C++ Multidimensional Arrays & Strings:-

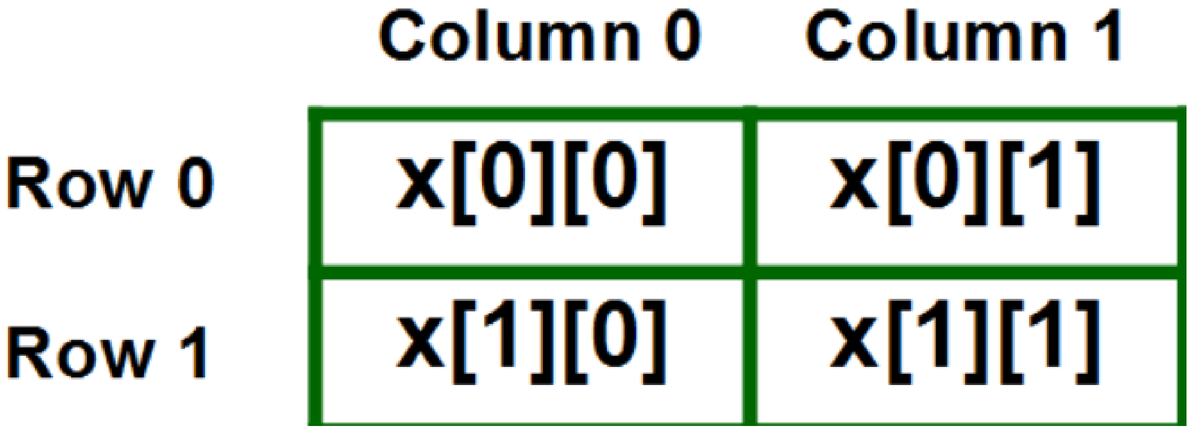
A multidimensional array is an array with more than one dimension.

Declaration:

```
Int array[2][2];            // Total 2x2 = 4 elements will be stored
```

How elements are stored in 2D Array:

A two-dimensional array in C++ is a collection of elements organized in rows and columns.



	Column 0	Column 1
Row 0	x[0][0]	x[0][1]
Row 1	x[1][0]	x[1][1]

Initializing 2D Array using the initializer List:

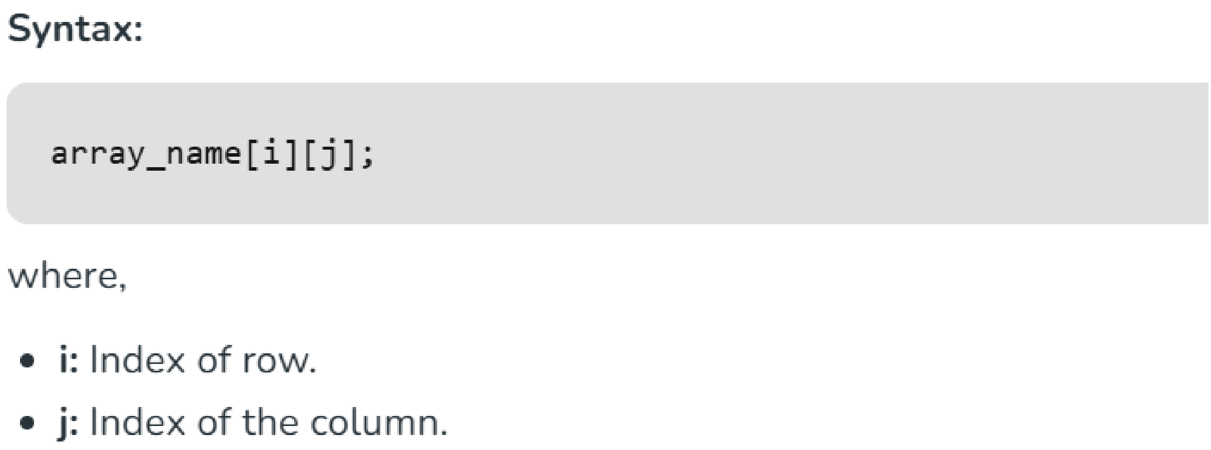
1st Method:

```
int arr[2][4] = {0, 1, 2, 3, 4, 5, 6, 7};
```

2nd Method:

```
int x[2][4] = {{0, 1, 2, 3}, {4, 5, 6, 7}};
```

Accessing Elements of Two-Dimensional Arrays in C++



Syntax:

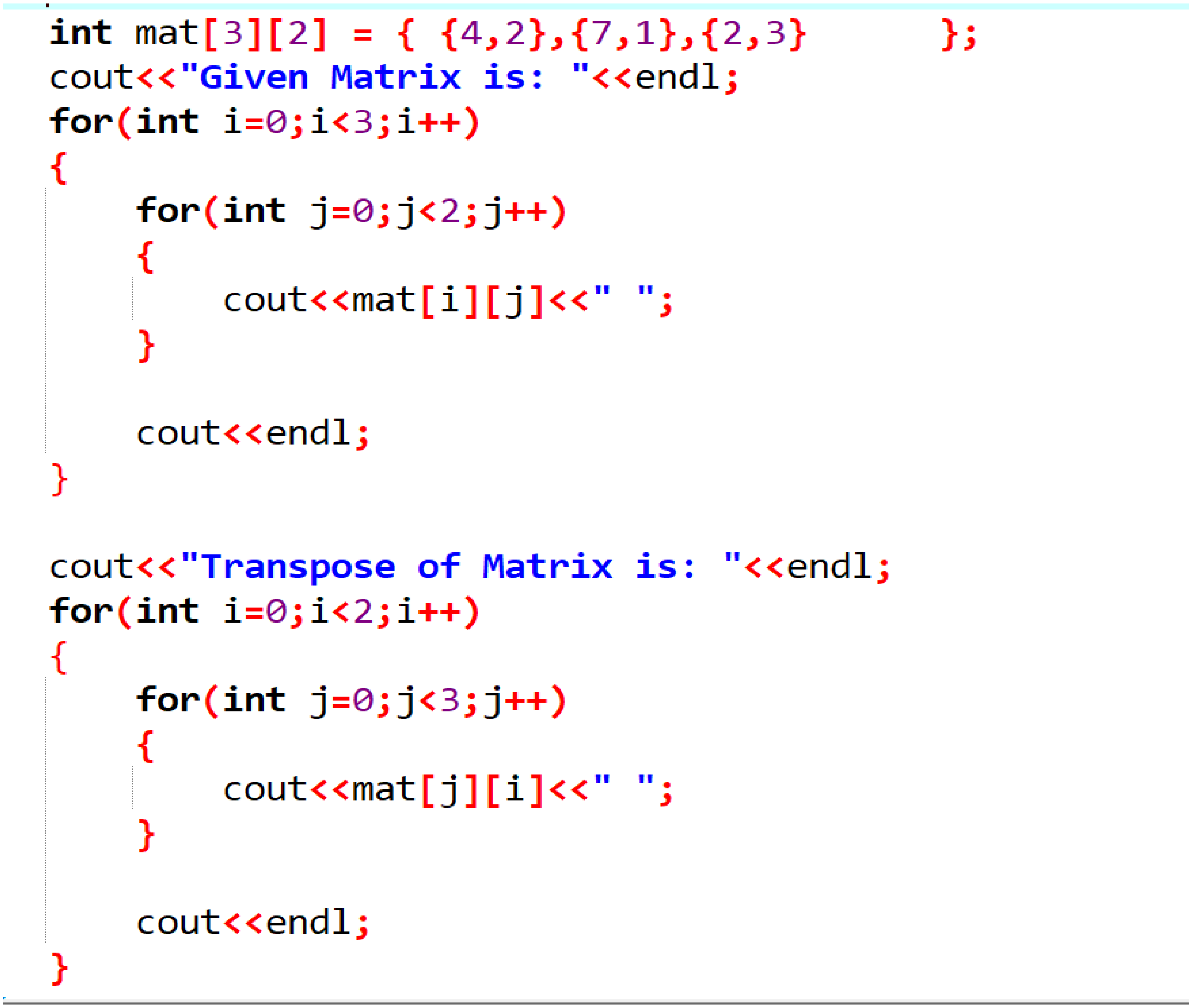
```
array_name[i][j];
```

where,

- **i:** Index of row.
- **j:** Index of the column.

Solving Real-World Problem:

Transpose of a Matrix:



```
int mat[3][2] = { {4,2},{7,1},{2,3}      };
cout<<"Given Matrix is: "<<endl;
for(int i=0;i<3;i++)
{
    for(int j=0;j<2;j++)
    {
        cout<<mat[i][j]<<" ";
    }

    cout<<endl;
}

cout<<"Transpose of Matrix is: "<<endl;
for(int i=0;i<2;i++)
{
    for(int j=0;j<3;j++)
    {
        cout<<mat[j][i]<<" ";
    }

    cout<<endl;
}
```

C-Style string:

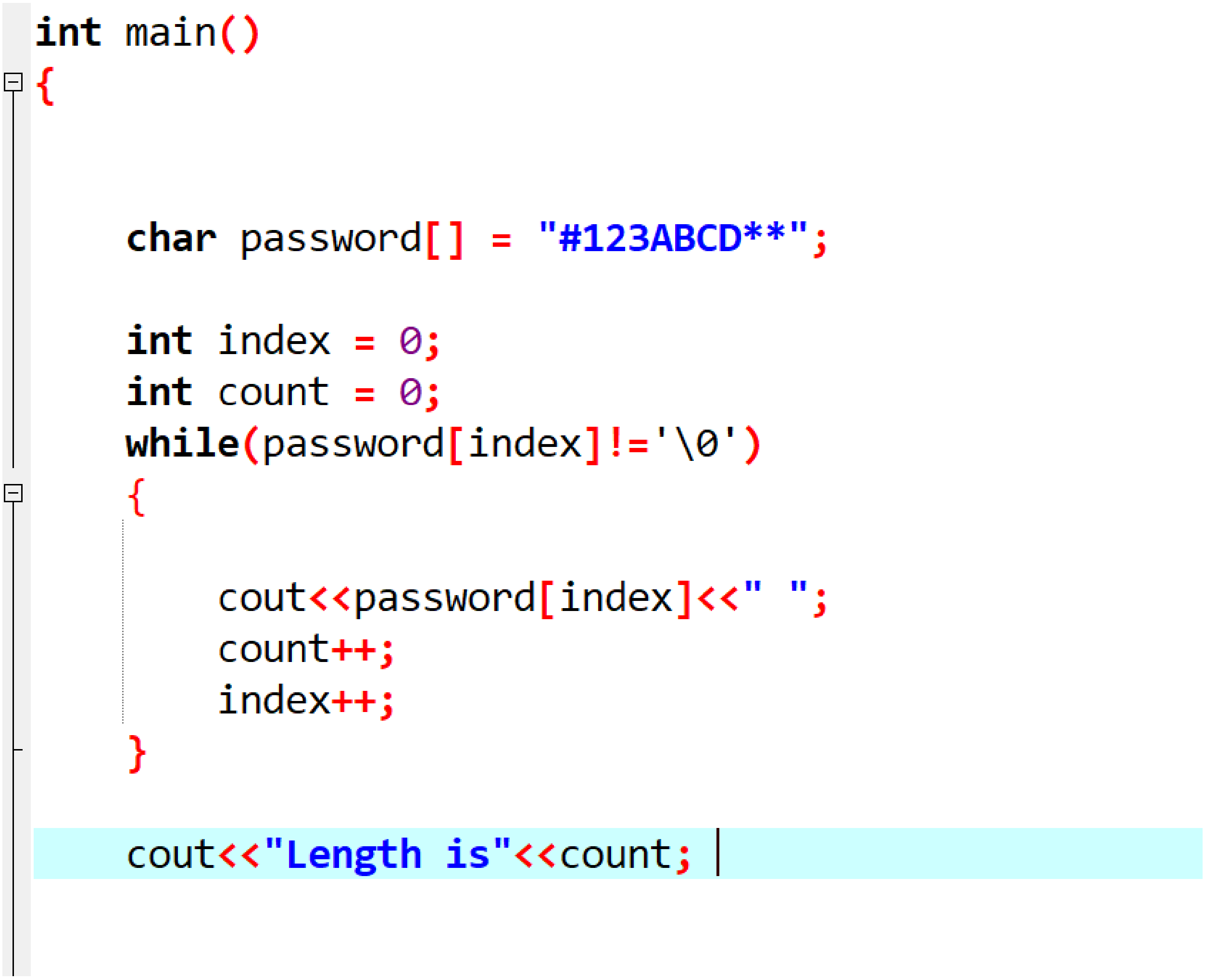
Char str[] = “ABCD”;

\0 will represent the end of the string.

Char str[] = {1,2,3,4};

No \0 will be found at the end of the string.

ASCII value of \0 is 0.



```
int main()
{


    char password[] = "#123ABCD**";

    int index = 0;
    int count = 0;
    while(password[index]!='\0')
    {

        cout<<password[index]<<" ";
        count++;
        index++;
    }

    cout<<"Length is"<<count;

```

Lab Task:

1. Find row-wise sum of the matrix (3x3), you can initialize or take input from the user. And then make another matrix (3x4) and write sum value of each row.
2. Check weather a matrix is identity or not. (3x3).

3. Read a string from the user and give number according to the following criteria.

    i. If you found special symbol, give one marks/number.

ii. If you found numeric digit, give one marks/number.

iii. If you found letter, give another marks/number.

iv. If the length of the password is greater than 6, then give another mark.

v. If you found capital letter, then give another number.

vi. If total number is 5, then show strong password otherwise weak password.

Do not use built-in function of string.