



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Karina García Morales

*Asignatura:* Fundamentos de programación

*Grupo:* 22

*No. de práctica(s):* Práctica No.10

*Integrante(s):* Freddy Beckham Cedillo Arias

*No. de lista o brigada:* No.11

*Semestre:* 1er Semestre

*Fecha de entrega:* 22 de octubre del 2024

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# “Practica 10”

## Arreglos multidimensionales y bidimensionales

### 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.

### Desarrollo:

For	While	Do - while
<pre>for (i=0 ; i&lt;3 ; i++) //Representa al renglón del arreglo { for (j=0 ; j&lt;3 ; j++)//Representa a la columna del arreglo { printf("%d, ",matriz[i][j]); } printf("\n"); }</pre>	<pre>while(i&lt;3) //Representa al renglón del arreglo { j=0; while (j&lt;3)//Representa a la columna del arreglo { printf("%d, ",matriz[i][j]); j++; } printf("\n"); i++; }</pre>	<pre>do //Representa al renglón del arreglo { j=0; do //Representa a la columna del arreglo { printf("%d, ",matriz[i][j]); j++; } while (j&lt;3); printf("\n"); i++; } while(i&lt;3);</pre>

For	While	Do - while
<pre> for (i=0 ; i&lt;5 ; i++)//Representa al renglón del arreglo { for (j=0 ; j&lt;5 ; j++)//Representa a la columna del arreglo { a[i][j]=i+j; printf("\t%d, ",a[i][j]); } printf("\n"); } </pre>	<pre> while (i&lt;5) //Representa al renglón del arreglo { j=0; while (j&lt;5) //Representa a la columna del arreglo { a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++; } printf("\n"); i++; } </pre>	<pre> do //Representa al renglón del arreglo { j=0; do //Representa a la columna del arreglo { a[i][j]=i+j; printf("\t%d, ",a[i][j]); j++; } while (j&lt;5); printf("\n"); i++; } while (i&lt;5); </pre>

Programa 1

```

1 // main.c
2 // Programa 1
3 // Created by Cedillo Arias Freddy Beckham on 16/10/24
4
5 #include <stdio.h>
6
7 int main() {
8     int matriz[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
9     int i, j;
10
11     printf("Ingresa los datos de la matriz\n");
12     for (i = 0; i < 3; i++) { // Representa al renglón del arreglo
13         for (j = 0; j < 3; j++) { // Representa a la columna del arreglo
14             scanf("%d", &matriz[i][j]);
15         }
16         printf("\n");
17     }
18
19     printf("Imprimir Matriz\n");
20     for (i = 0; i < 3; i++) { // Representa al renglón del arreglo
21         for (j = 0; j < 3; j++) { // Representa a la columna del arreglo
22             printf("%d ", matriz[i][j]);
23         }
24         printf("\n");
25     }
26
27     return 0;
28 }
29 |

```

#### Salida del programa

```
Ingresa los datos de la matriz
```

```
Imprimir Matriz
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
[Execution complete with exit code 0]
```

## Programa 2

```

1 // main.c
2 // Programa 2
3 // Created by Cedillo Arias Freddy Beckham on 16/10/24
4
5 #include <stdio.h>
6
7 int main() {
8     int i, j, a[5][5];
9
10    for (i = 0; i < 5; i++) // Representa al renglón del arreglo
11    {
12        for (j = 0; j < 5; j++) // Representa a la columna del arreglo
13        {
14            a[i][j] = (i + 1) + (j + 1);
15            // Suma la cantidad de números indicada a i y j
16            printf("\t%d ", a[i][j]);
17        }
18        printf("\n");
19    }
20
21    return 0;
22 }
23

```

#### Salida del programa

2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9
6	7	8	9	10

[Execution complete with exit code 0]

## Programa 3

```
1 // main.c
2 // Programa 3
3 // Created by Cedillo Arias Freddy Beckham on 16/10/24
4
5 #include <stdio.h>
6
7 int main() {
8     int lista[10][10]; // Se declara el arreglo multidimensional
9     int i, j;
10    int renglón, columna;
11
12    printf("\nDa el número de renglones y columnas separados con coma\n");
13    scanf("%d,%d", &renglón, &columna); // Ingresar los valores
14
15    if ((renglón >= 1) && (renglón <= 10) && (columna >= 1) && (columna <= 10)) {
16        // Acceso a cada elemento del arreglo multidimensional usando for
17        for (i = 0; i <= renglón - 1; i++) {
18            for (j = 0; j <= columna - 1; j++) {
19                printf("\nNúmero para el elemento %d,%d del arreglo", i, j);
20                scanf("%d", &lista[i][j]); // Solicita los datos del arreglo
21            }
22        }
23
24        printf("\nLos valores dados son:\n");
25        // Acceso a cada elemento del arreglo multidimensional usando for
26        for (i = 0; i <= renglón - 1; i++) {
27            for (j = 0; j <= columna - 1; j++) {
28                printf("%d ", lista[i][j]); // Imprime los valores del arreglo
29            }
30            printf("\n");
31        }
32    } else {
33        printf("Los valores dados no es válido\n");
34        printf("\n");
35    }
36
37    return 0;
38 }
39 |
```

