Name: Neelansh Pandey

MBA Tech Computer Engineering

Collaborating with : Arnav Sharma

Topic chosen for GitHub project: Matrix operation program

Programming language used : C++

Code for my project:

#include <iostream>

using namespace std;

void inputMatrix(int rows, int cols, int matrix[10][10]) {

cout << "Enter elements of the matrix (" << rows << "x" << cols << "):\n";

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cin >> matrix[i][j];

}

}

}

void displayMatrix(int rows, int cols, int matrix[10][10]) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void addMatrices(int rows, int cols, int mat1[10][10], int mat2[10][10], int result[10][10]) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = mat1[i][j] + mat2[i][j];

}

}

}

void subtractMatrices(int rows, int cols, int mat1[10][10], int mat2[10][10], int result[10][10]) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = mat1[i][j] - mat2[i][j];

}

}

}

void multiplyMatrices(int rows1, int cols1, int mat1[10][10], int rows2, int cols2, int mat2[10][10], int result[10][10]) {

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

result[i][j] = 0;

for (int k = 0; k < cols1; k++) {

result[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

}

int main() {

int mat1[10][10], mat2[10][10], result[10][10];

int rows1, cols1, rows2, cols2;

int choice;

cout << "Enter rows and columns for Matrix 1: ";

cin >> rows1 >> cols1;

inputMatrix(rows1, cols1, mat1);

cout << "Enter rows and columns for Matrix 2: ";

cin >> rows2 >> cols2;

inputMatrix(rows2, cols2, mat2);

cout << "Choose an operation:\n1. Addition\n2. Subtraction\n3. Multiplication\nEnter your choice: ";

cin >> choice;

switch (choice) {

case 1:

if (rows1 == rows2 && cols1 == cols2) {

addMatrices(rows1, cols1, mat1, mat2, result);

cout << "Result of Addition:\n";

displayMatrix(rows1, cols1, result);

} else {

cout << "Addition not possible. Matrices must have the same dimensions.\n";

}

break;

case 2:

if (rows1 == rows2 && cols1 == cols2) {

subtractMatrices(rows1, cols1, mat1, mat2, result);

cout << "Result of Subtraction:\n";

displayMatrix(rows1, cols1, result);

} else {

cout << "Subtraction not possible. Matrices must have the same dimensions.\n";

}

break;

case 3:

if (cols1 == rows2) {

multiplyMatrices(rows1, cols1, mat1, rows2, cols2, mat2, result);

cout << "Result of Multiplication:\n";

displayMatrix(rows1, cols2, result);

} else {

cout << "Multiplication not possible. Columns of Matrix 1 must match rows of Matrix 2.\n";

}

break;

default:

cout << "Invalid choice.\n";

}

return 0;

}