ЗВІТ

про виконання лабораторної роботи № 7.02

«Опрацювання багатовимірних масивів ітераційним і рекурсивним способами»

з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

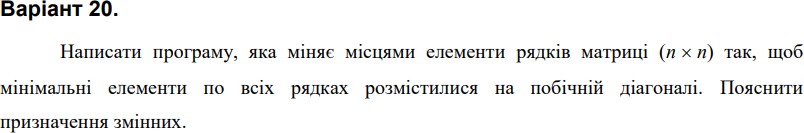
# Мета роботи:

Навчитися опрацьовувати двовимірні та багатовимірні масиви ітераційним і рекурсивним способами

# Умова завдання:

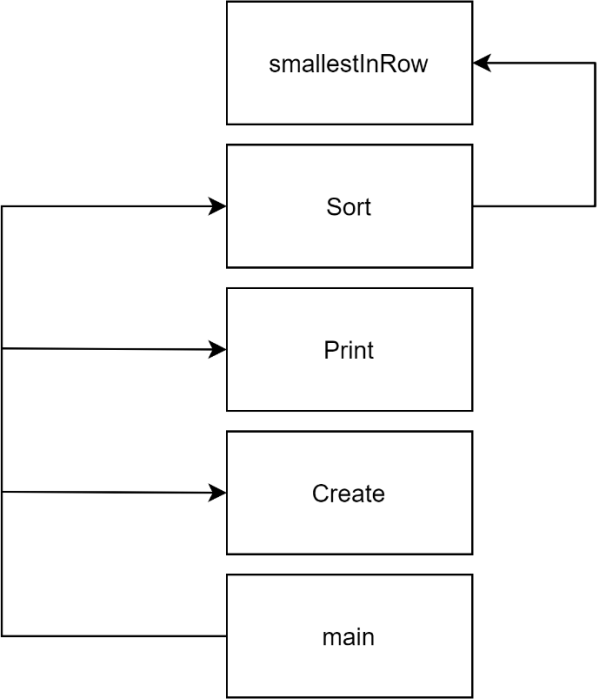
Необхідно: - сформувати матрицю із елементів цілого типу (величину кожної розмірності та діапазон значень елементів – вибрати самостійно); - вивести її на екран у вигляді таблиці, використовуючи форматне виведення; - виконати вказані у завданні дії; - вивести результат, причому, якщо матриця була змінена – то вивести на екран модифіковану матрицю у вигляді окремої таблиці, використовуючи виведення з тими самими специфікаціями формату.

1. Ітераційний спосіб. Усі вказані дії необхідно реалізувати за допомогою окремих функцій. Інформація у функції має передаватися лише за допомогою параметрів. Використання глобальних змінних – не допускається. 2. Рекурсивний спосіб. Усі вказані дії необхідно реалізувати за допомогою окремих рекурсивних функцій. Інформація у функції має передаватися лише за допомогою параметрів. Використання глобальних змінних – не допускається. Кожна функція має виконувати лише одну роль, і ця роль має бути відображена у назві функції. Рекурсивний та ітераційний способи – це 2 різні проекти, для яких потрібно 2 різних unit-тести і 2 різних звіти. «Функція, яка повертає / обчислює / шукає ...» – має не виводити ці значення, а повернути їх у місце виклику як результат функції або як відповідний вихідний параметр. Завдання: Написати програму, яка шукає найменший парний елемент побічної діагоналі матриці (n  n). Пояснити призначення змінних.



# Структурна схема програми:

**Ітераційний спосіб:**



# Текст програми:

#include <iostream> #include <iomanip> #include <time.h> using namespace std;

void Create(int\*\* a, const int n, const int Low, const int High); void Print(int\*\* a, const int n);

int smallestInRow(int\* a, const int n); void Sort(int\*\* a, const int n);

int main()

{

srand((unsigned)time(NULL)); int Low = 7;

int High = 65; int n;

cout << "n = "; cin >> n; int\*\* a = new int\* [n]; for (int i = 0; i < n; i++)

a[i] = new int[n]; Create(a, n, Low, High); Print(a, n);

cout << " Modified matrix :\n"; Sort(a, n);

