Tema 6

NICOLETA RADU

// Fundamentele.Programarii.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include <iostream>

#include <vector>

#include <string>

#define dim 20

using namespace std;

void zeros(int yourMatrix[][dim], int r, int c)

{

for (size\_t i = 0; i < r; i++)

{

for (size\_t j = 0; j < c; j++)

{

yourMatrix[i][j] = 0;

}

}

}

void readMatrix(int yourMatrix[][dim], int &row, int &col)

{

// assigning rows and colummns

char userChoice{};

do

{

cout << "Number of rows: ";

cin >> row;

cout << "Number of colummns: ";

cin >> col;

} while ((row < 1 || row > dim) || col < 1 || col > dim);

cout << "Would you like to enter the values for the matrix? (Y - yes/N - no)" << endl;

cin >> userChoice;

if (userChoice == 'Y' || userChoice == 'y')

{

for (size\_t i = 0; i < row; i++)

{

for (size\_t j = 0; j < col; j++)

{

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

cin >> yourMatrix[i][j];

}

}

}

else

{

zeros(yourMatrix, row, col);

}

cout << endl;

}

void printMatrix(int yourMatrix[][dim], int row, int col)

{

for (size\_t i = 0; i < row; i++)

{

for (size\_t j = 0; j < col; j++)

{

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

}

cout << endl;

}

cout << endl;

}

int sumMatrixElems(int yourMatrix[][dim], int rows, int cols)

{

int sumOfElements{ 0 };

for (size\_t i = 0; i < rows; i++)

{

for (size\_t j = 0; j < cols; j++)

{

sumOfElements += yourMatrix[i][j];

}

}

cout << endl;

cout << "Sum of matrix elements is: " << sumOfElements << endl;

return sumOfElements;

}

int sumMatrixRow(int yourMatrix[][dim], int rows, int cols)

{

int sumOfLineElements{ 0 };

int rowToAdd;

cout << endl;

cout << "Enter row: ";

cin >> rowToAdd;

for (size\_t i = 0; i < cols; i++)

{

sumOfLineElements += yourMatrix[rowToAdd][i];

}

cout << endl;

cout << "Sum of matrix row " << rowToAdd << " elements is: " << sumOfLineElements << endl;

return sumOfLineElements;

}

int sumMatrixColummn(int yourMatrix[][dim], int rows, int cols)

{

int sumOfColElements{ 0 };

int colToAdd;

cout << endl;

cout << "Enter colummn: ";

cin >> colToAdd;

for (size\_t i = 0; i < rows; i++)

{

sumOfColElements += yourMatrix[i][colToAdd];

}

cout << endl;

cout << "Sum of matrix colummn " << colToAdd << " elements is : " << sumOfColElements << endl;

return sumOfColElements;

}

void sumMatrixDiag(int yourMatrix[][dim],int rows, int cols)

{

int sumDiag{ 0 }, sumSecondaryDiag{0};

for (size\_t i = 0; i < cols; i++)

{

sumSecondaryDiag = sumSecondaryDiag + yourMatrix[i][cols - i - 1];

}

for (size\_t i = 0; i < cols; i++)

{

sumDiag = sumDiag + yourMatrix[i][i];

}

cout << endl;

cout << "Sum of matrix main diagonal elements is : " << sumDiag << endl;

cout << "Sum of matrix secondary diagonal elements is : " << sumSecondaryDiag << endl;

}

void sumMatrixMainDiagTriangles(int yourMatrix[][dim],int rows, int cols)

{

// sum of upper/lower triangles of MAIN DIAGONAL

int sumUpper{ 0 }, sumLower{ 0 };

for (size\_t i = 0; i < rows; i++)

{

for (size\_t j = 0; j < cols; j++)

{

if (i <= j)

{

sumUpper += yourMatrix[i][j]; // sum of upper triangle

}

}

}

for (size\_t i = 0; i < rows; i++)

{

for (size\_t j = 0; j < cols; j++)

{

if (j <= i)

{

sumLower += yourMatrix[i][j]; // sum of lower triangle

}

}

}

cout << endl;

cout << "Sum of matrix upper triangle (main diagonal) elements is : " << sumUpper <<endl;

cout << "Sum of matrix lower triangle (main diagonal) elements is : " << sumLower<<endl;

}

void sumMatrixSecondDiagTriangles(int yourMatrix[][dim], int rows, int cols)

