ЗВІТ

про виконання лабораторної роботи № 6.1 рекурсивним способом

«Пошук елементів одновимірного масиву ітераційним та рекурсивним способом»

з дисципліни

«Алгоритмізація та програмування»

студентки групи ІК-11

Турковської Марти Михайлівни

**Умова завдання:**

Написати програму, яка за допомогою генератора випадкових чисел формує вказаний масив – такий, що значення його елементів належать заданому діапазону. Обчислити кількість та суму тих елементів, які задовольняють вказаному критерію; а також замінити нулями ці елементи.

Необхідно:

- сформувати масив;

- вивести його на екран у вигляді рядка, використовуючи форматне виведення;

- виконати вказані у завдання дії (обчислити кількість та суму тих елементів, які задовольняють вказаному критерію; а також замінити нулями ці елементи);

- вивести результат – обчислену кількість та суму, та вивести на екран модифікований масив у вигляді наступного рядка, використовуючи виведення з тими самими специфікаціями формату.

Всі вказані дії необхідно реалізувати за допомогою окремих функцій.

*Спосіб 1.* Всі функції, у т.ч. ті, що реалізують обчислення кількості, суми, та заміну нулями вказаних елементів, – використовують ітераційні алгоритми.

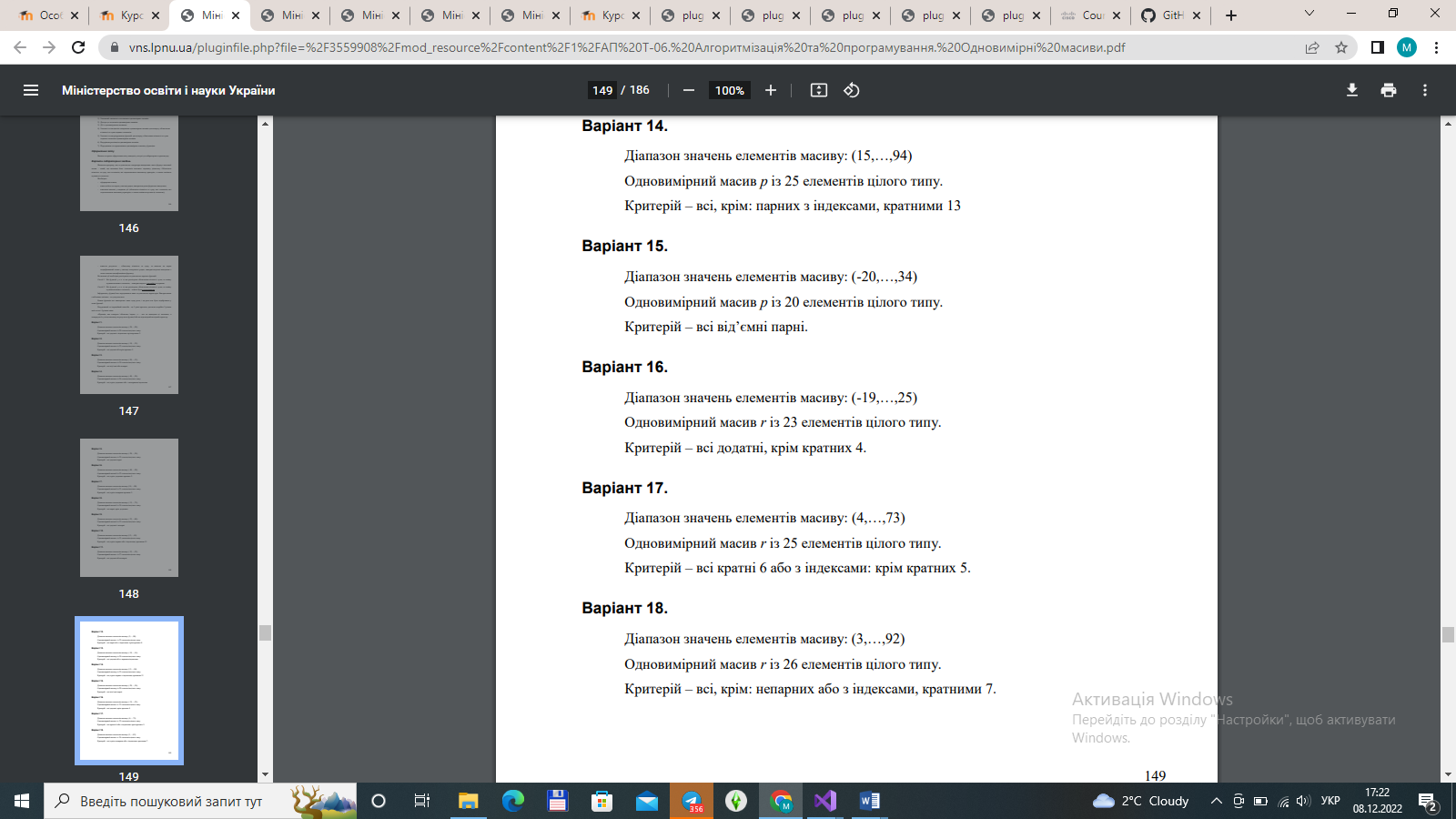
*Спосіб 2.* Всі функції, у т.ч. ті, що реалізують обчислення кількості, суми, та заміну нулями вказаних елементів, – мають бути рекурсивними.

