Data Structure Algorithm & Application (CT-159)
Lab 02

Metrix (2D Arrays) in C++

Objective:

The objective of this experiment to get familiar with

1. 2D Arrays (metrices) in C++
2. Operations that can be performed on metrix in C++.
3. Implementation of OOP concepts (classes and functions)

Introduction:

In C++, a matrix can be represented using a 2D array or a vector of vectors. Here’s how you can implement and work with matrices in C++.

1. Using a 2D Array:

A 2D array is a simple and fixed-size way to represent a matrix. Here’s how you can define, initialize, and perform basic operations like input, output, and addition on a matrix using a 2D array.

2. Using a Vector of Vectors:

Vectors offer dynamic sizing, making them more flexible than 2D arrays. You can use std::vector to create a matrix where the size can be determined at runtime.

Example Code:

```
vector<vector<int>> matrix(rows, vector<int>(cols));


  // Input elements in the matrix
  cout << "Enter elements of the matrix:" << endl;
  for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
      cin >> matrix[i][j];
    }
  }
```

Matrix Operations:

Matrix Addition:

Matrix A and B: Two matrices A and B are defined with predefined values.

Matrix Addition: A loop is used to add corresponding elements of A and B and store them in matrix C.

Result Display: The resulting matrix C is displayed.

Example Code:

```
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        C[i][j] = A[i][j] + B[i][j];
    }
  }
```

Matrix Subtraction:

Matrix subtraction in C++ can be implemented similarly to matrix addition. The idea is to subtract corresponding elements of two matrices and store the result in a third matrix.

Matrix Multiplication:

Matrix multiplication is a binary operation that takes a pair of matrices and produces another matrix. To multiply two matrices, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix has the same number of rows as the first matrix and the same number of columns as the second matrix.

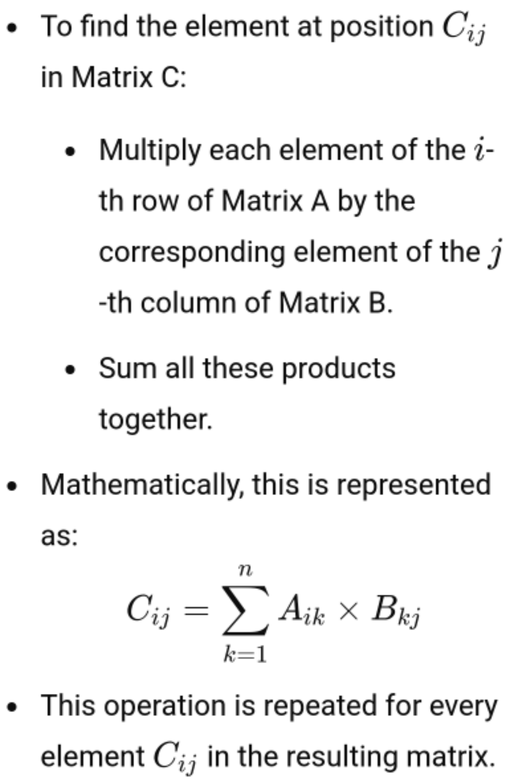
Steps for Matrix Multiplication:

1- Matrix Dimensions:

Let’s assume you have two matrices:
Matrix A of size $m \times n$
Matrix B of size $n \times p$

The result will be Matrix C of size $m \times p$

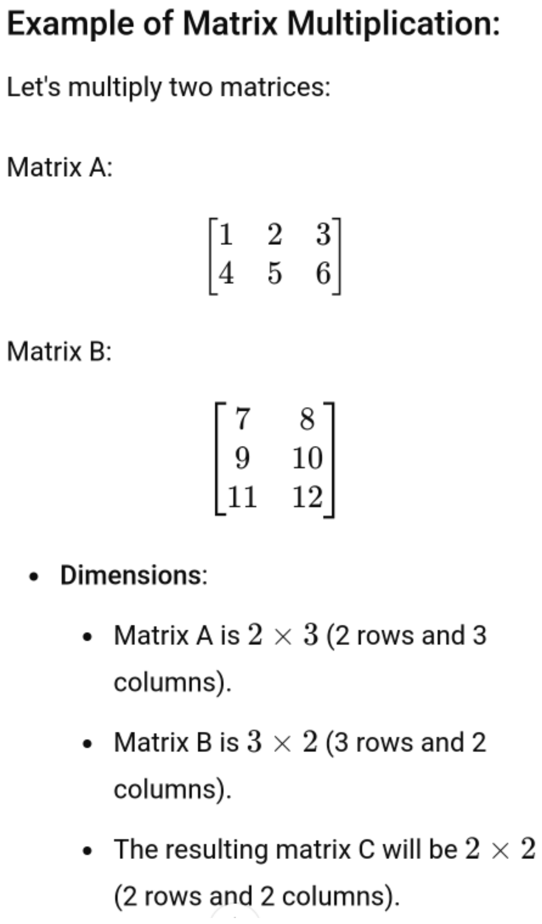
2- Element Calculation:



- To find the element at position C_{ij} in Matrix C:
  - Multiply each element of the i-th row of Matrix A by the corresponding element of the j-th column of Matrix B.
  - Sum all these products together.
- Mathematically, this is represented as:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj}$$

- This operation is repeated for every element C_{ij} in the resulting matrix.



