**мІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**нАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «лЬВІВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра ІСМ**

|  |
| --- |
|  |



**ЗВІТ**

**Про виконання лабораторної роботи № 7.3**

**« Опрацювання динамічних багатовимірних масивів»**

**З дисципліни**

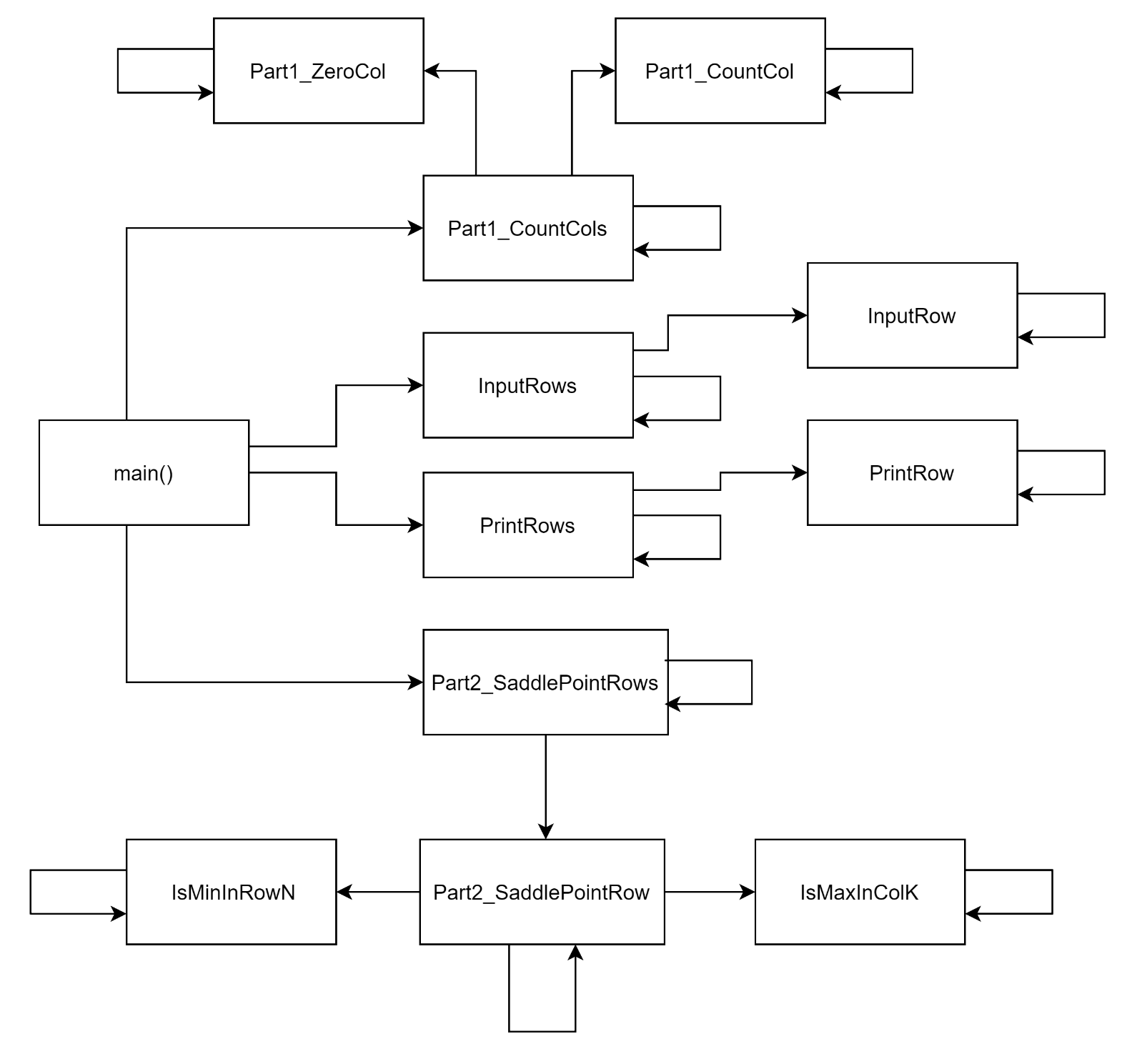
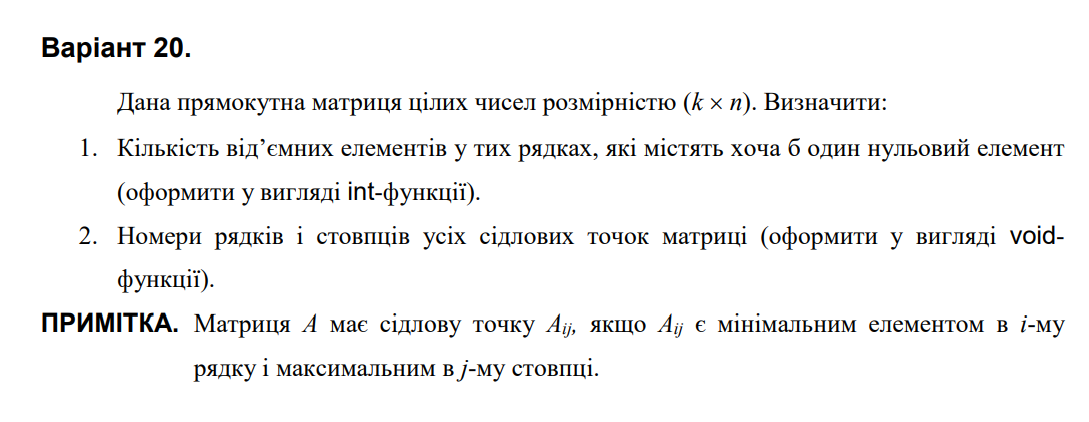
**«Алгоритмізація та програмування»**

