

Declaración de arreglos multidimensionales

En C++ es posible tener arreglos multidimensionales. Para definir un arreglo multidimensional utiliza el siguiente formato:

```
tipo Nom_var[tam1][tam2]?[tamN];
```

y para acceder un elemento de un arreglo multidimensional se utiliza:

```
Nom_var[sub1][sub2]?[subN];
```

Nota que cada subíndice se encierra en un par de corchetes([]).

El uso más común de los arreglos multidimensionales es para representar arreglos de 2 dimensiones o tablas, en donde los datos

están acomodados por renglones y columnas. Por convención el primer subíndice se refiere a los renglones y el segundo a las

columnas.

Por ejemplo podemos definir un arreglo de elementos enteros de 2 dimensiones con 3 renglones y 4 columnas usando la siguiente

instrucción:

```
int lista[3][4];
```

y por ejemplo para acceder el elemento que se encuentra en el renglón 0 y la columna 2 de este arreglo utilizamos la variable:

```
lista[0][2];
```

Aunque no es muy común, también podemos definir arreglos de más de 2 dimensiones, por ejemplo, podemos definir un arreglo de

elementos enteros de 3 dimensiones con 4 renglones, 3 columnas y 2 niveles de profundidad usando la siguiente línea.

```
int cubo[4][3][2];
```

También es posible declarar el arreglo declarando primero un tipo con typedef de la siguiente forma:

```
typedef tipo nombre_tipo[tamaño1][tamaño2]; // declaración del tipo
```

```
nombre_tipo Nombre_variable; // declaración de la variable de tipo arreglo
```

Por ejemplo:

```
typedef double tabla[8][4];
```

```
tabla lista;
```

En este caso se declara el tipo de dato tabla que es un arreglo de 2 dimensiones de elementos de tipo double, cuyos subíndices

van del valor 0 al 7 y del 0 al 3. En el segundo renglón de la declaración se declara el arreglo del tipo tabla.

En C++ un arreglo de 2 dimensiones es en realidad un arreglo de arreglos. Por ejemplo si declaramos el arreglo

```
int arr[10][5];
```

en realidad estamos declarando un arreglo de 10 elementos, cuyos elementos son arreglos de 5 elementos de tipo entero. Este es

la manera en que internamente se maneja el arreglo y el programador no tendría porque enterarse de ello; aquí lo mencionamos

porque esto nos ayudará a entender algunas características de los arreglos de 2 dimensiones.

Inicializar arreglos en la declaración

Cuando se requiere inicializar en la declaración un arreglo de 2 dimensiones es útil recordar que en C++ internamente un arreglo de

dos dimensiones es un arreglo de arreglos, entonces se debe dar el valor para cada uno de los arreglos (como una lista de

elementos encerrados entre llaves y separados por comas) como se muestra en el siguiente ejemplo:

```
int lista[3][2] = {{1,2},{3,4},{5,6}};
```

Ejemplo:

```
// Programa que encuentra el valor mayor de la matriz, y ademas dice  
// en que renglon y en que columna se encuentra dicho valor.  
#include <iostream.h>
```

```
typedef int Matriz[20][20];
```

```

int main()
{
    Matriz A;           // Variable de tipo arreglo de 2 dimensiones
    int N, M;           // Dimensiones de la matriz
    int ren, col;       // variables auxiliares para las búsquedas
    int mayor;          // variable en la que se guardará el valor mayor
    int renM, colM;     // variables que guardarán el reng y la col en las
                        // que se encontró el valor mayor

    cout<<"Cuantos renglones tiene la matriz? ";
    cin>>N;
    cout<<"Cuantas columnas tiene la matriz? ";
    cin>>M;

    // pide los datos para la matriz
    for (ren=0; ren < N; ren++)
        for (col=0; col < M; col++)
        {
            cout<<"Teclea el valor para el elemento "<<ren<<","<<col<<" ";
            cin>>A[ren][col];
        }

    // muestra la matriz
    for (ren=0; ren<N; ren++)
    {
        for (col=0; col<M; col++)
            cout<<A[ren][col]<<"\t";
        cout<<endl;
    }

    //Busca el valor mayor de la matriz

    mayor = A[0][0];
    renM = 0;
    colM = 0;
    for (ren = 0; ren < N; ren++)
        for (col = 0; col < M; col++)
            if (A[ren][col] > mayor)
            {
                mayor = A[ren][col];
                renM = ren;
                colM = col;
            }

    cout<<"El valor mayor es "<<A[renM][colM]<<endl;
    cout<<"y se encuentra en las posiciones "<<endl;
    cout<<"renglon "<<renM<<" y columna "<<colM<<endl;

    return 0;
}

```

Ejercicio

Realiza un programa que pida al usuario las dimensiones deseadas para un arreglo de dos dimensiones y luego le pida los datos para

llenar el arreglo. Después el programa debe dar la opción de calcular la suma de cualquier renglón o columna del arreglo. Tu

programa debe trabajar con arreglos que tengan como cantidad máxima de renglones y/o columnas 10. [ver solución](#)

Ligas sugeridas

<http://www.cplusplus.com/doc/tutorial/>

<http://www.cs.wustl.edu/~schmidt/C++/>

[Regresar](#)

[Siguiete módulo](#)