



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Karina Morales Garcia

*Asignatura:* Fundamentos de programacion

*Grupo:* 20

*No. de práctica(s):* 10

*Integrante(s):* Avila Pineda Samuel David

*No. de lista o brigada:* 06

*Semestre:* 2023-1

*Fecha de entrega:* 06 de diciembre del 2022

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo:

El alumno utilizará arreglos de dos dimensiones en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, en estructuras que utilicen dos índices.

## Actividades:

- Resolver problemas que requieran el uso de un arreglo de dos dimensiones, a través de programas en lenguaje C.
- Manipular este tipo de arreglos a través de índices.

## Introducción

### ¿Qué es un arreglo de dos dimensiones?

Es un conjunto de datos contiguos del mismo tipo y con tamaño fijo definido. Se requiere el uso de índices para acceder a un elemento.

### ¿Para qué se utilizan?

Para hacer más eficiente el código de un programa.

### Arreglos multidimensionales.

La 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, mientras el “tipo de dato” es el tipo de dato de elementos del arreglo, estos como sabemos poder ser de tipo real, entero, carácter o estructura.

## Programa 1a.c

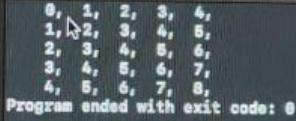
```
9 int main() {
10 int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
11 int i, j;
12 printf("Imprimir Matriz\n");
13 for (i=0 ; i<3 ; i++) //Representa al renglón del arreglo
14 {
15     for (j=0 ; j<3 ; j++)//Representa a la columna del arreglo
16     {
17         printf("%d, ",matriz[i][j]); }
18     printf("\n");
19 }
20 return 0;
21 }
```

Imprimir Matriz  
1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

El primer programa que se ejecuto es el que se muestra en la parte de arriba, esta muestra un programa que genera un arreglo de dos dimensiones y accede a cada uno de sus elementos, a través de dos ciclos for.

## Programa 2a.c

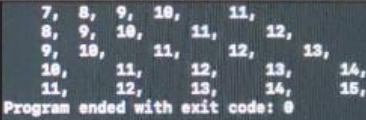
```
8 #include<stdio.h>
9 int main() {
10 int i,j,a[5][5];
11 for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
12 {
13 for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
14 {
15 a[i][j]=i+j;
16 printf("\t%d, ",a[i][j]);
17 }
18 printf("\n");
19 }
20 return 0;
21 }
```



```
0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,
Program ended with exit code: 0
```

Ese fue el segundo programa en él se genera un arreglo de dos dimensiones y accedes a sus elementos por la posición que indica el renglón y la columna, esto a través de dos ciclos for, el contenido es la suma de sus índices.

```
9 #include<stdio.h>
10 int main() {
11 int i,j,a[5][5];
12 for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
13 {
14 for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
15 {
16 a[i][j]=i+j+7;
17 printf("\t%d, ",a[i][j]);
18 }
19 printf("\n");
20 }
21 return 0;
22 }
23
```



```
7, 8, 9, 10, 11,
8, 9, 10, 11, 12,
9, 10, 11, 12, 13,
10, 11, 12, 13, 14,
11, 12, 13, 14, 15,
Program ended with exit code: 0
```

De este mismo programa solo agregamos un numero en la fila 16, esto con el fin de ver qué pasaba y pues es lo mismo, la suma de sus índices, solo que ahora empezaba a partir del 7, en mi caso.

## Programa 1b.c

```
9  #include<stdio.h>
10 int main()
11 {
12     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, j;
13     printf("Imprimir Matriz\n");
14     i=0;
15     while(i<3) //Representa al renglón del arreglo
16     {
17         j=0;
18         while (j<3)//Representa a la columna del arreglo
19         {
20             printf("%d, ",matriz[i][j]);
21             j++;
22         }
23         printf("\n");
24         i++;
25     }
26     return 0;
27 }
28
```