#### Salida del programa

Da el número de renglones y columnas separados con coma  
Los valores dados no es válido

[Execution complete with exit code 0]

## Programa 4

```
1 // main.c
2 // Programa 4
3 // Created by Cedillo Arias Freddy Beckham on 16/10/24
4
5 #include <stdio.h>
6
7 int main() {
8     int matriz[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
9     int i, cont = 0;
10    int *ap = &matriz[0][0]; // Esta sentencia es análoga a ap = &matriz[0][0];
11
12    printf("Imprimir Matriz\n");
13
14    // Usando apuntadores para recorrer la matriz
15    for (i = 0; i < 9; i++) {
16        printf("%d\t", *(ap + i));
17        // Se imprime el siguiente elemento de la matriz
18        cont++;
19
20        if (cont == 3) {
21            printf("\n"); // Se agrega un salto de línea al final de cada fila
22            cont = 0;     // Reinicia el contador para la siguiente fila
23        }
24    }
25
26    return 0;
27 }
28
```

### Salida del programa

```
Imprimir Matriz
1      2      3
4      5      6
7      8      9
```

[Execution complete with exit code 0]

## Tarea:

1.-Realiza un programa que muestre tu nombre y número de cuenta haciendo uso de 2 arreglos, emplear while y do-while

R=

```
8 *****
9 #include <stdio.h>
10
11 int main() {
12     // Arreglos para nombre y cuenta
13     char nombre[] = "Beckham "; // Coloca tu nombre aquí
14     int cuenta[] = {3, 2, 2, 1, 1, 6, 3, 6, 2}; // Número de cuenta
15
16     // Imprimir nombre usando un while
17     int i = 0;
18     printf("Nombre: ");
19     while (nombre[i] != '\0') {
20         printf("%c", nombre[i]);
21         i++;
22     }
23     printf("\n");
24
25     // Imprimir número de cuenta usando un do-while
26     i = 0;
27     printf("Número de cuenta: ");
28     do {
29         printf("%d", cuenta[i]);
30         i++;
31     } while (i < 8); // Se asume que el número de cuenta tiene 8 dígitos
32
33     printf("\n");
34
35     return 0;
36 }
```

input

Nombre: Beckham  
Número de cuenta: 32211636

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

2.-Modifica el programa del ejercicio (Programa3.c) para aplicar el uso de apuntadores(pagina 158)

R=

```

#include <stdio.h>

int main() {
    int lista[10][10]; // Se declara el arreglo multidimensional
    int *ptr;          // Declaración del apuntador
    int i, j;
    int renglon, columna;

    printf("\nDa el número de renglones y columnas separados por coma:\n");
    scanf("%d,%d", &renglon, &columna);

    // Verifica que los valores de renglón y columna estén en el rango válido
    if ((renglon >= 1 && renglon <= 10) && (columna >= 1 && columna <= 10)) {
        // Acceso a cada elemento del arreglo multidimensional usando for
        for (i = 0; i < renglon; i++) {
            for (j = 0; j < columna; j++) {
                printf("\nNúmero para el elemento %d,%d del arreglo: ", i, j);
                scanf("%d", &lista[i][j]);
            }
        }

        printf("\nLos valores dados son:\n");
        ptr = &lista[0][0]; // El apuntador apunta al primer elemento del arreglo

        // Acceso a cada elemento del arreglo multidimensional usando el apuntador
        for (i = 0; i < renglon; i++) {
            for (j = 0; j < columna; j++) {
                printf("%d ", *(ptr + i * columna + j));
                // Acceso mediante el apuntador
            }
            printf("\n");
        }
    } else {
        printf("Los valores dados no son válidos\n");
    }

    return 0;
}

```

input

```

Da el número de renglones y columnas separados por coma:
5
Los valores dados no son válidos

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

```

3.- Corrige e indica que realiza el siguiente programa 😞 Encuentra los 5 errores)

```
#include<studio.h>
```

```
main()
```

```
{
```

```
    int fila,columna, cont=0,contadorApuntador;
```

```
    float arreglo[3][3], acumulador=0, *ap;
```

```
    ap = &arreglo[0][0];
```

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

```
    {
```

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

```
        {
```

```
            printf("\t\n Teclear el elemento %d %d \t", fila+1,columna+1);
```

```
            scanf("%f",&arreglo[fila][columna]);
```

```
            acumulador+=arreglo[fila][columna];
```

```
        }
```

```
    }
```



```

printf("\tLa Matriz es:\n");
for (contadorApuntador=0 ; contadorApuntador<9 ; contadorApuntador++)
{
    if (cont == 3)
    {
        printf("\n");
        cont = 1;
    }
    printf("%.2f\t\t",*(ap+contadorApuntador));
    cont++;
}
printf("\t\n\n La suma de los elementos es:%0.2d", acumulador);
return 0;

}
R=

