
	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	1/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 10: Arreglos multidimensionales



Elaborado por:	Actualizado por:	Revisado por:
M.C. Edgar E. García Cano Ing. Jorge A. Solano Gálvez	M.C. Cintia Quezada Reyes Ing. María Guadalupe Morales Nava	M.C. Laura Sandoval Montaño

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	2/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 10: Arreglos multidimensionales

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 programas en lenguaje C.
- Manipular este tipo de arreglos a través de índices.

Introducción


Un arreglo de dos dimensiones es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. Para acceder a un elemento en este tipo de arreglos se requiere el uso de dos índices.

Los arreglos se utilizan para hacer más eficiente el código de un programa, ya que la manipulación de datos del mismo tipo que son agrupados en un arreglo por tener un significado común, se realiza de una forma más clara y eficaz.

Licencia GPL de GNU

El software presente en esta práctica es libre bajo la licencia GPL de GNU, es decir, se puede modificar y distribuir mientras se mantenga la licencia GPL.

```
/*
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
```

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	3/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

```

* the Free Software Foundation, either version 3 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*
* Author: María Guadalupe Morales, Cintia Quezada and Jorge A. Solano
*
*/

```

Arreglos multidimensionales

Lenguaje C permite crear arreglos de varias dimensiones con la siguiente sintaxis:

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


Donde *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). El *tipoDeDato* es el tipo de dato de los elementos del arreglo, el cual puede ser tipo entero, real, carácter o estructura.

De manera práctica se puede considerar que 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 apuntadores.

Esta práctica estará enfocada exclusivamente al manejo de arreglos de dos dimensiones (bidimensionales).

Códigos (arreglos multidimensionales usando for)

A continuación, se observa un programa que genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a cada uno de sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *for*, uno anidado dentro de otro.

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	4/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Programa1a.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    for (i=0 ; i<3 ; i++) //Representa al renglón del arreglo
    {
        for (j=0 ; j<3 ; j++)//Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```


El siguiente programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *for*, uno anidado dentro de otro, el contenido de cada elemento de este arreglo es la suma de sus índices.

Programa2a. c

```
#include<stdio.h>

int main()
{
    int i,j,a[5][5];
    for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
    {
        for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
        {
            a[i][j]=i+j;
            printf("\t%d, ",a[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	5/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Códigos (arreglos multidimensionales usando while)

El código que se observa a continuación genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *while*, uno anidado dentro de otro.


Programa1b.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    i=0;
    while(i<3) //Representa al renglón del arreglo
    {
        j=0;
        while (j<3)//Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
            j++;
        }
        printf("\n");
        i++;
    }
    return 0;
}
```

Enseguida se muestra el código de un programa que permite generar un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *while*, uno anidado dentro de otro, el contenido de cada elemento de este arreglo es la suma de sus índices.

Programa2b.c

```
#include<stdio.h>
int main()
{
    int i,j,a[5][5];
    i=0;
    while (i<5) //Representa al renglón del arreglo
    {
        j=0;
        while (j<5) //Representa a la columna del arreglo
        {
```

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	6/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

```

        a[i][j]=i+j;
        printf("\t%d, ",a[i][j]);
        j++;
    }
    printf("\n");
    i++;
}
return 0;
}

```

Códigos (arreglos multidimensionales usando do-while)

A continuación, se presenta un código que permite la generación de un arreglo de dos dimensiones (arreglo multidimensional) y se puede acceder a cada uno de sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *do-while*, uno anidado dentro de otro.


Programa1c.c

```

#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, j;
    printf("Imprimir Matriz\n");
    i=0;
    do //Representa al renglón del arreglo
    {
        j=0;
        do //Representa a la columna del arreglo
        {
            printf("%d, ",matriz[i][j]);
            j++;
        }
        while (j<3);
        printf("\n");
        i++;
    }
    while(i<3);
    return 0;
}

```

Como puede observarse en el programa que se lee a continuación, se genera un arreglo de dos dimensiones (arreglo multidimensional) y se accede a todos sus elementos por la posición que indica el renglón y la columna a través de dos ciclos *do-while*, uno anidado


	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	7/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

dentro de otro, el contenido de cada uno de los elementos de este arreglo es la suma de sus índices.

