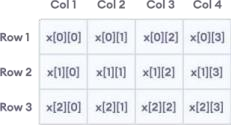
**ARRAYS-II (Multi-dimensional arrays)**

In C++, we can create an array of an array, known as a multidimensional array. For example: **int x[3][4];**

Here, x is a two-dimensional array. It can hold a maximum of 12 elements. We can think of this array as a table with 3 rows and each row has 4 columns as shown below.



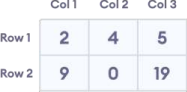
# Initialization of two-dimensional array

## int test[2][3] = {2, 4, 5, 9, 0, 19};

The above method is not preferred. A better way to initialize this array with the same array elements is given below:

## int test[2][3] = { {2, 4, 5}, {9, 0, 19}};

This array has 2 rows and 3 columns, which is why we have two rows of elements with 3 elements each.



|  |
| --- |
| Example: Two-Dimensional Array // C++ Program to display all elements  // of an initialized two-dimensional array  **#include <iostream>**  **using namespace std;**  **int main() {**  **int test[3][2] = {{2, -5}, {4, 0}, {9, 1}};**  // use of nested for loop  // access rows of the array  **for (int i = 0; i < 3; ++i) {**  // access columns of the array  **for (int j = 0; j < 2; ++j)** **{**  **cout << "test[" << i << "][" << j << "] = " << test[i][j] << endl; }**  **}**  **return 0;**  **}** |
| **Output**  test[0][0] = 2  test[0][1] = -5  test[1][0] = 4  test[1][1] = 0  test[2][0] = 9  test[2][1] = 1 |

# 2D array as Function arguments:

* Use array name as argument in function call:

## getExams(exams, 2);

* Use empty [] for row, size declarator for column in prototype, header:

## const int COLS = 2;

// Prototype

## void getExams(int [][COLS], int);

// Header

**void getExams(int exams[][COLS], int rows)**

# Example:

|  |
| --- |
| // C++ Program to display the elements of two// dimensional array by passing it to a function#include <iostream>using namespace std;const int col=2;// define a function// pass a 2d array as a parametervoid display(int n[][col], int row) {cout << "Displaying Values: " << endl;for (int i = 0; i <row ; ++i) {for (int j = 0; j < col; ++j) {cout << "num[" << i << "][" << j << "]: " << n[i][j] << endl;}}}int main() {// initialize 2d arrayint num[3][2] = {{3, 4},{9, 5},{7, 1}};// call the function// pass a 2d array as an argumentdisplay(num,3);return 0;} |
| Output: |

**LAB TASK**

**Note: Work in functions for the following lab tasks**

## Problem 1:

Write a program that allows two players to play a game of tic-tac-toe. Use a two dimensional char array with three rows and three columns as the game board. Each element of the array should be initialized with an asterisk (\*). The program should run a loop that

* Displays the contents of the board array
* Allows player 1 to select a location on the board for an X. The program should ask the user to enter the row and column number.
* Allows player 2 to select a location on the board for an O. The program should ask the user to enter the row and column number.
* Determines whether a player has won, or a tie has occurred. If a player has won, the program should declare that player the winner and end. If a tie has occurred, the program should say so and end.

Player 1 wins when there are three Xs in a row on the game board. The Xs can appear in a row, in a column, or diagonally across the board. A tie occurs when all of the locations on the board are full, but there is no winner.

|  |
| --- |
| Submission Instructions:  1. Save all .cpp files and screenshot with task number e.g. Task01.cpp 2. Now create a new folder with name ROLLNO\_LAB12 e.g. i22XXXX\_LAB12 3. Move all of your .cpp files and screenshots to this newly created directory and compress it into .zip file. 4. Now you have to submit this zipped file on Google Classroom. |