ЗВІТ

про виконання лабораторної роботи № <7.1>

*«Пошук заданих елементів та впорядкування рядків / стовпчиків матриці»*

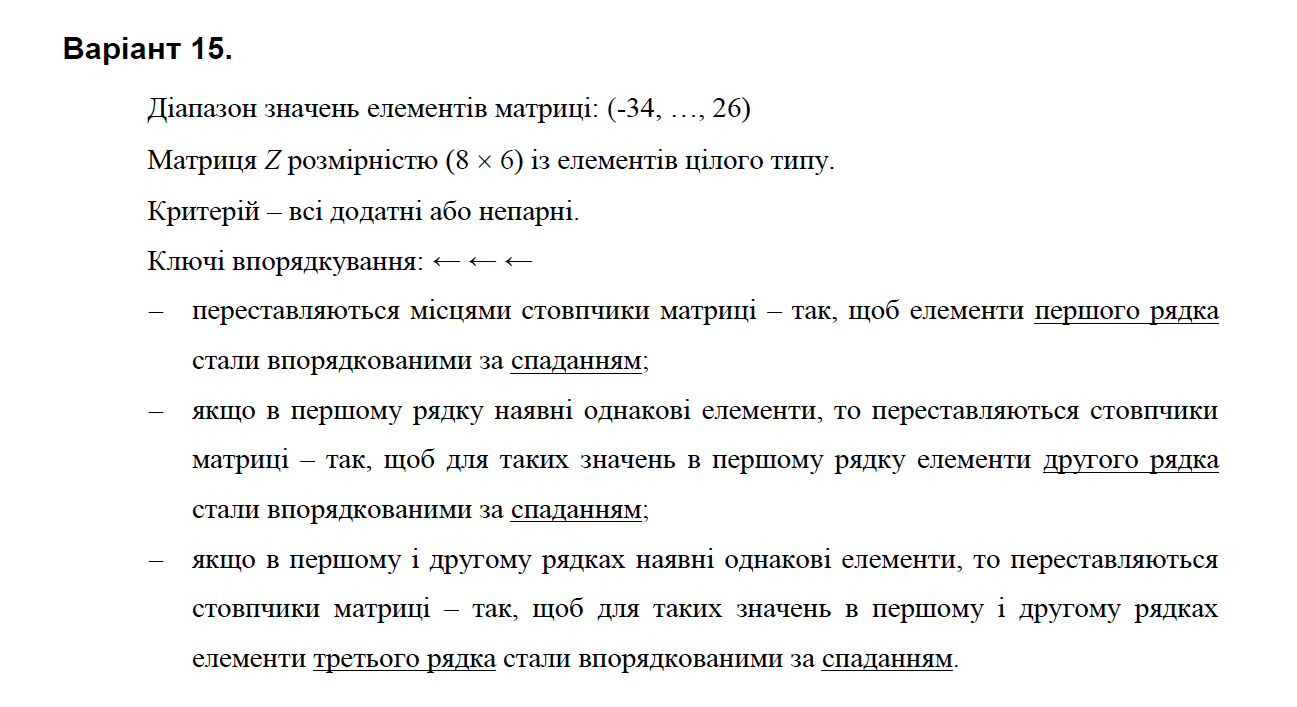
з дисципліни

«Алгоритмізація та програмування»

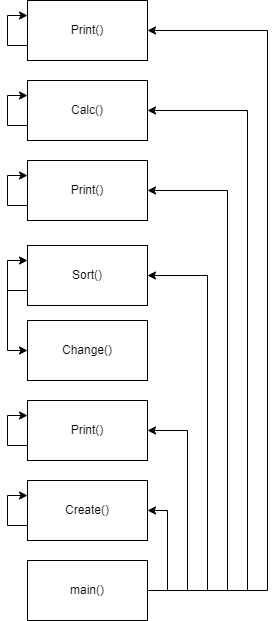
студента групи ІК-11

< *Снігура Стефана Андрійовича* >

**Умова завдання:**

****

**Структурна схема програми**

****

**Текст програми:**

#include <iostream>

#include <iomanip>

#include <time.h>

using namespace std;

void Create(int\*\* Z, const int rowCount, const int colCount,

const int Low, const int High, int i, int j)

{

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

if (j < colCount - 1)

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

else

if (i < rowCount - 1)

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

}

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

{

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

if (j < colCount - 1)

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

else

if (i < rowCount - 1)

{

cout << endl;

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

}

else

cout << endl << endl;

}

void Change(int\*\* Z, const int col1, const int col2, const int rowCount, int j)

{

int tmp;

if (j < rowCount) {

tmp = Z[j][col1];

Z[j][col1] = Z[j][col2];

Z[j][col2] = tmp;

Change(Z, col1, col2, rowCount, j + 1);

}

}

void Sort(int\*\* Z, const int rowCount, const int colCount, int i0, int i1)

