ЗВІТ  
про виконання лабораторної роботи № 6.3.1 (ітерація)  
«Опрацювання одновимірних  
масивів за допомогою звичайних функцій та шаблонів»  
з дисципліни  
«Алгоритмізація та програмування»  
студента групи ІН-105 Б  
Горанов Анастас

**Мета роботи:** навчитися опрацьовувати одновимірні масиви за допомогою звичайних функцій та шаблонів. Навчитися використовувати шаблони функцій.

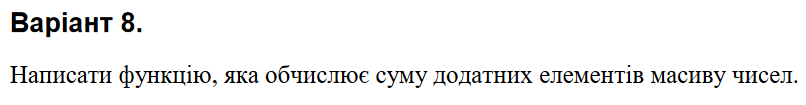
**Умова завдання:** необхідно написати програму для того, щоб виконати такі дії:

- 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.

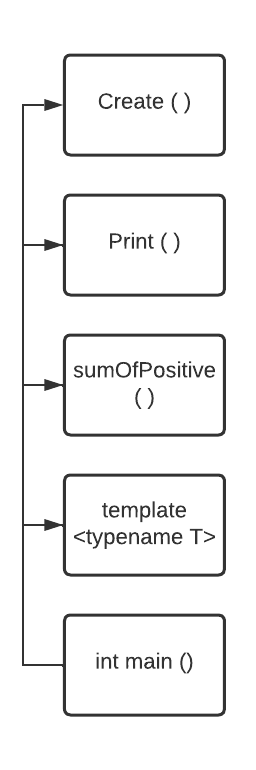
Всі вказані дії необхідно реалізувати за допомогою:

• 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:**



**Текст програми:**

// PR6.3.1(iter).cpp

// Горанов Анастас

// Лабораторна робота №6.3.1(iter)

// Опрацювання одновимірних масивів за допомогою звичайних функцій та шаблонів

// Варіант 8

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

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

{

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

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

}

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

{

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

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

cout << endl;

}

int sumOfPositive(int\* a, const int size)

{

int i, z = 0;

for (i = 0; i < size; i++) {

if (a[i] > 0) {

z += a[i];

}

}

return z;

}

template <typename T> // шаблон функції

T sumOfPositiveTemplete(T\* a, const int size)

{

T z = 0;

for (int i = 0; i < size; i++) {

if (a[i] > 0) {

z += a[i];

}

}

return z;

}

int main()

{

srand((unsigned)time(NULL));

const int n = 10;

int a[n];

int Low = -10;

int High = 10;

Create(a, n, Low, High);

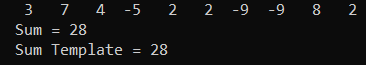
Print(a, n);

cout << " " << "Sum = " << sumOfPositive(a, n) << endl;

cout << " " << "Sum Template = " << sumOfPositiveTemplete(a, n) << endl;

return 0;

}



**Посилання на Git-репозиторій:** *https://github.com/StassNG/PR6.3.1*

**Результати unit-тесту:**

****

**Текст Unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../PR6.3.1(iter).cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(sumOfPositiveTempleteTest)

{

int arr[] = { 1, -2, -4,12, 1, -2, -3, 4 };

Assert::IsTrue(sumOfPositiveTemplete(arr, 8) == 18);

float arr1[] = { -1.4, 2.4, 1.6, -5.2, 3.5 };

Assert::IsTrue(fabs(sumOfPositiveTemplete(arr1, 5) - 7.5) < 0.0000000001);

unsigned arr2[] = { 1, 2, 4,12, 1, 2, 3, 4 };

Assert::IsTrue(sumOfPositiveTemplete(arr2, 8) == 29);

}

};

}

**Висновок:** навчився опрацьовувати одновимірні масиви за допомогою звичайних функцій та шаблонів. Навчився використовувати шаблони функцій.