Example of Matrix Multiplication:

Let's multiply two matrices:

Matrix A:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Matrix B:

$$\begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$$

- **Dimensions**:
  - Matrix A is 2×3 (2 rows and 3 columns).
  - Matrix B is 3×2 (3 rows and 2 columns).
  - The resulting matrix C will be 2×2 (2 rows and 2 columns).

- **Calculating Each Element**:
  - $C_{11} = (1 \times 7) + (2 \times 9) + (3 \times 11) = 7 + 18 + 33 = 58$
  - $C_{12} = (1 \times 8) + (2 \times 10) + (3 \times 12) = 8 + 20 + 36 = 64$
  - $C_{21} = (4 \times 7) + (5 \times 9) + (6 \times 11) = 28 + 45 + 66 = 139$
  - $C_{22} = (4 \times 8) + (5 \times 10) + (6 \times 12) = 32 + 50 + 72 = 154$
- **Resulting Matrix C**:

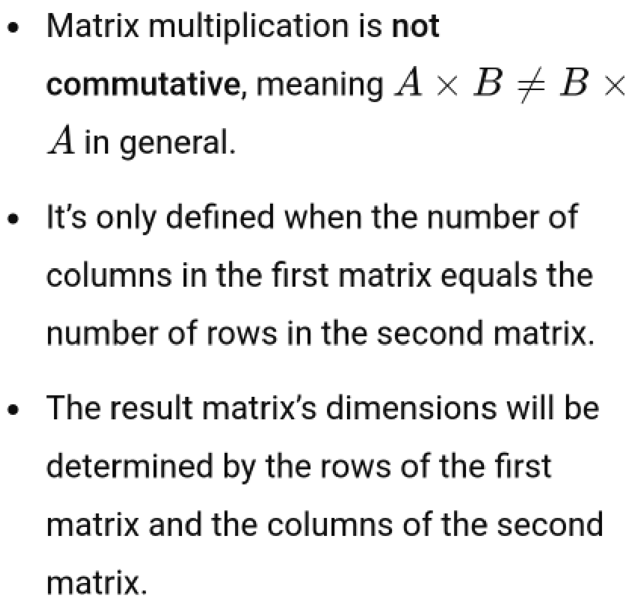
$$\begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

Example Code:

```
int m = 2, n = 3, p = 2;
   int A[2][3] = { {1, 2, 3}, {4, 5, 6} };
   int B[3][2] = { {7, 8}, {9, 10}, {11, 12} };
   int C[2][2] = {0};

   // Multiplying matrix A and B and storing the result in matrix C
   for(int i = 0; i < m; i++) {
      for(int j = 0; j < p; j++) {
         for(int k = 0; k < n; k++) {
            C[i][j] += A[i][k] * B[k][j];
         }
      }
   }
```

Points to Remember:



- Matrix multiplication is **not commutative**, meaning $A \times B \neq B \times A$ in general.
- It's only defined when the number of columns in the first matrix equals the number of rows in the second matrix.
- The result matrix's dimensions will be determined by the rows of the first matrix and the columns of the second matrix.

Matrix multiplication is a fundamental operation in many areas, including computer graphics, physics simulations, and machine learning algorithms.

Exercise:

1- Write a program in c++ that performs binary search on a 2D matrix. Implement OOP concepts (make a class and define function for searching an element in an array).

2- Write a program in C++ with a class Matrix that performs addition, subtraction, multiplication of two matrices. Multiplies a matrix with a constant and calculates transpose matrix.

3- Suppose A and B are n-elements vector array in memory and X and Y are scalars. Write a program to find.

- XA + YB
- A . B

Test the program using A= (16, -6,7), B=(4,2,-3), X= 2, Y= -5

<table>
<tr><th colspan="3">Lab 01 Evaluation</th></tr>
<tr><td colspan="3">Student Name: Student ID: Date:</td></tr>
<tr><td>Rubric</td><td>Marks (25)</td><td>Remarks by teacher in accordance with the rubrics</td></tr>
<tr><td>R1</td><td></td><td></td></tr>
<tr><td>R2</td><td></td><td></td></tr>
<tr><td>R3</td><td></td><td></td></tr>
<tr><td>R4</td><td></td><td></td></tr>
<tr><td>R5</td><td></td><td></td></tr>
</table>