```

```

#include <stdio.h> // Corrección 1: Cambio de studio.h a stdio.h

int main() { // Corrección 5: Declarar el tipo de retorno de main
    int fila, columna, cont = 0, contadorApuntador;
    float arreglo[3][3], acumulador = 0, *ap;
    ap = &arreglo[0][0];

    // Llenar la matriz
    for (fila = 0; fila < 3; fila++) {
        for (columna = 0; columna < 3; columna++) {
            // Corrección 2: Cambiar columna-- a columna++
            printf("\t\nTeclear el elemento %d %d\t", fila + 1, columna + 1);
            scanf("%f", &arreglo[fila][columna]);
            acumulador += arreglo[fila][columna];
        }
    }

    printf("\tLa Matriz es:\n");
    for (contadorApuntador = 0; contadorApuntador < 9; contadorApuntador++) {
        if (cont == 3) {
            printf("\n");
            cont = 0; // Corrección 3: Restablecer cont a 0 en lugar de 1
        }
        printf("%.2f\t\t", *(ap + contadorApuntador));
        cont++;
    }

    // Imprimir la suma de los elementos
    printf("\t\n\nLa suma de los elementos es: %0.2f\n", acumulador);
    // Corrección 4: Cambiar %0.2d a %0.2f
    return 0; // Corrección 5: Añadir return 0 para indicar final correcto
}

```

```

Teclear el elemento 1 1 1
Teclear el elemento 1 2
2
Teclear el elemento 1 3 3
Teclear el elemento 2 1 4
Teclear el elemento 2 2 5
Teclear el elemento 2 3 6
Teclear el elemento 3 1 7
Teclear el elemento 3 2 8
Teclear el elemento 3 3 9
La Matriz es:
1.00      2.00      3.00
4.00      5.00      6.00
7.00      8.00      9.00

La suma de los elementos es: 45.00

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

```

4.- Programa que solicite al usuario los valores de dos matrices de 3 x 3 y haga su multiplicación haciendo uso de arreglos. Conforme se muestra a continuación:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mp} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{p1} & b_{p2} & \dots & b_{pn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1p}b_{p1}$$

R=

```

#include <stdio.h>

int main() {
    int A[3][3], B[3][3], C[3][3];
    int i, j, k;

    // Pedir al usuario los valores de la matriz A
    printf("Ingresa los elementos de la matriz A (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("A[%d][%d]: ", i + 1, j + 1);
            scanf("%d", &A[i][j]);
        }
    }

    // Pedir al usuario los valores de la matriz B
    printf("\nIngresa los elementos de la matriz B (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("B[%d][%d]: ", i + 1, j + 1);
            scanf("%d", &B[i][j]);
        }
    }

    // Multiplicación de matrices A * B, guardando el resultado en C
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            C[i][j] = 0; // Inicializar el valor de la posición [i][j]
            for (k = 0; k < 3; k++) {
                C[i][j] += A[i][k] * B[k][j]; // Suma acumulada de la multiplicación
            }
        }
    }

    // Imprimir la matriz resultado C
    printf("\nEl resultado de la multiplicación de A y B es la matriz C (3x3):\n");
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            printf("%d\t", C[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

```

Ingresa los elementos de la matriz A (3x3):
A[1][1]: 4
A[1][2]: 3
A[1][3]: 3
A[2][1]: 5
A[2][2]: 5
A[2][3]: 3
A[3][1]: 4
A[3][2]: 6
A[3][3]: 6

Ingresa los elementos de la matriz B (3x3):
B[1][1]: 3
B[1][2]: 5
B[1][3]: 6
B[2][1]: 7
B[2][2]: 3
B[2][3]: 2
B[3][1]: 3
B[3][2]: 2
B[3][3]: 1

El resultado de la multiplicación de A y B es la matriz C (3x3):
42      35      33
59      46      43
72      50      42

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

```

## **Conclusión:**

Los arreglos bidimensionales y multidimensionales nos permitirán almacenar datos en tablas o matrices, lo que nos facilita organizar la información de manera estructurada; como por ejemplo, un arreglo bidimensional es útil para representar una tabla de datos como una matriz, mientras que los arreglos de más dimensiones nos permiten trabajar con datos más complejos, como coordenadas en 3D; estos arreglos son útiles para acceder a elementos individuales y son esenciales para resolver problemas matemáticos o trabajar con datos en varias dimensiones, además serán parte esencial para ayudarnos en comprender una parte importante para la programación y nuestro futuro en ella.

## **Bibliografía:**

(Para esta práctica no se hizo uso de búsqueda en la web)