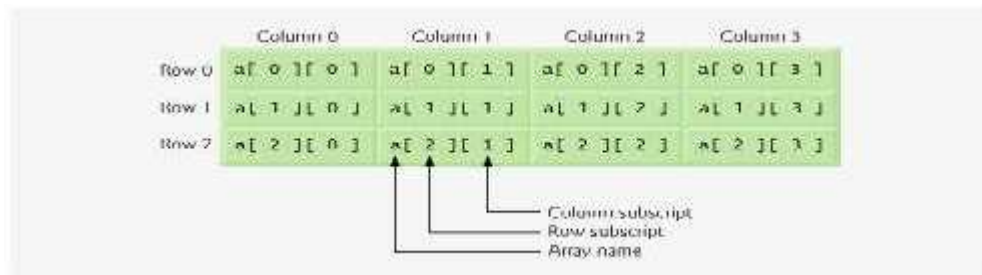


Multidimensional Arrays

Multidimensional arrays with two dimensions are often used to represent **tables of values** consisting of information arranged in rows and columns. To identify a particular table element, we must specify two subscripts. By convention, the first identifies the element's row and the second identifies the element's column. Arrays that require two subscripts to identify a particular element are called **two-dimensional arrays** or **2-D arrays**. Note that multidimensional arrays can have more than two dimensions (i.e., subscripts). Figure 1 illustrates a two-dimensional array, a. The array contains three rows and four columns, so it is said to be a 3-by-4 array. In general, an array with m rows and n columns is called an **m-by-n array**.

Figure 7.21. Two-dimensional array with three rows and four columns.



Every element in array a is identified in Fig. 1 by an element name of the form a[i][j], where a is the name of the array, and i and j are the subscripts that uniquely identify each element in a. Notice that the names of the elements in row 0 all have a first subscript of 0; the names of the elements in column 3 all have a second subscript of 3.

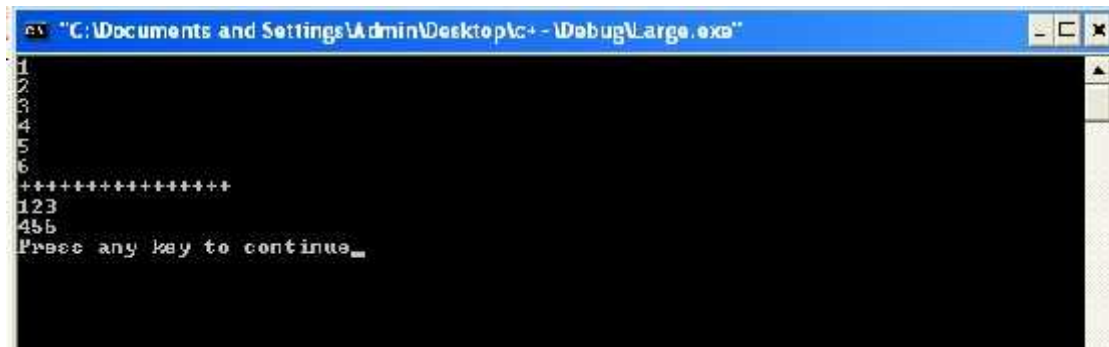
Initializing 2D-Array Elements:

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

Read / Write / Process Array Elements:

Example: Write C++ program, to read 6 numbers, 3 numbers per row, the print them:

```
#include<iostream.h>
void main ()
{
    int a [ 2 ] [ 3 ];
    int i , j;
    for ( i = 0 ; i < 2; i++ )
        for ( j = 0 ; j < 3; j++ )
            cin >> a [ i ] [ j ];
    cout<<"+++++++\n";
    for ( i = 0 ; i < 2; i++ )
    {
        for ( j = 0 ; j < 3; j++ )
            cout << a [ i ] [ j ];
        cout << endl;
    }
}
```



```
1  
2  
3  
4  
5  
6  
+++++  
123  
456  
Press any key to continue_
```

