Faculty of Computing

CS110: Fundamentals of Computer Programming

Class: BESE-16B

Lab 10: Two-Dimensional Arrays

CLO 2	Solve given real-world problem by applying appropriate programming concepts and techniques.
CLO 3	Build a program and associated documentation using appropriate IDE and supplementary tools.

Date: 18th November 2025

Time: 2:00 pm-5:00 pm

Instructor: Dr. Momina Moetesum

Lab Engineer: Mr. Nadeem Nawaz

Lab 10: Two-Dimensional Arrays

Learning Objectives:

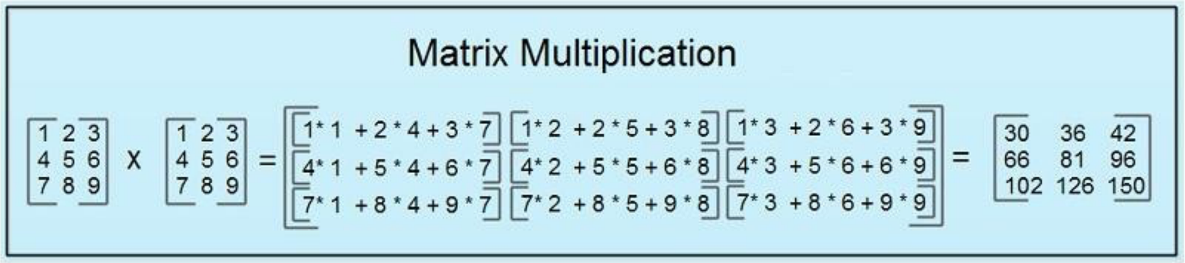
After completing this section, you will be able to:

- Insert and access data into and from 2-Dimensional arrays.
- Use 2-D arrays for matrix arithmetic.
- Use 2-D arrays for understanding basic image binarization.
- Use 2-D arrays for Adjacency Graph Representation.

Lab Tasks

Task 1[CLO1]

Write a C++ program that takes input from user for two 3x3 integer matrix and return the product of both. Display resultant matrix in rows and columns as shown in figure below.



$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix} \text{X} \begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix} = \begin{bmatrix}[1*1+2*4+3*7] & [1*2+2*5+3*8] & [1*3+2*6+3*9]\\ [4*1+5*4+6*7] & [4*2+5*5+6*8] & [4*3+5*6+6*9]\\ [7*1+8*4+9*7] & [7*2+8*5+9*8] & [7*3+8*6+9*9]\end{bmatrix} = \begin{bmatrix}30&36&42\\66&81&96\\102&126&150\end{bmatrix}$$

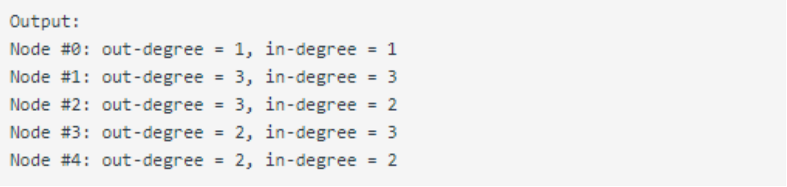
Task 2: [CLO 2]

Thresholding is the first step in any image segmentation technique. Write a C++ program that randomly initializes an image matrix of size 6x6 with pixel intensity values ranging from 0-255 (using rand() function). It then computes the average of all values and uses it as a threshold to binarize the image. The program should display the initial image matrix, threshold value, and the binarized image matrix. *Note: All values less than or equal to threshold become 0 and all values greater than the threshold become 1.

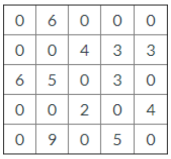
$$g(x,y)=\begin{cases}1\,if\ f(x,y)>T\\0\,if\ f(x,y)\le T\end{cases}$$

Task 3: [CLO 3]

Given the adjacency matrix of a directed graph, write a C++ program that computes the out-degree and the in-degree of each node. The in-degree of a node is computed using the sum of all its rows while the out-degree is computed as the sum of all its columns. The output should be in the following format:



```
Output:
Node #0: out-degree = 1, in-degree = 1
Node #1: out-degree = 3, in-degree = 3
Node #2: out-degree = 3, in-degree = 2
Node #3: out-degree = 2, in-degree = 3
Node #4: out-degree = 2, in-degree = 2
```



0	6	0	0	0
0	0	4	3	3
6	5	0	3	0
0	0	2	0	4
0	9	0	5	0

Deliverables:

Compile a single Word document with codes for each question and screenshots of the outputs and submit this Word file on LMS.

Lab Rubrics

Your Lab 10 will be graded out of 5 for each rubric according to the following rubrics. Grades for CLO3 will be shared at different intervals during the semester.

Lab Rubrics for Lab 10 (Two-dimensional Arrays)

Sr. No.	Assessment	Unacceptable (0 Marks)	Does Not Meet Expectations (1/2 Marks)	Meets Expectations (3/4 Marks)	Exceeds Expectations (5 Marks)
1	**Application of Programming Concepts** **(CLO2, PLO3)**	The student did not submit any work. OR The student plagiarized the solution and/or used unfair means.	The student is unable to apply the appropriate programming concepts to solve the given problem thus resulting in an incomplete or ineffective solution. The program flow is messy and incomprehensible. Codes are non-modular and cannot be reused.	The student requires some guidance to apply the appropriate programming concepts to solve the given problem. The program flow requires minor improvements. Codes are semi-modular and semi-reusable.	The student demonstrates a clear ability to apply the appropriate programming concepts to solve the given problem. The program flow is adequate. Codes are modular, reusable, and easily readable.
2	**Software Tool Usage** **(CLO3-PLO5)**		The student demonstrates a lack of understanding of tool usage. Implementation has syntax/semantic/runtime errors, and the student is unable to debug and correct the errors. The code has inadequate comments and variable names and does not adhere to the coding standards. No Error handling has been performed. Documentation is poorly structured.	The student demonstrates some understanding of tool usage. The codes are correct in terms of their syntax, however, the program output is not always correct in all test cases. The code has limited comments and inconsistent variable names and may not adhere to the coding standards. Some Error handling has been performed. Documentation is adequately structured.	The student demonstrates a good understanding of tool usage. Furthermore, his/her coding is complete and functional, and the program output is correct in all test cases. The code has sufficient comments and consistent variable names and reasonably adhere to the coding standards. Adequate Error handling has been performed. Documentation is well structured.