <stdio.h>  
/*  
Este programa genera un arreglo unidimensional de 5 elementos y  
accede a cada elemento del arreglo a travéz de un apuntador  
utilizando un ciclo for.  
*/  
int main (){  
    //Declaración de variables  
    char ac= 162;  
    //Tamaño del arreglo  
    #define TAMANO 5  
    //Valores del arreglo  
    int lista[TAMANO] = {10, 8, 5, 8, 7};  
    //Asignación del apuntador  
    int *ap = lista;  
    //Imprime título del programa  
    printf("\tLista\n");  
    //Ciclo que asigna a cada alumno su calificación segun el valor del arreglo correspondiente  
    for (int indice = 0 ; indice < 5 ; indice++){  
        //A cada apuntador se le suma uno  
        printf("\nCalificación del alumno %d es %d",ac, indice+1, *(ap+indice));  
    }  
    //Salto de linea  
    printf("\n");  
  
    return 0;  
}
```

```
Símbolo del sistema  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac116.c -o prac116.exe  
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac116.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\dellR\Desktop\LenguajeC\Ejemplos>
```

```
#include <stdio.h>
/*
Este programa muestra el manejo de cadenas en lenguaje C.
*/
int main(){
    //Declaración de variables
    char palabra[20];
    int i=0;
    //Solicitud de datos
    printf("Ingrese una palabra: ");
    //Guarda la palabra escrita en %s
    scanf("%s", palabra);
    //Muestra la palabra con salto de linea en cada letra
    printf("La palabra ingresada es: %s\n", palabra);
    for (i = 0 ; i < 20 ; i++){
        printf("%c\n", palabra[i]);
    }
    return 0;
}
```

```
T Símbolo del sistema

C:\Users\dellR\Desktop\LenguajeC\Ejemplos>>gcc prac117.c -o prac117.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac117.exe
Ingrese una palabra: Ceci
La palabra ingresada es: Ceci
C
e
c
i
↓
@
P
↓
@

@
,
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

```
#include<stdio.h>
/*
 * Este programa genera un arreglo de dos dimensiones (arreglo
 * multidimensional) y accede a sus elementos a través de dos ciclos
 * for, uno anidado dentro de otro.
 */
int main(){
    //Declaración de una matriz
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    //Declaración de variables
    int i, j;
    printf("Imprimir Matriz\n");
    //Imprime valores de la matriz
    for (i=0 ; i<3 ; i++){
        for (j=0 ; j<3 ; j++){
            printf("%d, ",matriz[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

norm Símbolo del sistema

```
1 C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac118.c -o prac118.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac118.exe
Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>
```

The image shows a Windows desktop environment. At the top, there is a taskbar with several icons. Below the taskbar, a code editor window is open, displaying a C program named 'prac119.c'. The code defines a 3x3 matrix and prints its elements row by row. A terminal window titled 'Símbolo del sistema' (Command Prompt) is also open, showing the command 'gcc prac119.c -o prac119.exe' being run, followed by the output of the program which prints the matrix.

```
#include<stdio.h>
/* Este programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos a travéz de un apuntador utilizando un ciclo for.
 */
int main(){
    //Declaración de la matriz y sus valores
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = matriz;
    //Impresión de datos en pantalla
    printf("Imprimir Matriz\n");
    //Se establece un contador, va desde cero hasta un número menor a nueve, agregando uno
    for (i=0 ; i<9 ; i++){
        //Cada tres valores se establece un salto de linea
        if (cont == 3){
            printf("\n");
            cont = 0;
        }
        printf("%d\t",*(ap+i));
        cont++;
    }
    printf("\n");
    return 0;
}
```

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc prac119.c -o prac119.exe
prac119.c: In function 'main':
prac119.c:9:7: warning: assignment to 'int *' from incompatible pointer type 'int (*)[3]' [-Wincompatible-pointer-types]
      9 |     ap = matriz;
           ^

C:\Users\dellR\Desktop\LenguajeC\Ejemplos>prac119.exe
Imprimir Matriz
1     2     3
4     5     6
7     8     9
```

```
1 #include<stdio.h>
2 int main ()
3 {
4     int ab[3][3];
5
6     printf("\n Abarrotes \n1) Dulceria\n2) Salchichoneria\n3) Lacteos\n");
7     printf("\n1) Dulces \n1.- Sandibrochas\n2.- Chocolates\n3.- Frituras V\n\n");
8     printf("\n2) Salchichoneria \n1.- Jamon\n2.- Queso\n3.- Salchichas\n\n");
9     printf("\n3) Lacteos \n1.- Yogur\n2.- Leche\n3.- Lechera\n\n");
10
11    for(int i=0; i<3; i++)
12    {
13        for(int j=0; j<3; j++)
14        {
15            printf("Numero de productos que desea %d de la secci%cn %d \t", j+1, 162, i+1);
16            scanf("%d", &ab[i][j]);
17        }
18    }
19 /*printf("\nProductos totales\n",163);
20 for(int i=0; i<3; i++)
21 {
22     for(int j=0; j<3; j++)
23     {
24         printf("%d\t",ab[i][j]);
25     }
26     printf("\n");
27 }
28 printf("\n");*/
29
30     int p[3];
31     int a=0;
32     for(int i=0; i<3; i++)
33     {
34         a=0;
35         for(int j=0; j<3; j++)
36         {
37             a=a+ab[i][j];
38         }
39         p[i]=a;
40         printf("\nEl n%cmero total de productos del apartado %d son: %d", 163, i+1, p[i]);
41     }
42     a=0;
43     for(int i=0; i<3; i++)
44     {
45         a=a+p[i];
46     }
47     printf("\n\nEl n%cmero total de productos que desea son: %d\n", 163, a);
48
49
50 }
```

### Símbolo del sistema

```
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>gcc p11.c -o p11.exe
C:\Users\dellR\Desktop\LenguajeC\Ejemplos>p11.exe

Abarrotes
1) Dulceria
2) Salchichoneria
3) Lacteos

1) Dulces
1.- Sandibrochas
2.- Chocolates
3.- Frituras V

2) Salchichoneria
1.- Jamon
2.- Queso
3.- Salchichas

3) Lacteos
1.- Yogur
2.- Leche
3.- Lechera

Número de productos que desea 1 de la sección 1      1
Número de productos que desea 2 de la sección 1      0
Número de productos que desea 3 de la sección 1      12
Número de productos que desea 1 de la sección 2      67
Número de productos que desea 2 de la sección 2      4
Número de productos que desea 3 de la sección 2      10
Número de productos que desea 1 de la sección 3      8
Número de productos que desea 2 de la sección 3      5
Número de productos que desea 3 de la sección 3      32

El número total de productos del apartado 1 son: 13
El número total de productos del apartado 2 son: 81
El número total de productos del apartado 3 son: 45

El número total de productos que desea son: 139
```

### Conclusión:

En esta práctica pude aprender la importancia de los arreglos, estos hacen el código eficiente, también pude identificar como se determina un arreglo unidimensional y multidimensional.

No obstante también pude aprender como utilizar un apuntador, su sintaxis y la utilidad que tienen.

Fue una práctica con mucha retroalimentación.

### Bibliografía

Guía práctica de estudio 11: Arreglos unidimensionales y multidimensionales  
[PDF] Universidad Nacional Autónoma de México. Consultado en enero de  
2021 del sitio: file:///C:/Users/dellR/Downloads/fp\_p11.pdf