ЗВІТ

про виконання лабораторної роботи № <7.1>

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

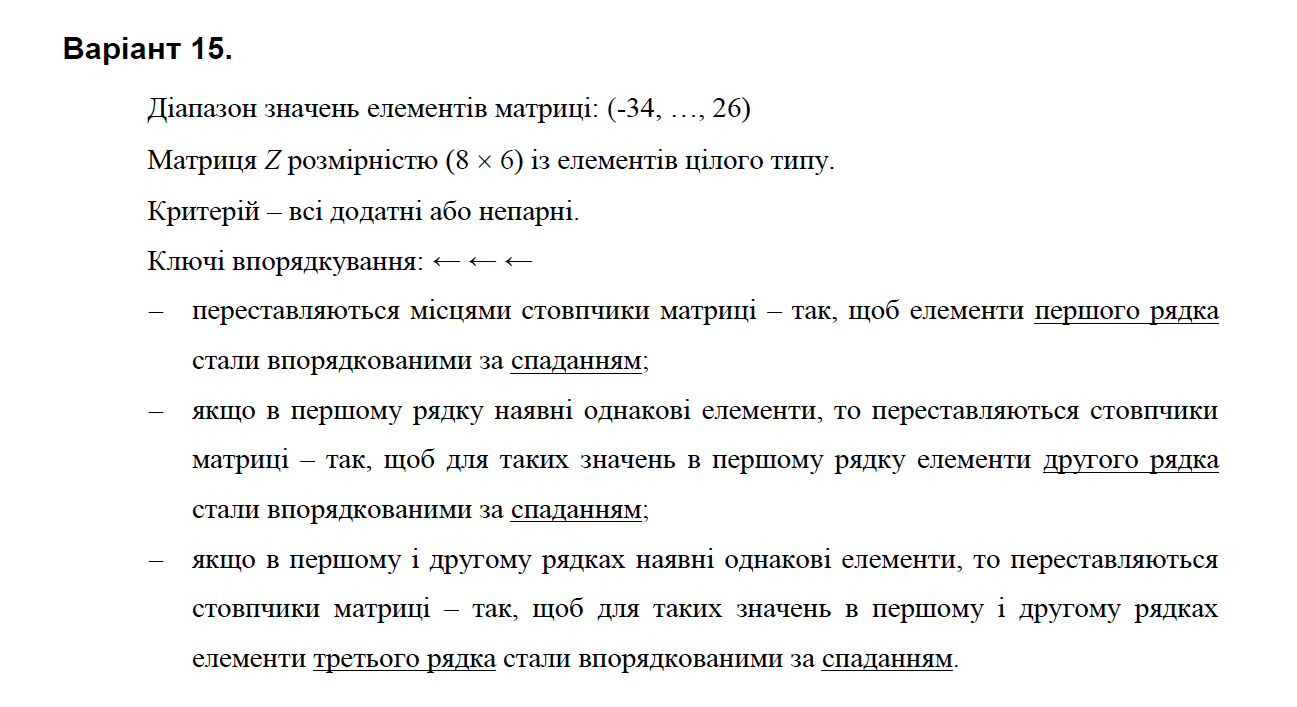
з дисципліни

«Алгоритмізація та програмування»

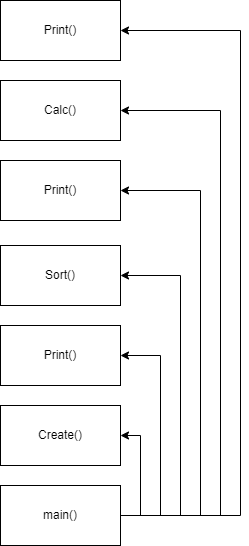
студента групи ІК-11

< *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);

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

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

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

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

int main()

{

srand((unsigned)time(NULL));

int Low = -34;

int High = 26;

int rowCount = 8;

int colCount = 6;

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

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

Z[i] = new int[colCount];

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

Print(Z, rowCount, colCount);

Sort(Z, rowCount, colCount);

Print(Z, rowCount, colCount);

int S = 0;

int k = 0;

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

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

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

Print(Z, rowCount, colCount);

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

delete[] Z[i];

delete[] Z;

return 0;

}

void Create(int\*\* Z, 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++)

Z[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\*\* Z, const int rowCount, const int colCount)

{

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

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

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);

}

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

{

int tmp;

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

{

tmp = Z[j][col1];

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

Z[j][col2] = tmp;

}

}

void Calc(int\*\* Z, 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 (Z[i][j] % 2 != 0 || (Z[i][j] > 0))

{

S += Z[i][j];

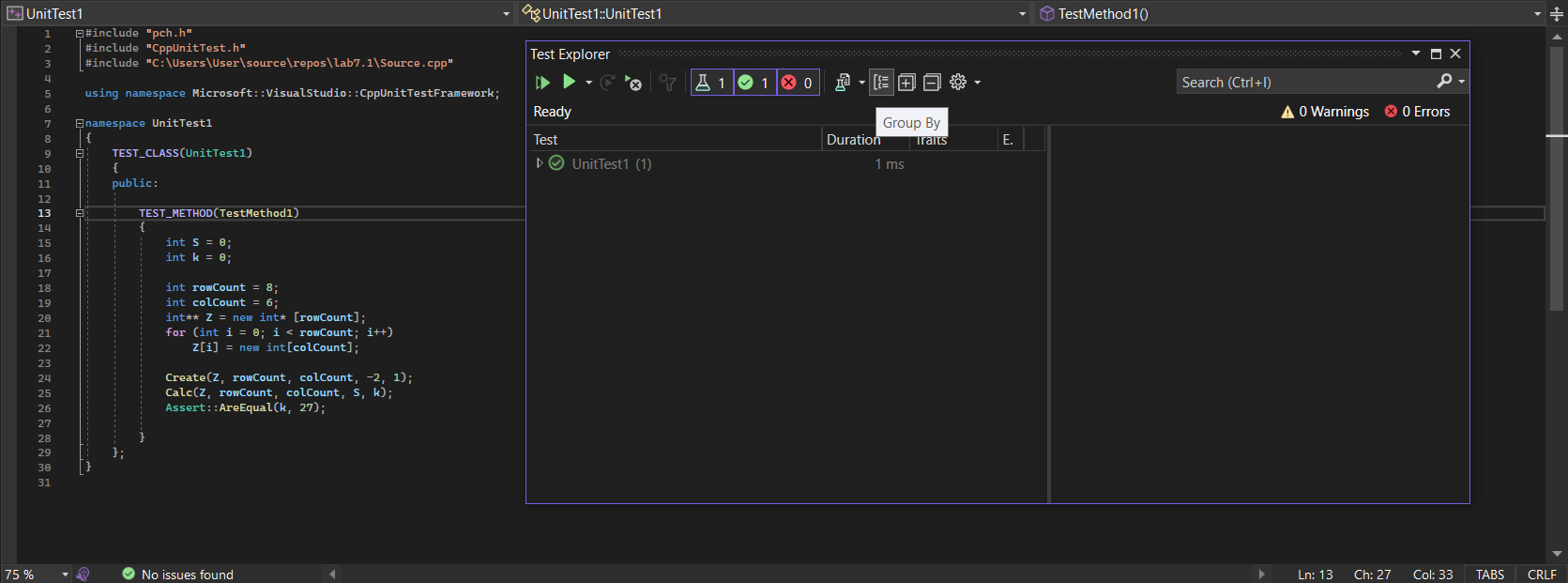
k++;

Z[i][j] = 0;

}

}

**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.