ЗВІТ

про виконання лабораторної роботи № 7.03

«Опрацювання динамічних багатовимірних масивів» з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

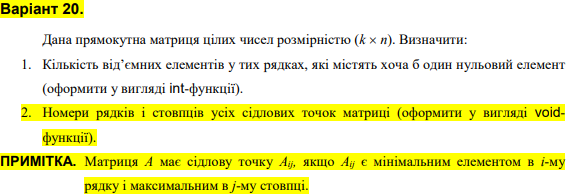
Мілянця Мар`яна

# Мета роботи:

Навчитися опрацьовувати динамічні двовимірні та багатовимірні масиви.

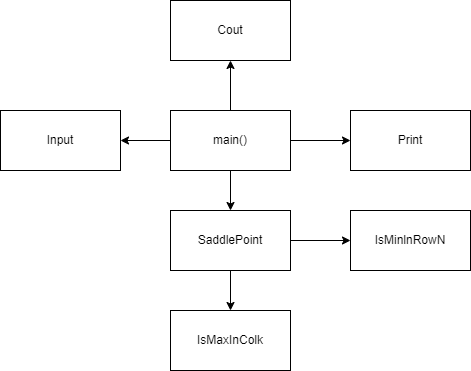
# Умова завдання:

Розмірності масивів слід задати за допомогою змінних а їх значення – вводити з клавіатури під час виконання програми (масиви мають бути динамічними). Усі необхідні дані мають передаватися функціям як параметри; всі величини, які використовуються лише всередині функцій, мають бути описані як локальні. Використання глобальних змінних у функціях не допускається. Виведення результатів роботи функцій має виконуватися в головній функції. Кожна функція має виконувати лише одну роль, і ця роль має бути відображена у назві функції. Рекурсивний та ітераційний способи – це 2 різні проекти, для яких потрібно 2 різних unit-тести і 2 різних звіти. «Функція, яка повертає / обчислює / шукає ...» – має не виводити ці значення, а повернути їх у місце виклику як результат функції або як відповідний вихідний параметр. Дана прямокутна матриця цілих чисел розмірністю (k  n). Визначити: 1. Кількість додатних елементів в тих стовпцях, які містять хоч б один нульовий елемент (оформити у вигляді int-функції). 2. Номери рядків та стовпців всіх сідлових точок матриці (оформити у вигляді voidфункції). ПРИМІТКА. Матриця А має сідлову точку Аij, якщо Аij є максимальним елементом в i-му рядку та мінімальним в j-му стовпці.



# Структурна схема програми:

**Ітераційний спосіб:**



# Текст програми:

#include <iostream> #include <iomanip> using namespace std;

void Input(int\*\* a, const int rowCount, const int colCount); void Print(int\*\* a, const int rowCount, const int colCount);

int Count(int\*\* a, const int rowCount, const int colCount, int& count); void SaddlePoint(int\*\* a, const int rowCount, const int colCount);

bool IsMinInRowN(int\*\* a, const int n, const int k, const int colCount); bool IsMaxInColK(int\*\* a, const int n, const int k, const int rowCount); int main()

{

int rowCount, colCount;

cout << "rowCount = "; cin >> rowCount; cout << "colCount = "; cin >> colCount; int\*\* a = new int\* [rowCount];

for (int i = 0; i < rowCount; i++) a[i] = new int[colCount];

Input(a, rowCount, colCount); Print(a, rowCount, colCount); int count = 0;

if (Count(a, rowCount, colCount, count))

else

cout << "count = " << count << endl;

cout << "there are no zero elements" << endl;

cout << endl;

SaddlePoint(a, rowCount, colCount); cout << endl;

for (int i = 0; i < rowCount; i++) delete[] a[i];

delete[] a; return 0;

}

void Input(int\*\* a, const int rowCount, const int colCount)

{

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

{

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

{

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

cin >> a[i][j];

}

cout << endl;

}

}

void Print(int\*\* a, const int rowCount, const int colCount)

{

cout << endl;

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

{

for (int j = 0; j < colCount; j++) cout << setw(4) << a[i][j];

cout << endl;

}

cout << endl;

}

int Count(int\*\* a, const int rowCount, const int colCount, int& count)

{

bool result = false; count = 0;

int k\_zero;

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

{

k\_zero = 0;

for (int j = 0; j < rowCount; j++) if (a[i][j] == 0)

{

result = true; k\_zero++; break;

}

if (k\_zero > 0)

{

for (int j = 0; j < colCount; j++) if (a[i][j] < 0)

count++;

}

}

return result;

