ЗВІТ

про виконання лабораторної роботи № 7.1.іт

«Пошук заданих елементів та впорядкування рядків / стовпчиків матриці»

з дисципліни

«Алгоритмізація та програмування»

студентки групи РІ-11

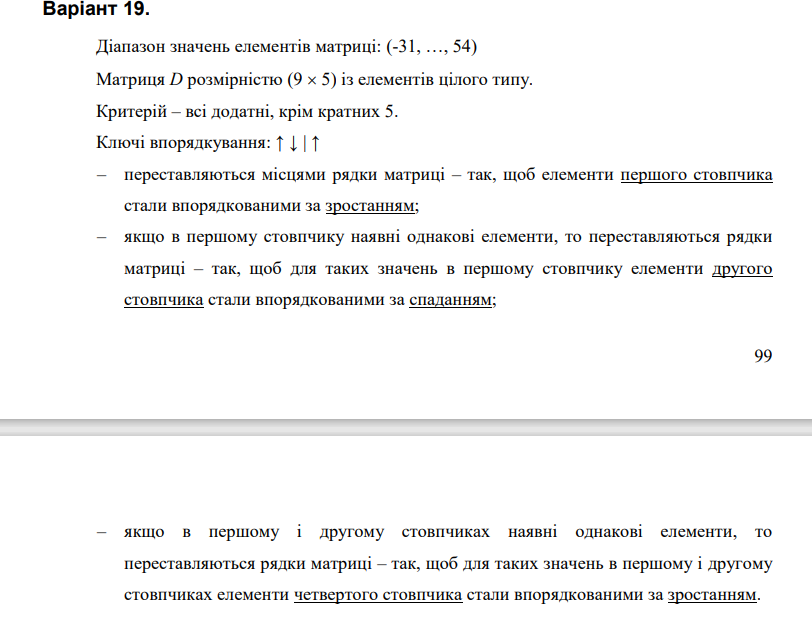
Пятницької Вікторії Володимирівни

**Мета роботи**

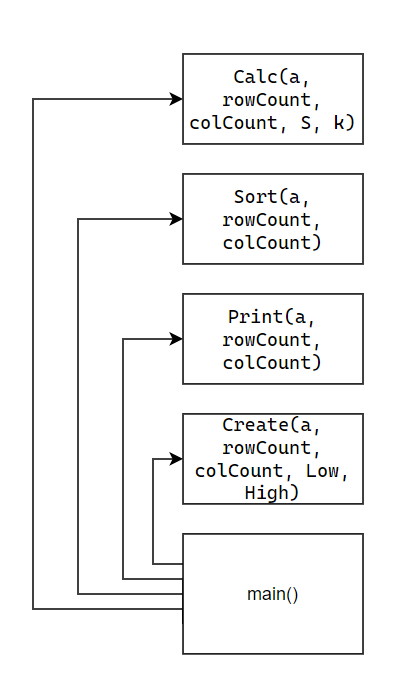
Навчитися обчислювати кількість, суму заданих елементів матриці; а також

впорядковувати матрицю, переставляючи місцями її рядки / стовпчики.

**Умова завдання:**

****

**Структурна схема програми:**

****

**Текст програми:**

#include <iostream>

#include <iomanip>

#include <time.h>

using namespace std;

void Create(int\*\* a, const int rowCount, const int colCount, const int Low,

const int High);

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

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

void Change(int\*\* a, const int row1, const int row2, const int colCount);

void Calc(int\*\* a, const int rowCount, const int colCount, int& S, int& k);

int main()

{

srand((unsigned)time(NULL));

int Low = -31;

int High = 54;

int rowCount = 9;

int colCount = 5;

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

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

a[i] = new int[colCount];

Create(a, rowCount, colCount, Low, High);

Print(a, rowCount, colCount);

Sort(a, rowCount, colCount);

Print(a, rowCount, colCount);

int S = 0;

int k = 0;

Calc(a, rowCount, colCount, S, k);

cout << "S = " << S << endl;

cout << "k = " << k << endl;

