ЗВІТ

про виконання лабораторної роботи № 6.02

«Опрацювання одновимірних масивів ітераційними та рекурсивними способами»

з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

# Мета роботи:

Навчитися опрацьовувати одновимірні масиви

# Умова завдання:

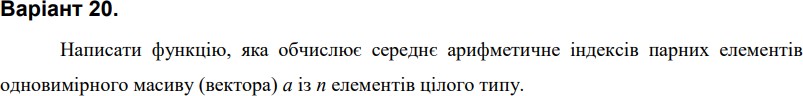
Необхідно написати програму для того, щоб виконати такі дії: - сформувати масив; - вивести його на екран у вигляді рядка, використовуючи форматне виведення; - виконати вказані у завдання дії; - вивести результат, причому, якщо масив був змінений – то вивести на екран модифікований масив у вигляді наступного рядка, використовуючи виведення з тими самими

специфікаціями формату. Спосіб 1. Всі вказані дії необхідно реалізувати за допомогою окремих функцій, що використовують ітераційні алгоритми.

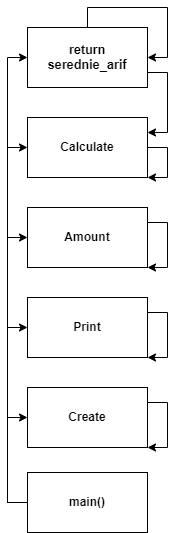
Спосіб 2. Всі вказані дії необхідно реалізувати за допомогою окремих рекурсивних функцій. Інформація у функції має передаватися лише за допомогою параметрів. Використання глобальних змінних – не допускається.

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.



# Структурна схема програми:



**Рекурсивний спосіб:**

# Текст програми:

#include <iostream> #include <ctime> #include <Windows.h>

using namespace std;

void create(int\* a, const int n, const int i, const int HIGH, const int LOW); void print(int\* a, const int n, const int i);

double calculate(int\* a, const int n, const int i, double& serednie\_arif, int& amount, int& all\_i\_elements);

int main()

{

srand(time(NULL)); int n;

cout << "A: "; cin >> n; int\* a = new int[n];

const int i = 0; const int HIGH = 53; const int LOW = -17;

create(a, n, i, HIGH, LOW);

cout << "a[] = { ";

print(a, n, i); cout << "}" << endl;

double serednie\_arif = 0.; int amount = 0;

int all\_i\_elements = 0; cout << "F = " <<

calculate(a, n, i, serednie\_arif, amount, all\_i\_elements) << endl; delete[]a;

system("pause"); return 0;

}

void create(int\* a, const int n, const int i, const int HIGH, const int LOW)

{

a[i] = LOW + rand() % (HIGH - LOW + 1);

if (i < n - 1)

create(a, n, i + 1, HIGH, LOW);

else

}

return;

void print(int\* a, const int n, const int i)

{

cout << a[i] << " "; if (i < n - 1)

print(a, n, i + 1);

else

}

return;

double calculate(int\* a, const int n, const int i, double& serednie\_arif, int& amount, int& all\_i\_elements)

{

if (i < n)

{

bool umova = false; if (a[i] % 2 == 0)

{

umova = true; amount += i; all\_i\_elements++;

}

if (umova)

{

}

else

}

else

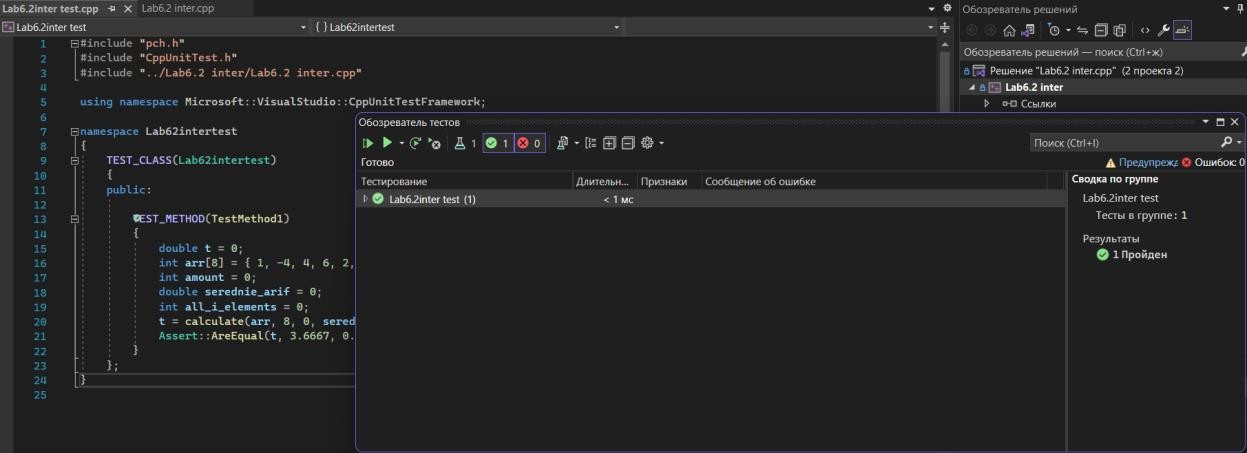
serednie\_arif = (1. \* amount) / all\_i\_elements;

calculate(a, n, i + 1, serednie\_arif, amount, all\_i\_elements); calculate(a, n, i + 1, serednie\_arif, amount, all\_i\_elements);

return serednie\_arif;

}

# Результати unit-тесту:



**Текст unit-тесту:**

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

#include "../Lab6.2 rec/Lab6.2 rec.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace LAB62rectest

{

TEST\_CLASS(LAB62rectest)

{

public:

TEST\_METHOD(TestMethod1)

{

double t = 0;

int arr[8] = { 0, -2, 3, 5, 1, -1, -6, 8 };

int amount = 0;

double serednie\_arif = 0; int all\_i\_elements = 0;

t = calculate(arr, 8, 0, serednie\_arif, all\_i\_elements, amount); Assert::AreEqual(t, 3.5);

}

};

}

**Посилання**: <https://github.com/milianets20/Lab-6.2-rec>

**Висновок**: виконуючи лабораторну роботу я навчився програмувати пошук послідовним переглядом, обчислення кількості та суми заданих елементів одновимірного масиву.