Print(a, n);

for (int i = 0; i < n; i++) delete[] a[i];

delete[] a; return 0;

}

void Create(int\*\* a, const int n, const int Low, const int High)

{

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

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

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

}

void Print(int\*\* a, const int n)

{

cout << endl;

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

{

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

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

}

cout << endl;

}

int smallestInRow(int\* a, const int n)

{

int min = a[0], index = 0; for (int i = 0; i < n; i++)

{

if (a[i] < min)

{

index = i; min = a[i];

}

}

return index;

}

void Sort(int\*\* a, const int n)

{

int tmp, index = 0;

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

{

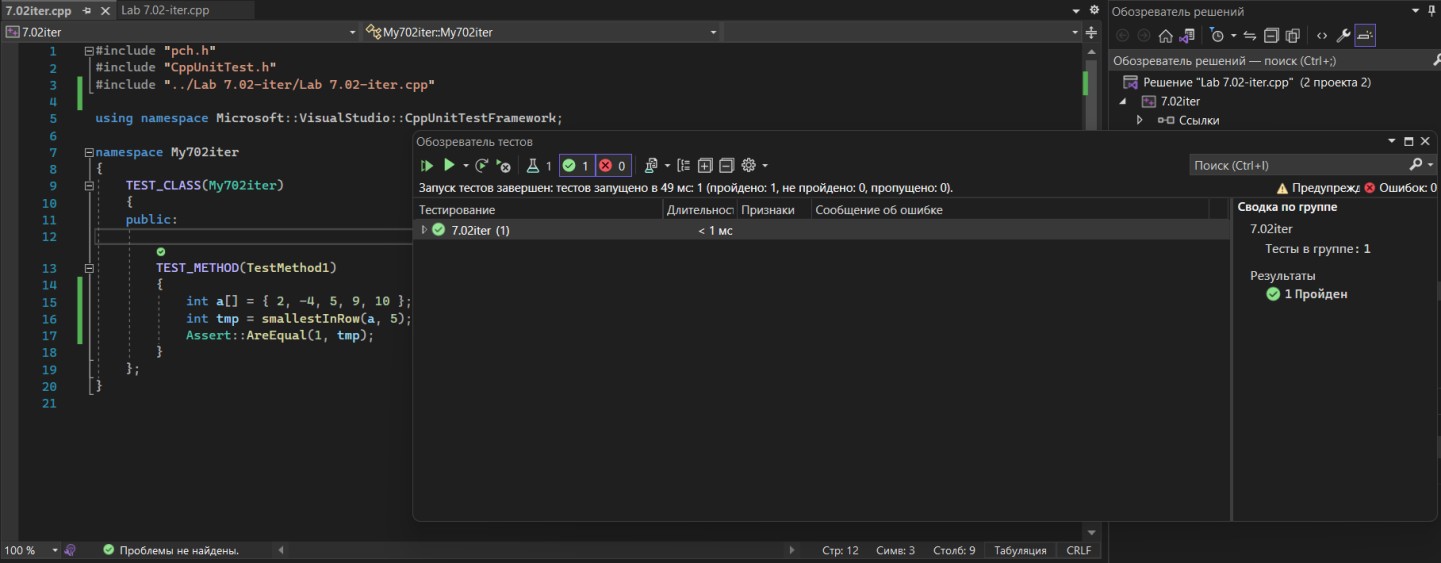
index = smallestInRow(a[i], n); tmp = a[i][index];

a[i][index] = a[i][n - 1 - i]; a[i][n - 1 - i] = tmp;

}

}

# Результати unit-тесту:



**Текст unit-тесту:**

#include "pch.h" #include "CppUnitTest.h"

#include "../Lab 7.02-iter/Lab 7.02-iter.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace My702iter

{

TEST\_CLASS(My702iter)

{

public:

TEST\_METHOD(TestMethod1)

{

int a[] = { 2, -4, 5, 9, 10 };

int tmp = smallestInRow(a, 5); Assert::AreEqual(1, tmp);

}

};

}

**Посилання**:<https://github.com/milianets20/Lab-7.2-7.3>

**Висновок**: виконуючи лабораторну роботу я навчився опрацьовувати двовимірні та багатовимірні масиви ітераційним і рекурсивним способами.