{

if (i0 < colCount) {

if (i1 < colCount - i0 - 1) {

if ((Z[0][i1] < Z[0][i1 + 1])

||

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

Z[1][i1] < Z[1][i1 + 1])

||

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

Z[1][i1] == Z[1][i1 + 1] &&

Z[2][i1] < Z[2][i1 + 1]))

Change(Z, i1, i1 + 1, rowCount, 0);

Sort(Z, rowCount, colCount, i0, i1 + 1);

}

Sort(Z, rowCount, colCount, i0 +1, i1);

}

}

double Calc(int\*\* Z, const int rowCount, const int colCount, int& S, int& k, int j, int i)

{

if (i < rowCount) {

if (j < colCount) {

if (Z[i][j] % 2 != 0 || (Z[i][j] > 0))

{

S += Z[i][j];

k++;

Z[i][j] = 0;

}

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

}

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

}

else

return 0;

}

int main() {

srand((unsigned)time(NULL));

int Low = -34;

int High = 26;

int rowCount = 8;

int colCount = 6;

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

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

a[i] = new int[colCount];

int S = 0;

int k = 0;

Create(a, rowCount, colCount, Low, High, 0, 0);

cout << endl;

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

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

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

cout << endl;

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

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

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

cout << endl;

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

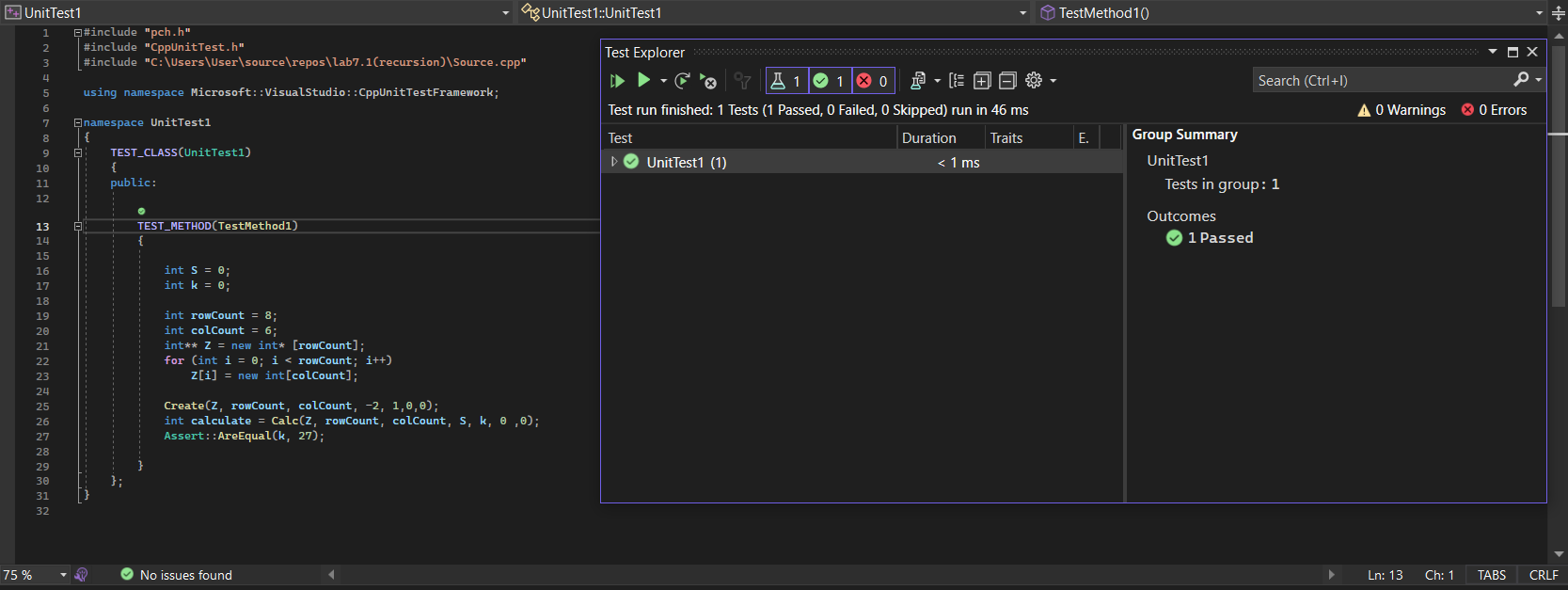
for (int i = 0; i < rowCount; i++)

delete[] a[i];

delete[] a;

}

**UNIT-test:**

****

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

<https://github.com/BigTrouble-Git/ashtray.git>

**Висновки**:

Виконавши цю лабораторну роботу я навчився обчислювати кількість, суму заданих елементів матриці, а також впорядковувати матрицю, переставляючи місцями її рядки / стовпчики.