



Guia #3: Arrays multidimensionales

Objetivo: Conocer el concepto, declaracion y uso de los arrays bidimensionales en el lenguaje C y comprender su utilidad.

Los arreglos pueden tener más de una dimensión. Cada dimensión puede tener diferente número de componentes. Un arreglo de dos dimensiones puede representarse como una tabla. Los arreglos de dos dimensiones se declaran de acuerdo al esquema:

tipo nombre [filas][columnas];

Por ejemplo el siguiente arreglo declara una tabla con cinco renglones y cuatro columnas. Los renglones varían de 0 a 4 y las columnas de 0 a 3.

int a[5][4];

Podemos representarlo como una tabla como se muestra.

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]
a[4][0]	a[4][1]	a[4][2]	a[4][3]

DECLARACIÓN O INICIALIZACIÓN.

Los arreglos de dos dimensiones se inicializan de forma similar a los de una dimensión.

Hay varias opciones en la notación que se muestran a continuación.

`int a[4][3] = {{3,6,1},{5,7,2},{2,1,5},{8,7,6}};`

`int a[][3] = {{3,6,1},{5,7,2},{2,1,5},{8,7,6}};`

`int a[4][3] = {3,6,1,5,7,2,2,1,5,8,7,6};`

Note que la opción

`int a[4][] = {{3,6,1},{5,7,2},{2,1,5},{8,7,6}};`

No es aceptada por el compilador ya que debe definirse el tamaño de todas las dimensiones excepto la primero.

ACCESO A LOS ELEMENTOS DE UN ARRAY BIDIMENSIONAL

Se puede acceder a los elementos de arrays bidimensionales de igual forma que a los elementos de un array unidimensional. La diferencia reside en que en los elementos bidimensionales deben especificarse los índices de la fila y la columna.

El formato general para asignación directa de valores a los elementos es:

<nombre array>[índice fila] [índice columnal]=valor elemento;

LECTURA Y ESCRITURA DE ELEMENTOS DE ARRAYS BIDIMENSIONALES

Las funciones de entrada o salida se aplican de igual forma a los elementos de un array bidimensional.

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

```
int main( ){
```

```
int numeros[2 ][2 ];
```

```
printf( "%d", numeros[0][0]);
```

```
scanf( "%d", %numeros[1][1]);
```

```
return 0;
```

```
}
```

RECORRER MATRICES

Para recorrer una matriz se anidan dos bucles for. En general para recorrer un array multidimensional se anidan tantas instrucciones for como dimensiones tenga el array.

ACCESO A LOS ELEMENTOS MEDIANTE BUCLES

Se puede acceder a los elementos de arrays bidimensionales mediante bucles anidados. Su sintaxis es:

```
int IndiceFila, IndiceCol;
```

```
for (IndiceFila = 0 ; IndiceFila < N u m F i l a s ; IndiceFila++)
```

```
for (IndiceCol = 0 ; IndiceCol < NumCol; IndiceCol++)
```

```
Procesar elemento[IndiceFila] [IndiceCol];
```

Ejemplo1. Declare un arreglo bidimensional de 4x4 y guarde valores en el , y luego imprímalos en pantalla así como su posición.

```
ejemplo1.c
1  #include<stdio.h>
2
3  int main(){
4      int i, j;
5      int arreglo[4][4];
6
7      //LLENAR EL ARREGLO
8      for(i=0;i<4;i++){
9          for(j=0;j<4;j++){
10             printf("Ingrese un numero: ");
11             scanf("%d", &arreglo[i][j]);
12         }
13     }
14     //IMPRIMIR EL ARREGLO
15     for(i=0;i<4;i++){
16         printf("\n");
17         for(j=0;j<4;j++){
18             printf("%d%d: %d\t",i,j, arreglo[i][j]);
19         }
20     }
21     return 0;
22 }
23
```

```
Instructoria3 : bash — Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
oscar@oscar:~$ cd Documentos/Instructoria3
oscar@oscar:~/Documentos/Instructoria3$ gcc -o ejemplo1 ejemplo1.c
^[[A^[[[oscar@oscar:~/Documentos/Instructoria3$ ./ejemplo1
Ingrese un numero: 1
Ingrese un numero: 2
Ingrese un numero: 3
Ingrese un numero: 4
Ingrese un numero: 5
Ingrese un numero: 6
Ingrese un numero: 7
Ingrese un numero: 8
Ingrese un numero: 9
Ingrese un numero: 10
Ingrese un numero: 11
Ingrese un numero: 12
Ingrese un numero: 13
Ingrese un numero: 14
Ingrese un numero: 15
Ingrese un numero: 16

00: 1   01: 2   02: 3   03: 4
10: 5   11: 6   12: 7   13: 8
20: 9   21: 10  22: 11  23: 12
30: 13  31: 14  32: 15  33: 16  oscar@oscar:~/Documentos/Instructoria3$
```