Programa2c.c

```
#include<stdio.h>
int main()
{
    int i,j,a[5][5];
    i=0;
    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<5);
        printf("\n");
        i++;
    }
    while (i<5);
    return 0;
}
```


A continuación, se muestra un programa que genera un arreglo multidimensional de máximo 10 renglones y 10 columnas, para poder almacenar datos en cada elemento y posteriormente mostrar el contenido de esos elementos se hace uso de ciclos *for* anidados respectivamente.

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	8/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Programa3.c

```
#include <stdio.h>
int main ()
{
    int lista[10][10]; // Se declara el arreglo multidimensional
    int i,j;
    int renglon,columna;
    printf("\nDa el número de renglones y columnas separados con coma\n");
    scanf("%d,%d",&renglon,&columna);
    if(((renglon>=1) && (renglon<=10))&&((columna>=1) && (columna<=10)))
    {
        // Acceso a cada elemento del arreglo multidimensional usando for
        for (i= 0 ; i <= renglon-1 ; i++)
        {
            for(j= 0 ; j <= columna-1 ; j++)
            {
                printf("\nNúmero para el elemento %d,%d del arreglo", i,j );
                scanf("%d",&lista[i][j]);
            }
        }
        printf("\nLos valores dados son: \n");
        // Acceso a cada elemento del arreglo multidimensional usando for
        for (i= 0 ; i <= renglon-1 ; i++)
        {
            for(j= 0 ; j <= columna-1 ; j++)
            {
                printf("%d ", lista[i][j]);
            }
            printf("\n");
        }
    }
    else printf("Los valores dados no es válido");
    printf("\n");
    return 0;
}
```

El código anterior también puede realizarse utilizando los ciclos *while* y *do-while*, se invita al lector a que pruebe la implementación de ambos casos.

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	9/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Apuntadores y su relación con arreglos de dos dimensiones

Recordemos que un apuntador es una variable que contiene la dirección de una variable, es decir, 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 referirse a la dirección de memoria donde se ubica el valor de dicha variable (es lo que pasa cuando se lee un dato con scanf).

Los apuntadores 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 *.

Como se mencionó al inicio de esta práctica, en la memoria cada elemento del arreglo se guarda de forma contigua, por lo tanto, se puede recorrer un arreglo multidimensional con apuntadores.

Por ejemplo, los elementos de un arreglo de 3 renglones y 2 columnas de nombre *arr* se almacenan en memoria como lo muestra la Figura 1.

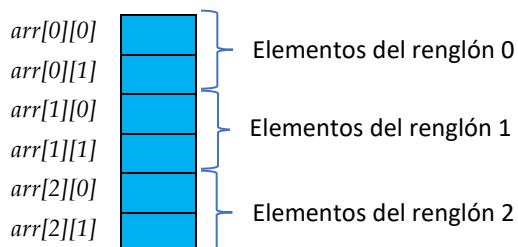



Figura 1. Almacenamiento en memoria de un arreglo de 3X2

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	10/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			


Como podemos observar en la Figura 1, los elementos de un arreglo bidimensional se almacenan por renglones; esto se debe considerar al momento de acceder a los elementos del arreglo usando apuntadores.

Códigos (arreglos multidimensionales con apuntadores)

El programa siguiente genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos a través de un apuntador utilizando un ciclo for.

Programa4a.c

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


	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	11/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

El código del siguiente programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos a través de un apuntador utilizando un ciclo *while*.

Programa4b.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = *matriz;//Esta sentencia es análoga a: ap = &matriz[0][0];
    printf("Imprimir Matriz\n");
    i=0;
    while (i<9)
    {
        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
        {
            printf("\n");
            cont = 0; //Inicia conteo de elementos del siguiente renglón
        }
        printf("%d\t",*(ap+i));//Se imprime el siguiente elemento de la matriz


        cont++;
        i++;
    }
    printf("\n");
    return 0;
}
```

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	12/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

La generación de un arreglo de dos dimensiones (arreglo multidimensional) y el acceso a sus elementos a través de un apuntador utilizando un ciclo *do-while* se puede observar en el programa siguiente:

Programa4c.c

```
#include<stdio.h>
int main()
{
    int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    int i, cont=0, *ap;
    ap = *matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];
    printf("Imprimir Matriz\n");
    i=0;
    do
    {
        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
        {
            printf("\n");
            cont = 0; //Inicia conteo de elementos del siguiente renglón
        }
        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
        cont++;
        i++;
    }
    while (i<9);
    printf("\n");
    return 0;
}
```

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	03
		Página	13/13
		Sección ISO	8.3
		Fecha de emisión	26 / agosto / 2021
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Bibliografía



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