{

int upperTriangle{ 0 }, lowerTriangle{ 0 };

for (size\_t i = 0; i < cols - 1; i++)

{

for (size\_t j = 0; j < cols - 2 - i; j++)

{

cout << "Print i " << i << ", print j " << j << endl;

upperTriangle = upperTriangle + yourMatrix[i][j];

}

}

for (size\_t i = 0; i < rows; i++)

{

for (size\_t j = 0; j < cols - 1; j++)

{

lowerTriangle = lowerTriangle + yourMatrix[i][j];

}

}

cout << endl;

cout << "Sum of matrix upper triangle (secondary diagonal) elements is : " << upperTriangle << endl;

cout << "Sum of matrix lower triangle (secondary diagonal) elements is : " << lowerTriangle << endl;

}

void sumOf2Matrices(int matrix1[][dim],int matrix2[][dim], int finalMatrix[][dim] ,int size)

{

for (size\_t i = 0; i < size; i++)

{

for (size\_t j = 0; j < size; j++)

{

finalMatrix[i][j] += matrix1[i][j] + matrix2[i][j];

}

}

cout << endl;

cout << "New matrix is: " << endl;

printMatrix(finalMatrix, size, size);

}

void productOf2Matrices(int matrix1[][dim], int matrix2[][dim] ,int resultMatrix[][dim], int row1, int col1, int row2, int col2)

{

for (size\_t i = 0; i < row1; i++)

{

for (size\_t j = 0; j < col2; j++)

{

for (size\_t k = 0; k < row2; k++)

{

resultMatrix[i][j] += matrix1[i][k] + matrix2[k][j];

}

}

}

printMatrix(resultMatrix, row1, col2);

}

int main()

{

int myMatrix[dim][dim];

int secondMatrix[dim][dim];

int thirdMatrix[dim][dim];

int rowMyMatrix{ 0 }, colMyMatrix{ 0 };

cout << "Matrix 1" << endl;

readMatrix(myMatrix, rowMyMatrix, colMyMatrix);

printMatrix(myMatrix, rowMyMatrix, colMyMatrix);

cout << "Matrix 2" << endl;

readMatrix(secondMatrix, rowMyMatrix, colMyMatrix);

printMatrix(secondMatrix, rowMyMatrix, colMyMatrix);

zeros(thirdMatrix, rowMyMatrix, colMyMatrix);

char userChoice{};

// MENU

do

{

cout << endl;

cout << "A. Sum of all matrix elements" << endl;

cout << "B. Sum of matrix specific row elements" << endl;

cout << "C. Sum of matrix specific column elements" << endl;

cout << "D. Sum of matrix main/secondary diagonal" << endl;

cout << "E. Sum of matrix main diagonal upper/lower triangle" << endl;

cout << "F. Sum of matrix secondary diagonal upper/lower triangle" << endl;

cout << "G. Sum of 2 matrices " << endl;

cout << "H. Product of 2 matrices " << endl;

cout << "Q. Quit Program " << endl;

cout << endl;

cout << "Choose an option: ";

cin >> userChoice;

if (userChoice == 'A' || userChoice == 'a')

{

cout << "Matrix 1" << endl;

sumMatrixElems(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'B' || userChoice == 'b')

{

cout << "Matrix 1" << endl;

sumMatrixRow(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'C' || userChoice == 'c')

{

cout << "Matrix 1" << endl;

sumMatrixColummn(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'D' || userChoice == 'd')

{

cout << "Matrix 1" << endl;

sumMatrixDiag(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'E' || userChoice == 'e')

{

cout << "Matrix 1" << endl;

sumMatrixMainDiagTriangles(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'F' || userChoice == 'f')

{

cout << "Matrix 1" << endl;

sumMatrixSecondDiagTriangles(myMatrix, rowMyMatrix, colMyMatrix);

}

if (userChoice == 'G' || userChoice == 'g')

{

sumOf2Matrices(myMatrix, secondMatrix, thirdMatrix, rowMyMatrix);

}

if (userChoice == 'H' || userChoice == 'h')

{

productOf2Matrices(myMatrix, secondMatrix, thirdMatrix, rowMyMatrix, colMyMatrix, rowMyMatrix, colMyMatrix);

}

} while (userChoice != 'Q' && userChoice != 'q');

return 0;

}