Print(a, rowCount, colCount);

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

delete[] a[i];

delete[] a;

return 0;

}

void Create(int\*\* a, const int rowCount, const int colCount, const int Low,

const int High)

{

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

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

a[i][j] = Low + rand() % (High - Low + 1);

}

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;

}

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

{

for (int i0 = 0; i0 < rowCount - 1; i0++)

for (int i1 = 0; i1 < rowCount - i0 - 1; i1++)

if ((a[i1][0] > a[i1 + 1][0])

||

(a[i1][0] == a[i1 + 1][0] &&

a[i1][1] > a[i1 + 1][1])

||

(a[i1][0] == a[i1 + 1][0] &&

a[i1][1] == a[i1 + 1][1] &&

a[i1][3] < a[i1 + 1][3]))

Change(a, i1, i1 + 1, colCount);

}

void Change(int\*\* a, const int row1, const int row2, const int colCount)

{

int tmp;

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

{

tmp = a[row1][j];

a[row1][j] = a[row2][j];

a[row2][j] = tmp;

}

}

void Calc(int\*\* a, const int rowCount, const int colCount, int& S, int& k)

{

S = 0;

k = 0;

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

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

if (a[i][j] > 0 && !(a[i][j] % 5 == 0))

{

S += a[i][j];

k++;

a[i][j] = 0;

}

}

**Посилання на git-репозиторій з проектом:**

**Результати unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../lab\_7.1.іт/lab\_7.1.іт.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest71іт

{

TEST\_CLASS(UnitTest71іт)

{

public:

TEST\_METHOD(TestMethod1)

{

int count = 0;

int s = 0;

int result;

const int rowCOUNT = 4;

const int colCOUNT = 4;

int a[rowCOUNT][colCOUNT] = {

{1, 5, 8, 10},

{2, 15, 7, 9},

{-1, -2, 4, 0},

{-3, 4, 6, 9}

};

int\* pa[4] = { a[0], a[1], a[2], a[3] };

Calc(pa, rowCOUNT, colCOUNT, s, count);

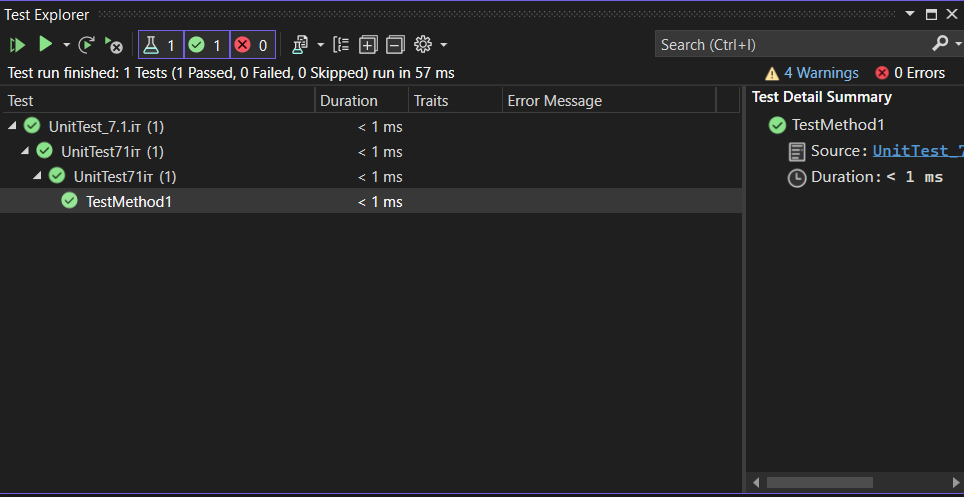
result = count;

Assert::AreEqual(9, result);

}

};

}

****

**Висновки:**

Отже, виконавши цю лабораторну роботу я навчилася обчислювати кількість, суму заданих елементів матриці; а також впорядковувати матрицю, переставляючи місцями її рядки / стовпчики.