ЗВІТ

про виконання лабораторної роботи № 6.03

«Опрацювання одновимірних масивів за допомогою звичайних функцій та шаблонів»

з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

# Мета роботи:

Навчитися опрацьовувати одновимірні масиви за допомогою звичайних функцій та шаблонів. Навчитися використовувати шаблони функцій.

# 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

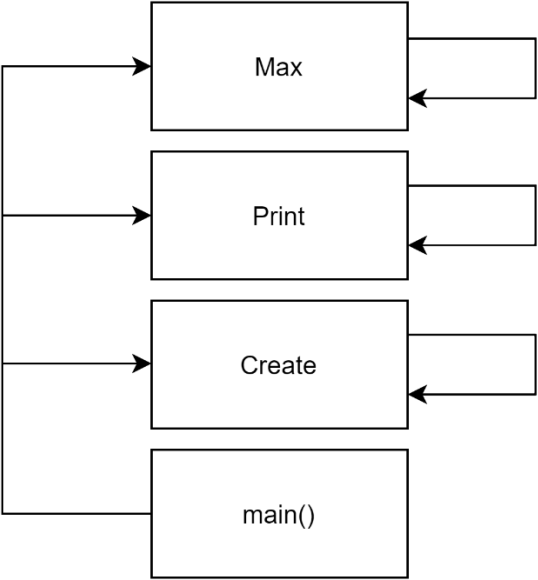
специфікаціями формату. Всі вказані дії необхідно реалізувати за допомогою:  окремих функцій – перше завдання, або  окремих рекурсивних функцій – друге завдання. Інформацію у функції слід передавати лише за допомогою параметрів. Використання глобальних змінних – не допускається.

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. У кожному випадку (ітераційні чи рекурсивні функції) масиви слід опрацьовувати двома способами – 1) за допомогою звичайних функцій; 2) використовувати шаблони функцій для реалізації універсального алгоритму.



# Структурна схема програми:



**Рекурсивний спосіб:**

# Текст програми:

#include <iostream> #include <iomanip> #include <time.h>

using namespace std;

void Create(int\* a, const int size, const int A, const int B, const int i)

{

a[i] = A + rand() % (B - A + 1);

if (i < size - 1)

Create(a, size, A, B, i + 1);

}

template <typename T>

void Create(T\* a, const int size, const T A, const T B, const int i)

{

a[i] = A + rand() % (B - A + 1);

if (i < size - 1)

Create(a, size, A, B, i + 1);

}

void Print(int\* a, const int size, const int i)

{

cout << setw(4) << a[i]; if (i < size - 1)

Print(a, size, i + 1); else

cout << endl;

}

template <typename T>

void Print(T\* a, const int size, const int i)

{

cout << setw(4) << a[i]; if (i < size - 1)

Print(a, size, i + 1); else

cout << endl;

}

int Max(int\* a, const int size, int max, const int i)

{

if(a[i] > max)

max = a[i];

if (i < size - 1)

return Max(a, size, max, i+1); else

return max;

}

template <typename T>

T Max(T\* a, const int size, T max, const int i)

{

if (a[i] > max)

max = a[i]; if (i < size - 1)

return Max(a, size, max, i + 1); else

return max;

}

int main()

{

srand((unsigned)time(NULL)); const int n = 15;

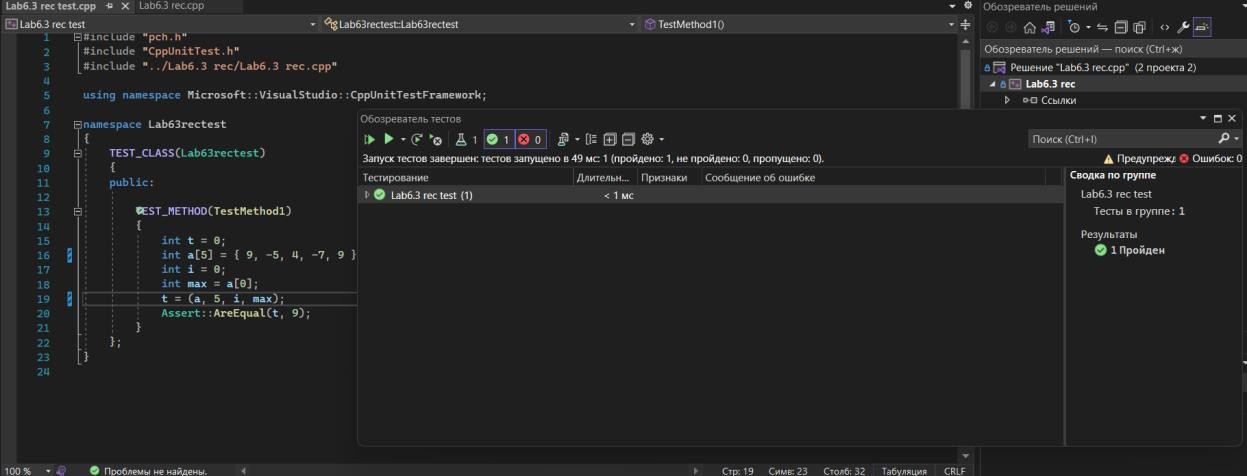
int a[n]; int A = -17; int B = 53;

Create<int>(a, n, A, B, 0); Print<int>(a, n, 0);

cout << "The largest number = " << Max<int>(a, n, a[0], 1) << endl; return 0;

}

# Результати unit-тесту:



**Текст unit-тесту:**

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

#include "../Lab6.3 rec/Lab6.3 rec.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace Lab63rectest

{

TEST\_CLASS(Lab63rectest)

{

public:

TEST\_METHOD(TestMethod1)

{

int t = 0;

int a[5] = { 9, -5, 4, -7, 9 };

int i = 0;

int max = a[0];

t = (a, 5, i, max); Assert::AreEqual(t, 9);

}

};

}

**Посилання**: [https://github.com/milianets20/Lab-6.3-rec](https://github.com/milianets20/Lab-6.3-iter)

**Висновок**: виконуючи лабораторну роботу я навчився опрацьовувати одновимірні масиви за допомогою звичайних функцій та шаблонів та навчився використовувати шаблони функцій.