**Студентки групи ІТ-11**

**Проців Роксолани Василівни**

**Прийняв доцент**

**Григорович В. Г.**

**Умова завдання:**

#include <iostream>

#include <iomanip>

using namespace std;

void PrintRow(int\*\* a, const int rowNo, const int colCount, int colNo)

{

cout << setw(4) << a[rowNo][colNo];

if (colNo < colCount - 1)

PrintRow(a, rowNo, colCount, colNo + 1);

else

cout << endl;

}

void PrintRows(int\*\* a, const int rowCount, const int colCount, int rowNo)

{

PrintRow(a, rowNo, colCount, 0);

if (rowNo < rowCount - 1)

PrintRows(a, rowCount, colCount, rowNo + 1);

else

cout << endl;

}

void InputRow(int\*\* a, const int rowNo, const int colCount, int colNo)

{

cout << "a[" << rowNo << "][" << colNo << "] = ";

cin >> a[rowNo][colNo];

if (colNo < colCount - 1)

InputRow(a, rowNo, colCount, colNo + 1);

else

cout << endl;

}

void InputRows(int\*\* a, const int rowCount, const int colCount, int rowNo)

{

InputRow(a, rowNo, colCount, 0);

if (rowNo < rowCount - 1)

InputRows(a, rowCount, colCount, rowNo + 1);

else

cout << endl;

}

void Part1\_ZeroCol(int\*\* a, const int colCount, bool& result, int colNo, int rowNo, int& k\_zero)

{

if (a[rowNo][colNo] == 0)

{

result = true;

k\_zero++;

}

else

if (colNo < colCount - 1)

Part1\_ZeroCol(a, colCount, result, colNo + 1, rowNo, k\_zero);

}

void Part1\_CountCol(int\*\* a, const int colCount, int& count, int colNo, int rowNo)

{

if (a[rowNo][colNo] < 0)

count++;

if (colNo < colCount - 1)

Part1\_CountCol(a, colCount, count, colNo + 1, rowNo);

}

int Part1\_CountCols(int\*\* a, const int rowCount, const int colCount, int& count, bool& result, int rowNo)

{

int k\_zero = 0;

