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

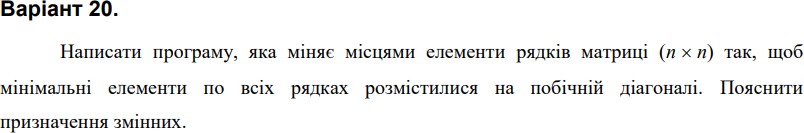
# Мета роботи:

Навчитися опрацьовувати двовимірні та багатовимірні масиви ітераційним і рекурсивним способами

# Умова завдання:

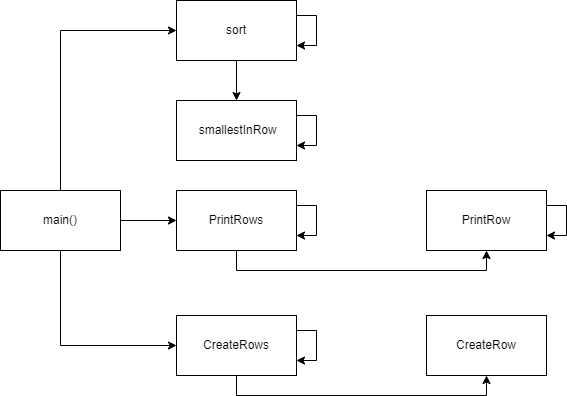
Необхідно: - сформувати матрицю із елементів цілого типу (величину кожної розмірності та діапазон значень елементів – вибрати самостійно); - вивести її на екран у вигляді таблиці, використовуючи форматне виведення; - виконати вказані у завданні дії; - вивести результат, причому, якщо матриця була змінена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

**Рекурсивний спосіб:**



# Текст програми:

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

void CreateRow(int\*\* a, const int rowNo, const int N, const int Low, const int High, int colNo)

{

a[rowNo][colNo] = Low + rand() % (High - Low + 1); if (colNo < N - 1)

CreateRow(a, rowNo, N, Low, High, colNo + 1);

}

void CreateRows(int\*\* a, const int N, const int Low, const int High, int rowNo)

{

CreateRow(a, rowNo, N, Low, High, 0); if (rowNo < N - 1)

CreateRows(a, N, Low, High, rowNo + 1);

}

void PrintRow(int\*\* a, const int rowNo, const int N, int colNo)

{

cout << setw(4) << a[rowNo][colNo]; if (colNo < N - 1)

PrintRow(a, rowNo, N, colNo + 1);

else

}

cout << endl;

void PrintRows(int\*\* a, const int N, int rowNo)

{

PrintRow(a, rowNo, N, 0); if (rowNo < N - 1)

PrintRows(a, N, rowNo + 1);

else

}

cout << endl;

int smallestInRow(int\* a, const int N, int index, int min, int i)

{

if (i >= N)

return index;

else

if (a[i] < min) {

index = i; min = a[i];

}

return smallestInRow(a, N, index, min, i + 1);

}

void Sort(int\*\* a, const int N, int i)

{

int tmp;

int index = smallestInRow(a[i], N,0, a[i][0], 0); tmp = a[i][index];

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

if (i < N - 1)

Sort(a, N, i + 1);

}

int main()

{

srand((unsigned)time(NULL)); int N;

cout << "N = "; cin >> N; cout << endl;

int\*\* a = new int\* [N]; for (int i = 0; i < N; i++)

a[i] = new int[N]; int Low = 7, High = 65;

CreateRows(a, N, Low, High, 0); PrintRows(a, N, 0);

cout << " Modified matrix :\n\n"; Sort(a, N, 0);

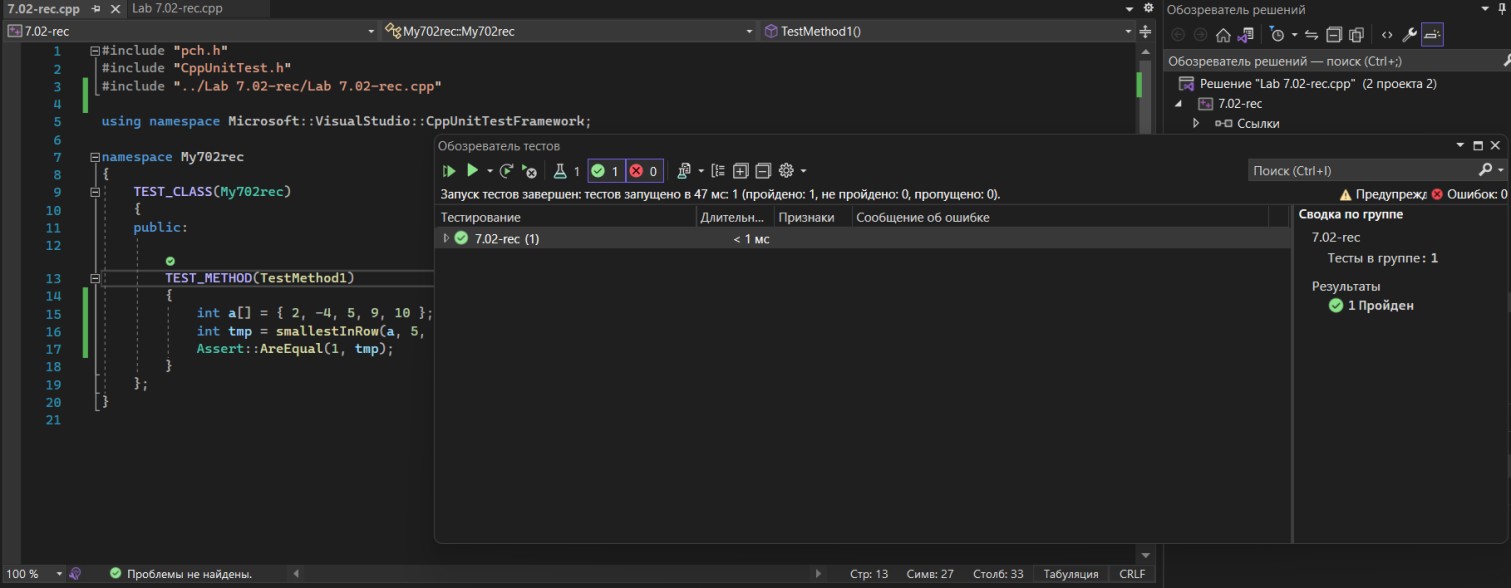
PrintRows(a, N,0);

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

delete[] a; return 0;

}

# Результати unit-тесту:



**Текст unit-тесту:**

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

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

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace My702rec

{

TEST\_CLASS(My702rec)

{

public:

TEST\_METHOD(TestMethod1)

{

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

int tmp = smallestInRow(a, 5, 0, 2, 0); 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.