Інформація у функції має передаватися лише за допомогою параметрів. Використання глобальних змінних – не допускається.

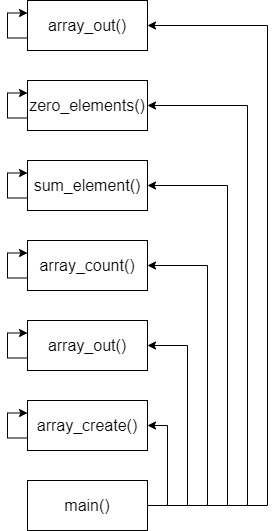
Кожна функція має виконувати лише одну роль, і ця роль має бути відображена у назві функції.

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.



**Структурна схема програми:**



**Текст програми:**

#include <iostream>

#include <iomanip>

using namespace std;

void array\_create(int\* arr, const int min, const int max, const int size, int i) {

arr[i] = min + rand() % (max - min + 1);

if (i < size - 1)

array\_create(arr, min, max, size, i + 1);

}

void array\_out(int\* arr, const int size, int i)

{

cout << setw(4) << arr[i];

if (i < size - 1)

array\_out(arr, size, i + 1);

}

int array\_count(int\* arr, int const size, int count, int i)

{

if (i <= size - 1) {

if (arr[i] % 6 == 0 || i % 5 != 0)

count++;

array\_count(arr, size, count, i + 1);

}

else

return count;

}

int sum\_element(int\* arr, int const size, int sum, int i)

{

if (i <= size - 1) {

if (arr[i] % 6 == 0 || i % 5 != 0)

sum += arr[i];

sum\_element(arr, size, sum, i + 1);

}

else

return sum;

}

void zero\_elements(int\* arr, int const size, int i)

{

if (i < size - 1) {

if (arr[i] % 6 == 0 || i % 5 != 0)

arr[i] = 0;

zero\_elements(arr, size, i + 1);

}

}

int main()

{

srand((unsigned)time(NULL));

const int n = 25;

int r[n];

int count = 0;

const int max = 73;

const int min = 4;

array\_create(r, min, max, n, 0);

array\_out(r, n, 0);

cout << endl;

cout << endl;

array\_count(r, n, count, 0);

int num = array\_count(r, n, count, 0);

cout << "The number of elements that are multiples of 6 or with indices : except for multiples of 5 = " << num << endl;;

cout << endl;

sum\_element(r, n, 0, 0);

int suma = sum\_element(r, n, 0, 0);

cout << "The sum of elements that are multiples of 6 or with indices : except for multiples of 5 = " << suma << endl;

cout << endl;

