**ЗВІТ**

**до лабораторної роботи № < 7.1 >**

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

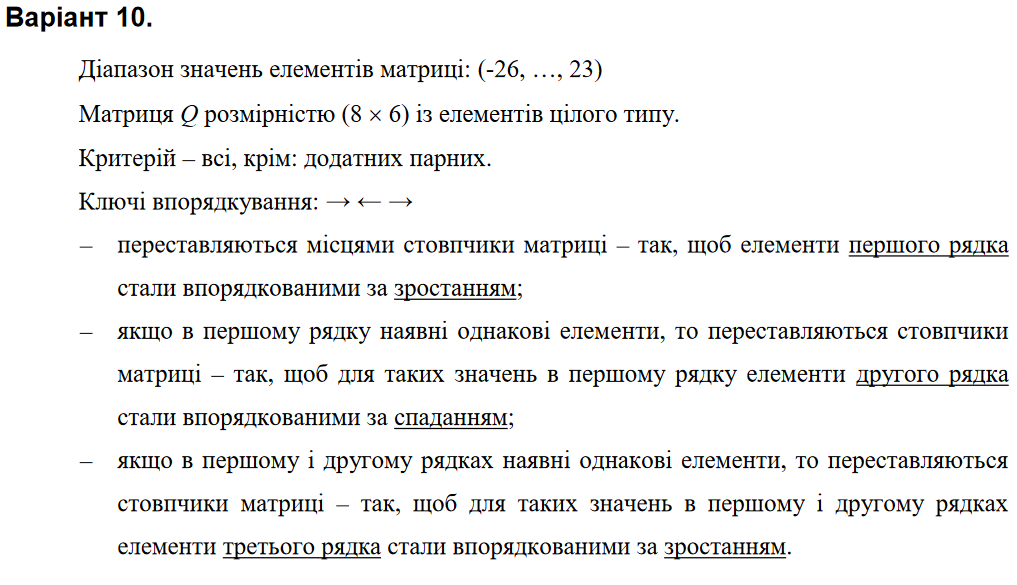
**з дисципліни**

**«Алгоритмізація та програмування»**

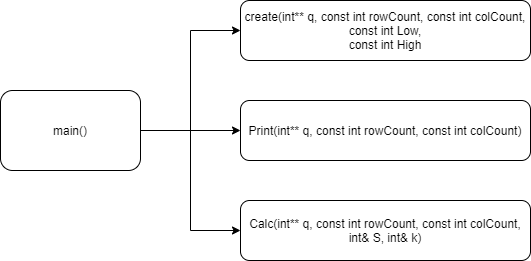
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**

****

**Текст програми:**

// 7.1.cpp

// < Горанов Анастас >

// Лабораторна робота № 7.1

// Пошук заданих елементів та впорядкування рядків / стовпчиків матриці

// Варіант 10

#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 = -26;

int High = 23;

int rowCount = 6;

int colCount = 8;

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

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

q[i] = new int[colCount];

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

Print(q, rowCount, colCount);

int S = 0;

int k = 0;

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

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

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

Print(q, rowCount, colCount);

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

delete[] q[i];

delete[] q;

return 0;

}

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

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

}

void Print(int\*\* q, 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) << q[i][j];

cout << endl;

}

cout << endl;

}

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

{

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

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

if (!(q[i][j] % 2 == 0 && q[i][j] > 0))

{

S += q[i][j];

k++;

q[i][j] = 0;

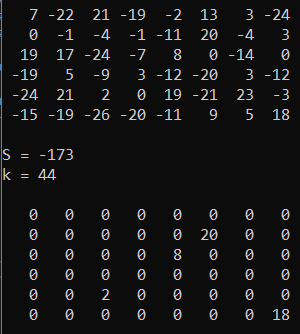
}

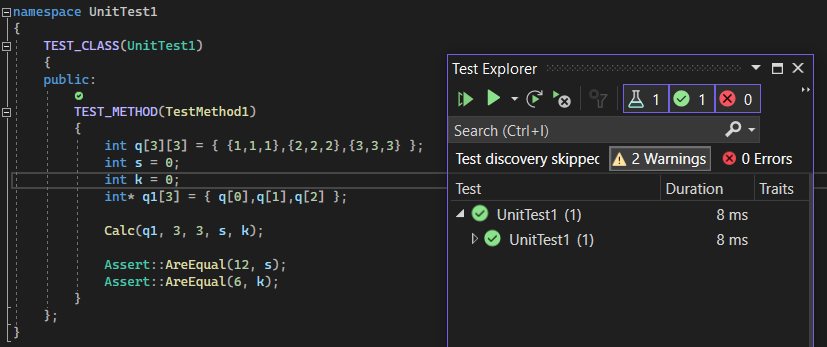
}

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

*https://github.com/StassNG/7.1*

**Результати програми та unit-тесту:**

****

****

**Висновок:** я навчився шукати задані елементи та впорядковувати матриці.