**Program Code**

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

// A program that declares and initializes a two-dimensional array of floating point numbers (use an initializer list)

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

// Declare ROWS and COLUMNS as constants

const int ROWS = 5, COLS = 3;

double getMinInRow(double numbers[ROWS][COLS], int row);

double getMaxInRow(double numbers[ROWS][COLS], int row);

double calcColTot(double numbers[ROWS][COLS], int col);

double calcRowTot(double numbers[ROWS][COLS], int row);

double calcTot(double numbers[ROWS][COLS]);

double calcAvg(double numbers[ROWS][COLS]);

int main() {

// Display array as rows and columns and input numbers as doubles

double numbers[ROWS][COLS] = { {16.1, 2.8, 7.2},

{11.4, 5.9, 1.5},

{7.9, 3.6, 19.3},

{4.1, 14.4, 20.4},

{5.0, 7.2, 13.9} };

double rowTot = 0, colTot = 0, rowHighest, rowlowest;

double tot = calcTot(numbers);

double average = calcAvg(numbers);

//Display results to user as array elements in for statements

cout << "Total of the array elements :" << tot << endl;

cout << "Average elements:" << average << endl;

for (int i = 0; i < ROWS; i++)

{

rowTot = calcRowTot(numbers, i);

cout << "Row " << i << " total:" << rowTot << endl;

}

for (int i = 0; i < COLS; i++)

{

colTot = calcColTot(numbers, i);

cout << "Column " << i << " total:" << colTot << endl;

}

for (int i = 0; i < ROWS; i++)

{

rowHighest = getMaxInRow(numbers, i);

cout << "Row#" << (i) << " Highest :" << rowHighest << endl;

}

for (int i = 0; i < ROWS; i++)

{

rowlowest = getMinInRow(numbers, i);

cout << "Row#" << (i) << " Lowest :" << rowlowest << endl;

}

return 0;

}

// Declare getMinInRow as array for minimum

double getMinInRow(double numbers[ROWS][COLS], int row) {

double min = numbers[row][0];

for (int i = 0; i < COLS; i++)

{

if (min > numbers[row][i])

min = numbers[row][i];

}

return min;

}

// Declare getMaxInRow as array for maximum

double getMaxInRow(double numbers[ROWS][COLS], int row) {

double max;

max = numbers[row][0];

for (int i = 0; i < COLS; i++)

{

if (max < numbers[row][i])

max = numbers[row][i];

}

return max;

}

// Declare double calcColTot as array for rows and calculate total

double calcColTot(double numbers[ROWS][COLS], int col) {

double tot = 0;

for (int i = 0; i < ROWS; i++)

{

tot += numbers[i][col];

}

return tot;

}

// Declare calcRowTot as array for columns and calculate total

double calcRowTot(double numbers[ROWS][COLS], int row) {

double tot = 0;

for (int i = 0; i < COLS; i++)

{

tot += numbers[row][i];

}

return tot;

}

// Declare calcAvg as double

double calcAvg(double numbers[ROWS][COLS]) {

return ((double)(calcTot(numbers))) / (ROWS + COLS);

}

// Declare calcTot as doubles array for total

double calcTot(double numbers[ROWS][COLS]) {

double tot = 0;

for (int i = 0; i < ROWS; i++)

{

for (int j = 0; j < COLS; j++)

{

tot += numbers[i][j];

}

}

return tot;

}