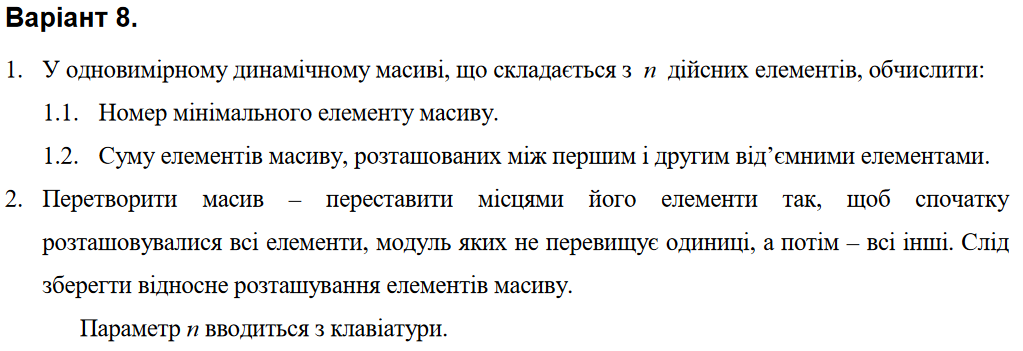
ЗВІТ  
про виконання лабораторної роботи № 6.4.1 (ітерація)  
«Опрацювання та  
впорядкування одновимірних динамічних масивів»  
з дисципліни  
«Алгоритмізація та програмування»  
студента групи ІН-105 Б  
Горанов Анастас

**Мета роботи:** навчитися опрацьовувати одновимірні динамічні масиви.

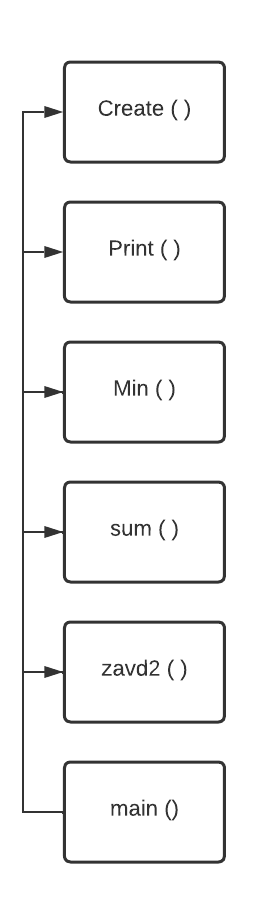
**Умова завдання:** Необхідно написати програму для того, щоб виконати такі дії:

- сформувати масив;  
- вивести його на екран у вигляді рядка, використовуючи форматне виведення;  
- виконати вказані у завдання дії;  
- вивести результати, причому модифікований масив вивести на екран 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.  
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.



**Структурна схема програми:**



**Текст програми:**

// PR6.4.1(iter).cpp

// Горанов Анастас

// Лабораторна робота №6.4.1(iter)

// Опрацювання та впорядкування одновимірних динамічних масивів

// 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 Min(int\* a, const int size)

{

int z = 0;

int min = a[0];

for (int i = 1; i < size; i++) {

if (a[i] < min) {

min = a[i];

z = i;

}

}

return z + 1;

}

int sum(int\* a, const int size) {

int rez = 0;

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

if (a[i] < 0) {

for (int x = i + 1; x < size; x++) {

if (a[x] < 0) {

return rez;

}

else {

rez = rez + a[x];

}

}

}

}

return rez;

}

void zavd2(int\* a, const int size) {

int i, j;

for (i = 0; i < size - 1; i++)

for (j = 0; j < size - i - 1; j++)

if (fabs(a[j]) >= 1 && fabs(a[j + 1]) <= 1)

swap(a[j], a[j + 1]);

}

int main()

{

srand((unsigned)time(NULL));

int n;

cout << "Input: "; cin >> n;

int\* arr = new int[n];

int Low = -2;

int High = 10;

Create(arr, n, Low, High);

Print(arr, n);

cout << "Min elem: " << Min(arr, n) << endl;

cout << "Sum: " << sum(arr, n) << endl;

zavd2(arr, n);

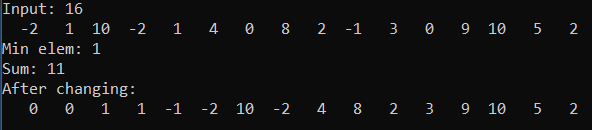
cout << "After changing: " << endl;

Print(arr, n);

delete[] arr;

return 0;

}



**Посилання на Git-репозиторій:** *https://github.com/StassNG/PR6.4.1*

**Результати unit-тесту:**



**Текст Unit-тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../PR6.4.1(iter).cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(TestMethod1)

{

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

Assert::IsTrue(sum(arr, 8) == 13);

}

};

}

**Висновок:** навчився опрацьовувати одновимірні динамічні масиви.