Imprimir Matriz  
1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

En este programa que se muestra arriba lo único que hicimos fue copiar y pegar, no hicimos ninguna modificación.

```
9  #include<stdio.h>
10 int main()
11 {
12     int matriz[2][2] = {{1,2},{4,5}}; int i, j;
13     printf("Imprimir Matriz\n");
14     i=0;
15     while(i<2) //Representa al renglón del arreglo
16     {
17         j=0;
18         while (j<2) //Representa a la columna del arreglo
19         {
20             printf("%d, ",matriz[i][j]);
21             j++;
22         }
23         printf("\n");
24         i++;
25     }
26     return 0;
27 }
```

Imprimir Matriz  
1, 2,  
4, 5,  
Program ended with exit code: 0

La profesora nos pidió que cambiáramos el tipo de matriz, yo lo hice 2x2, pero también teníamos que cambiar while, esto para que no nos marcara error.

### Programa 2b.c

```
9  #include<stdio.h>
10 int main() {
11     int i,j,a[5][5];
12     i=0;
13     while (i<5) //Representa al renglón del arreglo
14     {
15         j=0;
16         while (j<5) //Representa a la columna del arreglo
17         {
18             a[i][j]=i+j;
19             printf("\t%d, ",a[i][j]); j++;
20         } printf("\n"); i++;
21     }
22     return 0; }
23
```

0, 1, 2, 3, 4,  
1, 2, 3, 4, 5,  
2, 3, 4, 5, 6,  
3, 4, 5, 6, 7,  
4, 5, 6, 7, 8,  
Program ended with exit code: 0

### CUADRO COMPARATIVO

For	while	Do - while
<pre> { for() { for() { //Acciones } //Acciones } Return 0; } </pre>	<pre> { //Declaramos variables While() { //Asignamos valor While() { //Acciones } //Acciones } Return 0; } </pre>	<pre> { //Declaramos variables Do { //asignamos valor Do { //Acciones } While() //acciones } While(); Return 0; } </pre>

### Programa 1c.c

```

9  #include<stdio.h>
10 int main() {
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, j;
13     printf("Imprimir Matriz\n");
14     i=0;
15     do //Representa al renglón del arreglo
16     {
17         j=0;
18         do //Representa a la columna del arreglo
19         {
20             printf("%d, ",matriz[i][j]);
21             j++; }
22         while (j<3); printf("\n");
23         i++; }
24     while(i<3);
25     return 0; }
26

```

Imprimir Matriz  
1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

### Programa 2c.c



```

9  #include<stdio.h>
10 int main() {
11     int i,j,a[5][5];
12     i=0;
13     do //Representa al renglón del arreglo
14     {
15         j=0;
16         do //Representa a la columna del arreglo
17         {
18             a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++;
19         }
20         while (j<5);
21         printf("\n");
22         i++;
23     }
24     while (i<5);
25     return 0;
26 }
27

```

```

0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,

```

Program ended with exit code: 0

### Programa 3.c

```

10 // Ejercicio 4a.c
11 int main ()
12 {
13     int lista[3][3]; // Se declara el arreglo multidimensional
14     int i,j;
15     int renglon,columna;
16     printf("\nDe el número de renglones y columnas separados con coma\n");
17     scanf("%d,%d",&renglon,&columna);
18     if(((renglon>=1) && (renglon<=10))&&((columna>=1) && (columna<=10)))
19     {
20         // Acceso a cada elemento del arreglo multidimensional usando for
21         for (i= 0 ; i <= renglon-1 ; i++)
22         {
23             for(j= 0 ; j <= columna-1 ; j++)
24             {
25                 printf("\nNúmero para el elemento %d,%d del arreglo", i,j );
26                 scanf("%d",&lista[i][j]);
27             }
28         }
29         printf("\nLos valores dados son: \n");
30         // Acceso a cada elemento del arreglo multidimensional usando for
31         for (i= 0 ; i <= renglon-1 ; i++) {
32             for(j= 0 ; j <= columna-1 ; j++) {
33                 printf("%d ", lista[i][j]);
34             }
35             printf("\n");
36         }
37     }
38     else printf("Los valores dados no es válido"); printf("\n");
39     return 0;
40 }
41