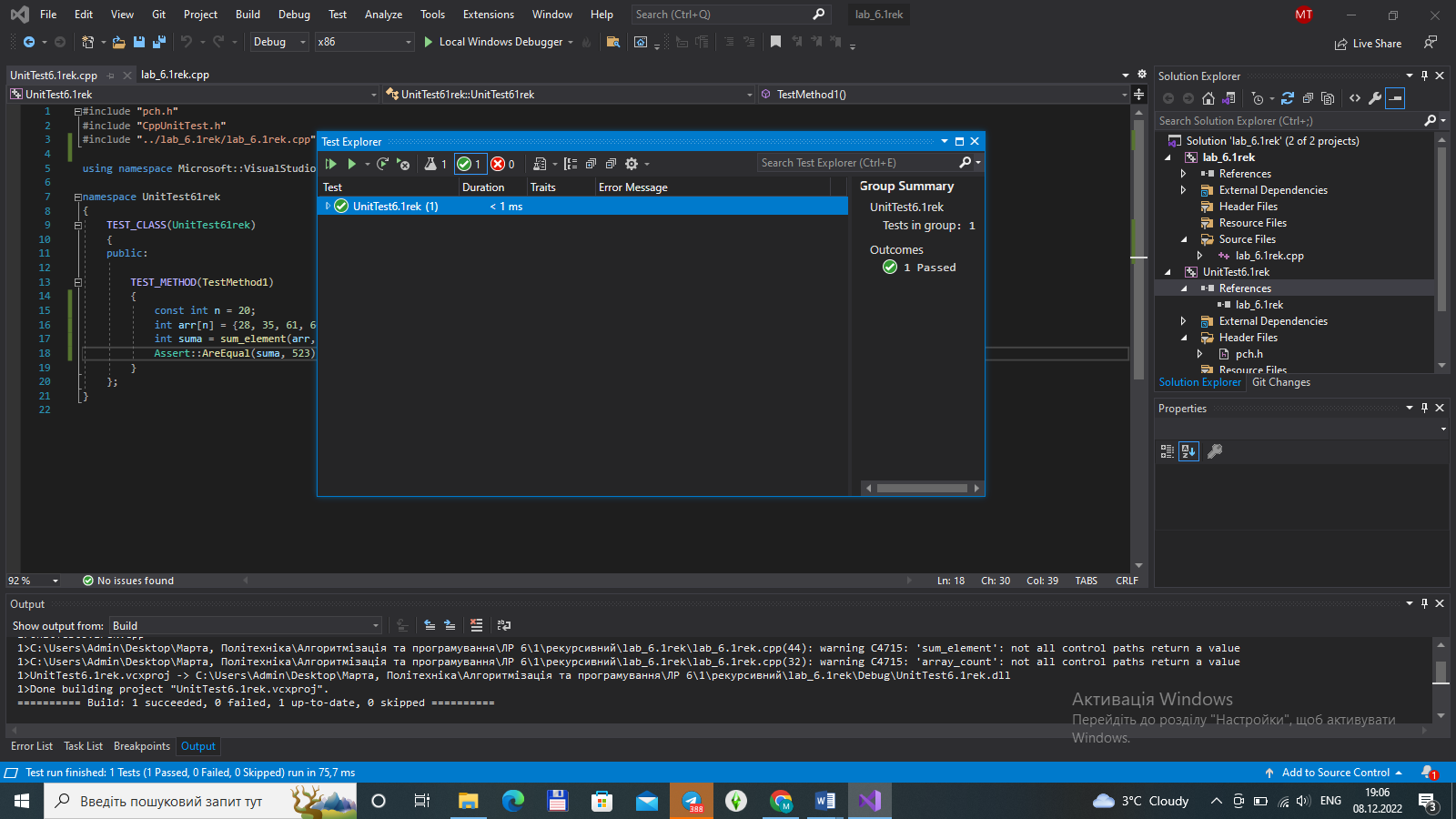
zero\_elements(r, n, 0);

array\_out(r, n, 0);

}

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

**Результати unit-тесту:**



**Висновок:** навчилися програмувати пошук послідовним переглядом, обчислення кількості та суми заданих елементів одновимірного масиву.