Example: Write C++ program, to read 3*3 2D-array, then find the summation of the array elements, finally print these elements:

```
#include<iostream.h>  
void main ( )  
{  
    int a [ 3 ] [ 3 ];  
    int i , j, sum = 0;  
    for ( i = 0 ; i < 3; i++ )  
        for ( j = 0 ; j < 3; j++ )  
            cin >> a [ i ] [ j ];  
    for ( i = 0 ; i < 3; i++ )  
        for ( j = 0 ; j < 3; j++ )  
            sum += a [ i ] [ j ];  
    cout << "sum is: " << sum << endl;  
    for ( i = 0 ; i < 3; i++ )  
    {  
        for ( j = 0 ; j < 3; j++ )  
            cout << a [ i ] [ j ];  
        cout << endl;  
    }  
}
```



```
1  
2  
3  
4  
5  
6  
7  
8  
9  
sum is: 45  
123  
456  
789  
Press any key to continue
```

Example: Write C++ program, to read 3*4 2D-array, then find the summation of each row:

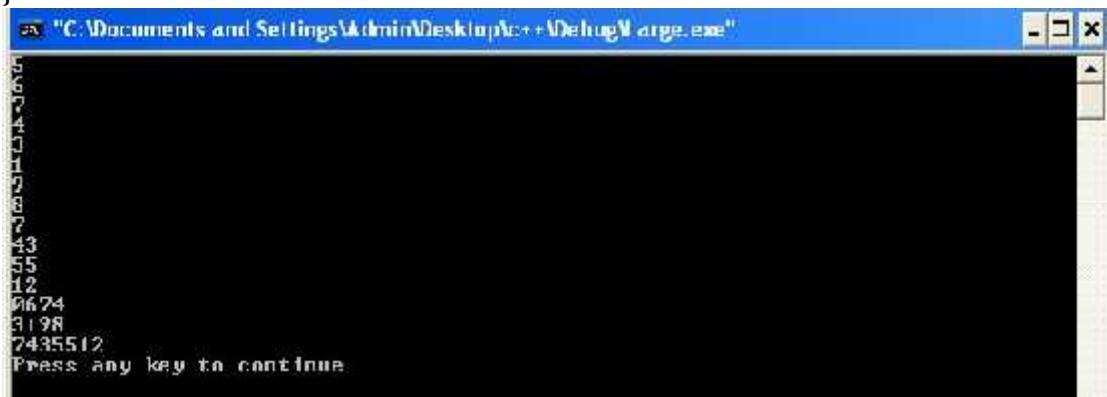
```
#include<iostream.h>
void main ( )
{
    int a [ 3 ] [ 4 ];
    int i , j, sum = 0;
    for ( i = 0 ; i < 3; i++ )
        for ( j = 0 ; j < 4; j++ )
            cin >> a [ i ] [ j ];
    for ( i = 0 ; i < 3; i++ )
    {
        sum = 0;
        for ( j = 0 ; j < 4; j++ )
            sum += a [ i ] [ j ];
        cout << "sum of row " << i << " is: " << sum << endl;
    }
}
```



Example: Write C++ program, to read 3*4 2D-array, then replace each value equal 5 with 0:

```
#include<iostream.h>
void main ( )
{
    int a [ 3 ] [ 4 ];
    int i , j;
    for ( i = 0 ; i < 3; i++ )
        for ( j = 0 ; j < 4; j++ )
            cin >> a [ i ] [ j ];
    for ( i = 0 ; i < 3; i++ )
        for ( j = 0 ; j < 4; j++ )
            if ( a [ i ] [ j ] == 5 ) a [ i ] [ j ] = 0;
    for ( i = 0 ; i < 3; i++ )
    {
        for ( j = 0 ; j < 4; j++ )
```

```
        cout << a [ i ] [ j ];  
    cout << endl;  
    }  
}
```



```
"C:\Documents and Settings\Admin\Desktop\c++\Debug\arg.exe"  
5  
3  
4  
1  
1  
2  
3  
3  
4  
5  
5  
12  
0674  
3198  
7435512  
Press any key to continue
```