```

De el número de renglones y columnas separados con coma  
3,2

Número para el elemento 0,0 del arreglo4

Número para el elemento 0,1 del arreglo3

Número para el elemento 1,0 del arreglo1

Número para el elemento 1,1 del arreglo6

Número para el elemento 2,0 del arreglo4

Número para el elemento 2,1 del arreglo2

Los valores dados son:

4 3  
1 6  
4 2

Program ended with exit code: 0

All Output

## Programa 4a.c



```
9  #include<stdio.h>
10 int main() {
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, cont=0, *ap;
13     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
14     printf("Imprimir Matriz\n");
15     for (i=0 ; i<9 ; i++)
16     {
17         if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
18         {
19             printf("\n");
20             cont = 0; //Inicia conteo de elementos del siguiente renglón
21         }
22         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
23         cont++;
24     } printf("\n"); return 0;
25 }
26
```

Imprimir Matriz

1	2	3
4	5	6
7	8	9

Program ended with exit code: 0

### Programa 4b.c

```
9  #include<stdio.h>
10 int main() {
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, cont=0, *ap;
13     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0]; printf("Imprimir Matriz\n");
14     i=0;
15     while (i<9)
16     {
17         if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
18         {
19             printf("\n");
20             cont = 0; //Inicia conteo de elementos del siguiente renglón
21         }
22         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
23         cont++;
24         i++; }
25     printf("\n");
26     return 0; }
27
```

1	2	3
4	5	6
7	8	9

Program ended with exit code: 0

La imagen mostrada anteriormente es el programa 4b.c

```

9  #include<stdio.h>
10 int main() {
11     int matriz[4][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
12     int i, cont=0, *ap;
13     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0]; printf("Imprimir Matriz\n");
14     i=0;
15     while (i<16)
16     {
17         if (cont == 4) //Se imprimió un renglón y se hace un salto de línea
18         {
19             printf("\n");
20             cont = 0; //Inicia conteo de elementos del siguiente renglón
21         }
22         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
23         cont++;
24         i++; }
25     printf("\n");
26     return 0; }
27

```

```

1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
Program ended with exit code: 0

```

Solo modifique a un arreglo 4x4

### Programa 4c.c

```

9  #include<stdio.h>
10 int main() {
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, cont=0, *ap;
13     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
14     printf("Imprimir Matriz\n");
15     i=0;
16     do
17     {
18         if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
19         {
20             printf("\n");
21             cont = 0; //Inicia conteo de elementos del siguiente renglón
22         }
23         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
24         cont++;
25         i++;
26     }
27     while (i<9);
28     printf("\n");
29     return 0; }
30

```

```

Imprimir Matriz
1  2  3
4  5  6
7  8  9
Program ended with exit code: 0

```

Ese es el programa original

```

9  #include<stdio.h>
10 int main() {
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
12     int i, cont=0, *ap;
13     ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
14     printf("Imprimir Matriz\n");
15     i=0;
16     do //inicio del hacer mientras
17     {
18         if (cont == 3) //Nos ayuda a hacer el salto de línea una vez que en la fila hay 3 valores
19         {
20             printf("\n");
21             cont = 0; //Inicia conteo de elementos del siguiente renglón
22         }
23         printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
24         cont++;
25         i++;
26     }
27     while (i<9); //fin del hacer mientras
28     printf("\n");
29     return 0; }
30

```

Imprimir Matriz

```

1 2 3
4 5 6
7 8 9
Program ended with exit code: 0

```

Aquí solo comentamos el do while, donde inicia y en donde termina.

### Cuadro Comparativo

For	while	Do - while
<pre> { //Declaramos variables y utilizamos el apuntador For() { If() { //Acciones } //Acciones } //Acción Return 0; } </pre>	<pre> { //Declaramos variables y apuntador //Acciones While() { If() { //Acciones } //Acciones } //Acción Return 0; } </pre>	<pre> { //Declaramos variables //Acciones Do { If() { //Acciones } //Acciones } While(); //Acción Return 0; } </pre>

