ЗВІТ

про виконання лабораторної роботи № 7.1

«Пошук заданих елементів та впорядкування рядків / стовпчиків матриці»

з дисципліни

«Алгоритмізація та програмування»

студентки групи РІ-11

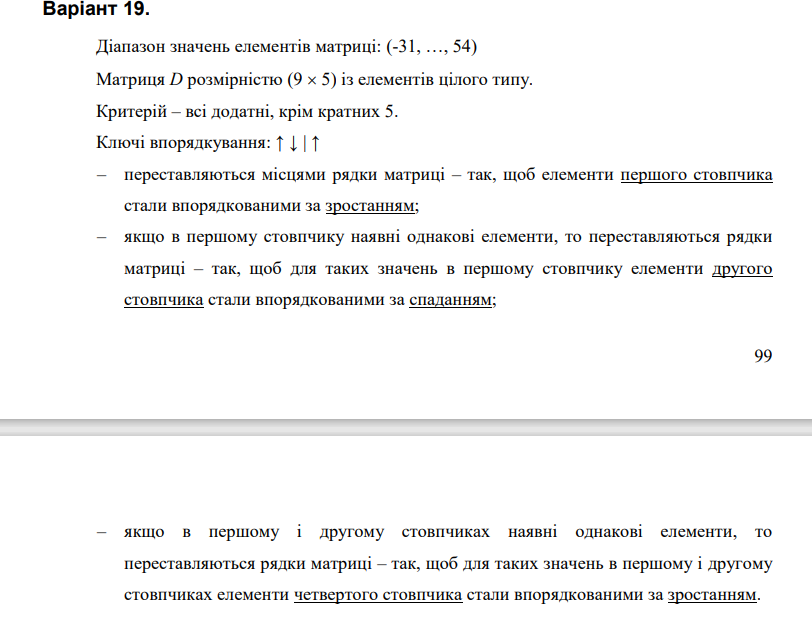
Пятницької Вікторії Володимирівни

**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, int i, int j)

{

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

if (j < colCount - 1)

Create(a, rowCount, colCount, Low, High, i, j + 1);

else

if (i < rowCount - 1)

Create(a, rowCount, colCount, Low, High, i + 1, 0);

}

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

{

cout << setw(4) << a[i][j];

if (j < colCount - 1)

Print(a, rowCount, colCount, i, j + 1);

else

if (i < rowCount - 1)

{

cout << endl;

Print(a, rowCount, colCount, i + 1, 0);

}

else

cout << endl << endl;

}

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

{

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

{

S += a[i][j];

k++;

a[i][j] = 0;

}

if (j < colCount - 1)

Calc(a, rowCount, colCount, S, k, i, j + 1);

else

if (i < rowCount - 1)

Calc(a, rowCount, colCount, S, k, i + 1, 0);

}

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

{

int tmp = a[row1][j];

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

a[row2][j] = tmp;

if (j < colCount - 1)

Change(a, row1, row2, colCount, j + 1);

}

void Sort(int\*\* a, const int rowCount, const int colCount, int j, int i0, int 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, j);

if (i1 < rowCount - i0 - 1)

Sort(a, rowCount, colCount, j, i0, i1 + 1);

else

if (i0 < rowCount - 1)

Sort(a, rowCount, colCount, j, i0 + 1, 0);

}

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, 0, 0);

cout << endl;

Print(a, rowCount, colCount, 0, 0);

Sort(a, rowCount, colCount, 0, 1, 1);

Print(a, rowCount, colCount, 0, 0);

int S = 0, k = 0;

Calc(a, rowCount, colCount, S, k, 0, 0);

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

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

Print(a, rowCount, colCount, 0, 0);

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

delete[] a[i];

delete[] a;

return 0;

}

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

**Результати unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

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

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest71rec

{

TEST\_CLASS(UnitTest71rec)

{

public:

TEST\_METHOD(TestMethod1)

{

int count = 0;

int s = 0;

int k;

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, k, 0, 0);

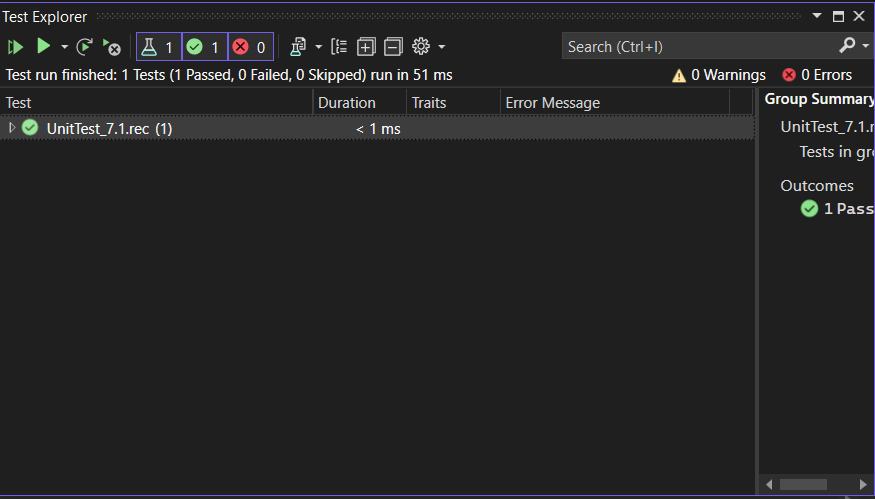
result = count;

Assert::AreEqual(0, result);

}

};

}

****

**Висновки:**

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