**National University of Computer and Emerging Sciences**

****

Lab Manual # 11

Programming Fundamentals

(Section BCS-1J)

|  |  |
| --- | --- |
| Course Instructor | Mr Owais Idrees |
| Lab Instructor(s) | Muntaha Zaigham  Abiha Aftab |
| Section | BCS-1J |
| Semester | Fall 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

2-D 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.



## Multidimensional Array Initialization

Like a normal array, we can initialize a multidimensional array in more than one way.

### 1. 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 1: Two Dimensional Array

// C++ Program to display all elements

// of an initialised 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

# Lab Manual:

**Problem 1: (2-D array)**

Write a program which should detect whether a given matrix is identity or not.

An identity matrix is a matrix is a matrix whose diagonal entries are 1 and non-diagonal entries

are 0.

**Problem 2:**

Write a program for addition/subtraction of two Matrices of same size.

**Problem 3:**

Write a user defined function named Upper-half() which takes a two dimensional array A, with size N rows and N columns as argument and prints the upper half of the array.

e.g.,  
2 3 1 5 0                              2 3 1 5 0  
7 1 5 3 1                                 1 5 3 1  
2 5 7 8 1   Output will be:       1 7 8  
0 1 5 0 1                                       0 1  
3 4 9 1 5                                          5

**Problem 4:**

Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column.  
[Assuming the 2D Array to be a square matrix with odd dimension i.e. 3x3, 5x5, 7x7 etc...]

Sample Run:

Input: 2 dimension 3X3 array

3  5  4  
7  6  9  
2  1  8

Output through the function should be:  
Middle Row: 7 6 9

Middle column: 5

6

1

**Problem 5:**

Write a program to find transpose of a matrix