Ejemplo 2: Comparar si dos matrices de 3x3 son iguales o no

```
1  #include <stdio.h>
2  #define tamano 3
3  int main(){
4      int A[tamano][tamano];
5      int B[tamano][tamano];
6      int fila, columna, comparador;
7      /*Se llena la primera matriz*/
8      printf("Llene la primera matriz A de tamano %dx%d: \n",tamano,tamano);
9      for(fila = 0; fila < tamano; fila++){
10         for(columna = 0; columna < tamano; columna++){
11             printf("Numero - [%d][%d]: ", fila, columna);
12             scanf("%d", &A[fila][columna]);
13         }
14     }//fin llenado
15     /*Se llena la matriz B*/
16     printf("Llene la segunda matriz B de tamano %dx%d: \n", tamano, tamano);
17     for(fila = 0; fila < tamano; fila++){
18         for(columna = 0; columna < tamano; columna++){
19             printf("Numero - [%d][%d]: ", fila, columna);
20             scanf("%d", &B[fila][columna]);
21         }
22     }//fin llenado
23
24     //Asumiendo que las dos matrices sean iguales
25     comparador = 1;
26     //se recorre ambas matrices para comparar sus elementos
27     for(fila = 0; fila < tamano; fila++){
28         for(columna = 0; columna < tamano; columna++){
29             //Si los elementos de las matrices no son iguales
30             if(A[fila][columna] != B[fila][columna]){
31                 /*con un elemento que no sea igual se termina la comparacion
32                 y sale del bucle */
33                 comparador = 0;
34                 break;
35             }
36         }
37     }//fin comparacion
38     /*
39     * Compara el valor de comparador
40     * Si es igual a 1 se asume que ambas matrices son iguales
41     * Caso contrario son distintas
42     */
43     if(comparador == 1){
44         printf("\nLa matriz A es igual a la matriz B");
45     }else{
46         printf("\nLa matriz A es distinta a la matriz B");
47     }
48     return 0;
49 }
50
```

Ejemplo3. Ordenar una matriz en orden ascendente. Haciendo uso del método de la burbuja.

```
ejemplo3.c
1  #include<stdio.h>
2  #define filas 2
3  #define columnas 2
4  int main(){
5      int matriz[filas][columnas];
6      int i, j , x, y, auxiliar;
7      //LLENAR EL ARREGLO
8      for(i=0;i<filas;i++){
9          for(j=0;j<columnas;j++){
10             printf("Ingrese un numero: ");
11             scanf("%d", &matriz[i][j]);
12         }
13     }
14     printf("\n");
15     //IMPRIMIR MATRIZ DESORDENADA
16     printf("Matriz desordenada..... \n");
17     for(i=0;i<filas;i++){
18         printf("\n");
19         for(j=0;j<columnas;j++){
20             printf("%d\t", matriz[i][j]);
21         }
22     }
23     //ORDENAR LA MATRIZ
24     for(i=0;i<filas;i++){
25         for(j=0;j<columnas;j++){
26             for(x=0;x<filas;x++){
27                 for(y=0;y<columnas;y++){
28                     if(matriz[i][j]<matriz[x][y]){
29                         auxiliar=matriz[i][j];
30                         matriz[i][j]=matriz[x][y];
31                         matriz[x][y]=auxiliar;
32                     }
33                 }
34             }
35         }
36     }
37     printf("\n");
38     //IMPRIMIR MATRIZ YA ORDENADA
39     printf("Matriz ordenada..... \n");
40     for(i=0;i<filas;i++){
41         printf("\n");
42         for(j=0;j<columnas;j++){
43             printf("%d\t", matriz[i][j]);
44         }
45     }
46     printf("\n");
47     return 0;
48 }
49
```

```
Instructoria3 : bash — Konsole

Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda

oscar@Oscar:~/Documentos/Instructoria3$ gcc -o ejemplo3 ejemplo3.c
oscar@Oscar:~/Documentos/Instructoria3$ ./ejemplo3
Ingrese un numero: 4
Ingrese un numero: 3
Ingrese un numero: 2
Ingrese un numero: 1

Matriz desordenada.....
4      3
2      1
Matriz ordenada.....
1      2
3      4
oscar@Oscar:~/Documentos/Instructoria3$
```

PROBLEMAS PROPUESTOS.

1. Modifique el ejemplo3 pero que ahora el programa ordene en forma descendente.
2. Cree un programa que sume 2 matrices.
3. Cree un programa que sume todos los elementos de una matriz de n filas * n columnas.
4. Cree un programa que permita al usuario:
 - Crear una matriz con las dimensiones que desee.
 - Llenar las matrices con números al azar a través del teclado.
 - Sumar ya sea todos los elementos de una columna o de una fila, el programa deberá preguntar que desea sumar(filas, columnas) y luego mostrar el resultado.