}

bool IsMinInRowN(int\*\* a, const int n, const int k, const int colCount)

{

for (int j = 0; j < colCount; j++) if (a[n][j] < a[n][k])

{

return false;

}

return true;

}

bool IsMaxInColK(int\*\* a, const int n, const int k, const int rowCount)

{

for (int i = 0; i < rowCount; i++) if (a[i][k] > a[n][k])

{

return false;

}

return true;

}

void SaddlePoint(int\*\* a, const int rowCount, const int colCount)

{

cout << "Saddle points: min in row & max in col" << endl;

cout << setw(4) << "No" << setw(6) << "Row" << setw(6) << "Col" << endl; int No = 0;

for (int n = 0; n < rowCount; n++)

for (int k = 0; k < colCount; k++)

if (IsMinInRowN(a, n, k, colCount) && IsMaxInColK(a, n, k, rowCount))

{

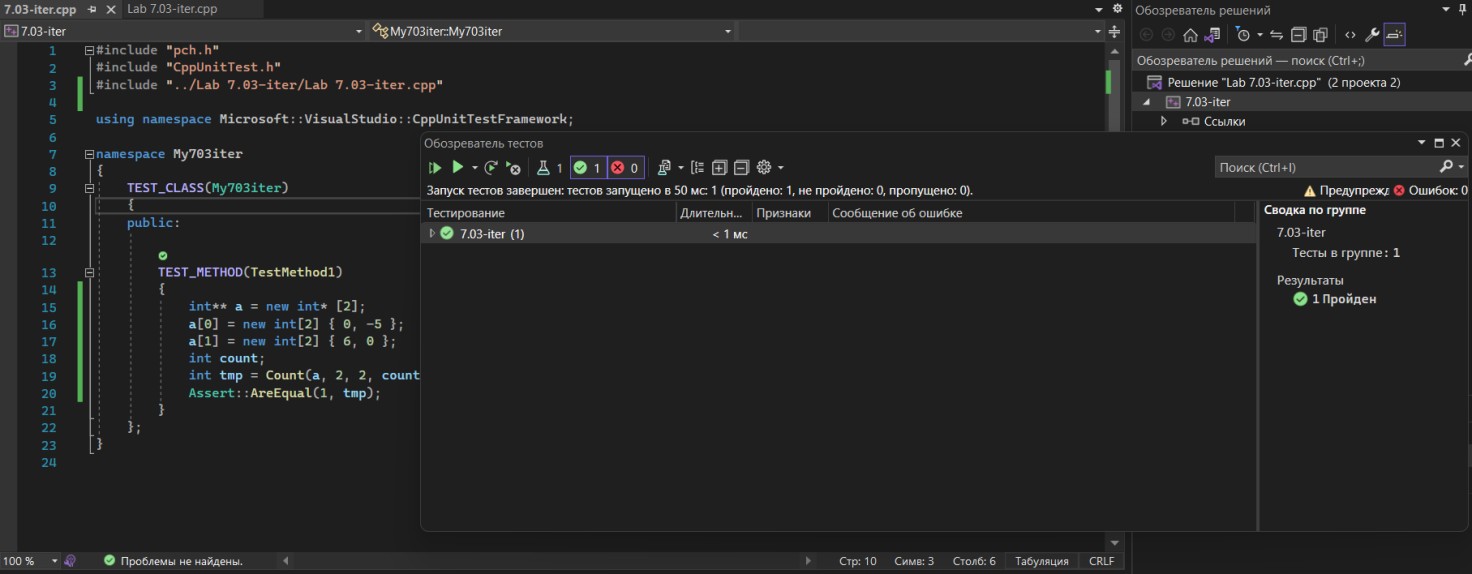
endl;

}

cout << setw(4) << ++No << setw(6) << n << setw(6) << k <<

}

# Результати unit-тесту:



**Текст unit-тесту:**

#include "pch.h" #include "CppUnitTest.h"

#include "../Lab 7.03-iter/Lab 7.03-iter.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace My703iter

{

TEST\_CLASS(My703iter)

{

public:

TEST\_METHOD(TestMethod1)

{

int\*\* a = new int\* [2];

a[0] = new int[2] { 0, -5 };

a[1] = new int[2] { 6, 0 }; int count;

int tmp = Count(a, 2, 2, count); Assert::AreEqual(1, tmp);

}

};

}

**Посилання**: <https://github.com/milianets20/Lab-7.2-7.3>

**Висновок**: виконуючи лабораторну роботу я навчився опрацьовувати динамічні двовимірні та багатовимірні масиви.