Part1\_ZeroCol(a, colCount, result, 0, rowNo, k\_zero);

if (k\_zero > 0)

Part1\_CountCol(a, colCount, count, 0, rowNo);

if (rowNo < rowCount - 1)

return Part1\_CountCols(a, rowCount, colCount, count, result, rowNo + 1);

}

bool IsMinInRowN(int\*\* a, const int rowNo, const int min, const int colCount, int colNo)

{

if (a[rowNo][colNo] < min)

return false;

if (colNo < colCount - 1)

return IsMinInRowN(a, rowNo, min, colCount, colNo + 1);

else

return true;

}

bool IsMaxInColK(int\*\* a, const int colNo, const int max, const int rowCount, int rowNo)

{

if (a[rowNo][colNo] > max)

return false;

if (rowNo < rowCount - 1)

return IsMaxInColK(a, colNo, max, rowCount, rowNo + 1);

else

return true;

}

void Part2\_SaddlePointRow(int\*\* a, const int rowCount, const int colCount, int& No, const int rowNo, int colNo)

{

if (IsMinInRowN(a, rowNo, a[rowNo][colNo], colCount, 0) &&

IsMaxInColK(a, colNo, a[rowNo][colNo], rowCount, 0))

{

cout << setw(4) << ++No << setw(6) << rowNo << setw(6) << colNo << endl;

}

if (colNo < colCount - 1)

Part2\_SaddlePointRow(a, rowCount, colCount, No, rowNo, colNo + 1);

}

void Part2\_SaddlePointRows(int\*\* a, const int rowCount, const int colCount, int& No, int rowNo)

{

Part2\_SaddlePointRow(a, rowCount, colCount, No, rowNo, 0);

if (rowNo < rowCount - 1)

Part2\_SaddlePointRows(a, rowCount, colCount, No, rowNo + 1);

}

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];

InputRows(a, rowCount, colCount, 0);

PrintRows(a, rowCount, colCount, 0);

int count = 0;

bool result = false;

Part1\_CountCols(a, rowCount, colCount, count, result, 0);

if (result)

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

else

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

cout << endl;

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

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

int No = 0;

Part2\_SaddlePointRows(a, rowCount, colCount, No, 0);

cout << endl;

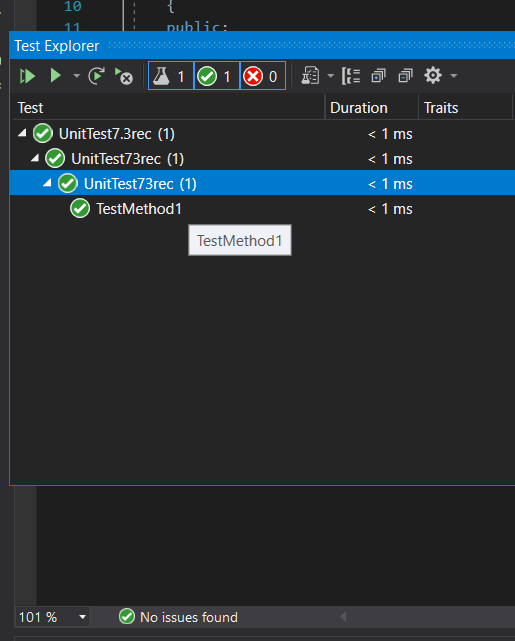
for (int i = 0; i < rowCount; i++)

delete[] a[i];

delete[] a;

return 0;

}

****

#include "pch.h"

#include "CppUnitTest.h"

#include "../Lab-7.3rec/Lab-7.3rec.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest73rec

{

TEST\_CLASS(UnitTest73rec)

{

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;

bool result;

int tmp = Part1\_CountCols(a, 2, 2, count, result, 0);

Assert::AreEqual(1, tmp);

}

};

}

**Посилання:** <https://github.com/fxxwol/lab-7.git>

**Висновок**: я навчилась опрацьовувати динамічні двовимірні та багатовимірні масиви