### Tarea:

#### Ejercicio 1:

Realiza un programa que muestre tu nombre y número de cuenta haciendo uso de 2 arreglos, emplear while y for

```

9  #include <stdio.h>
10 int main()
11 {
12     int cuenta = 320278901;
13     char *nombres[30]="Samuel David Avila Pineda";
14     int i;
15     for (i=0; i<1; i++)
16     {
17         printf("Nombre: %s\n Cuenta: %d\n", *(nombres + i), (+cuenta + i));
18     }
19     i=0;
20     while (i < 1)
21     {
22         printf("Nombre: %s\n Cuenta: %d\n", *(nombres + i), (+cuenta + i));
23         i++;
24     }
25     return 0;
26 }
27

```

input

```

Nombre: Samuel David Avila Pineda
Cuenta: 320278901
Nombre: Samuel David Avila Pineda
Cuenta: 320278901

```

```

...Program finished with exit code 0
Press ENTER to exit console.

```

## Ejercicio 2:

Modifica el programa del ejercicio 1 utilizando apuntadores

```

9  #include <stdio.h>
10 int main()
11 {
12     int i, j, M[3][3]={{3,2,0},{2,7,8},{9,0,1}};
13     int *ap, cont=0;
14     char nombre [1][30]="Samuel David Avila Pineda";
15     ap = "M";
16     printf("\n Numero de cuenta \n");
17     i=0;
18     while(i<=8)
19     {
20         printf("%d\t", *(ap+i));
21         i++;
22     }
23     printf("\n");
24     printf("\nNombre:\n");
25     j=0;
26     while(j<=1)
27     {
28         printf("\n%s", &nombre[j]);
29         j++;
30     }
31     return 0;
32 }

```

input

```

main.c: In function 'main':
main.c:28:20: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)
[30]' [-Wformat=]
    28 |         printf("\n%s", &nombre[j]);
       |                ^~
       |                |
       |                | char (*)[30]
       |                char *

```

Numero de cuenta

```

3      2      0      2      7      8      9      0      1

```

Nombre:

```

Samuel David Avila Pineda

```

### Ejercicio 3:

Corrige e indica que realiza el siguiente programa:

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    int i,j, cont=0,n;
```

```
    float M[2][2], s=0, *ap;
```

```
    ap = M;
```

```
    for(i=0; i<=3; i++)
```

```
    {
```

```
        for(j=0; j<=1;j--)
```

```
        {
```

```
            printf("\t\n Teclear el elemento %d %d \t",
```

```
i+1,j+1);
```

```
            scanf("%f",&M[i][j]);
```

```
            s+=M[i][j];
```

```
        }
```

```

    }
    printf("\tLa Matriz es:\n");
    for (n=0 ; n<4 ; n++)
    {
        if (cont == 2)
        {
            printf("\n");
            cont = 1;
        }
        printf("%.2f\t\t",*(ap+n));
        cont++;
    }
    printf("\t\n\n La suma de los elementos es:%0.2f", s);
return 0;
}

```

```

9  #include <stdio.h>
10 int main()
11 {
12     int i, j, cont=0, n;
13     float M[2][2], s=0, *ap;
14     ap=M;
15     for(i=0; i<2; i++)
16     {
17         for(j=0; j<2; j++)
18         {
19             printf("\t\n Teclar el elemento %d %d \t", i+1, j+1);
20             scanf("%f",&M[i][j]);
21             s+=M[i][j];
22         }
23     }
24     printf("\tLa Matriz es:\n");
25     for(n=0; n<4; n++)
26     {
27         if (cont == 2)
28         {
29             printf("\n");
30             cont = 2;
31         }
32         printf("%.2f\t\t",*(ap+n));
33         cont ++;
34     }
35     printf("\t\n\n La suma de todos los elementos es:%.2f",s);
36     return 0;
37 }
38

```

input

```

Teclar el elemento 1 1      11
Teclar el elemento 1 2      2
Teclar el elemento 2 1      3
Teclar el elemento 2 2      7
    La Matriz es:
11.00      2.00
3.00      7.00
La suma de todos los elementos es:23.00

```



**Conclusiones:**

Esta práctica estuvo un poco más fácil, al mi parecer, poco a poco voy entendiendo mas esto de los arreglos, aun me sigue costando, pero creo que con el tiempo y practica me resultara más fácil. Ya pude diferenciar mas facil entre el ciclo for, while y el do while

**Referencias:**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.