ЗВІТ

про виконання лабораторної роботи № 6.04

«Опрацювання та впорядкування одновимірних динамічних масивів» з дисципліни

«Алгоритмізація та програмування» студента групи РІ-12

Мілянця Мар`яна

# Мета роботи:

Навчитися опрацьовувати одновимірні динамічні масиви.

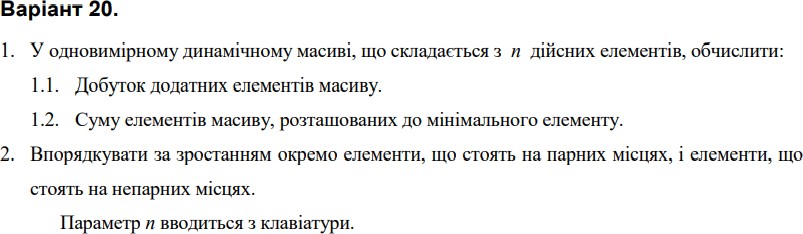
# Умова завдання:

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

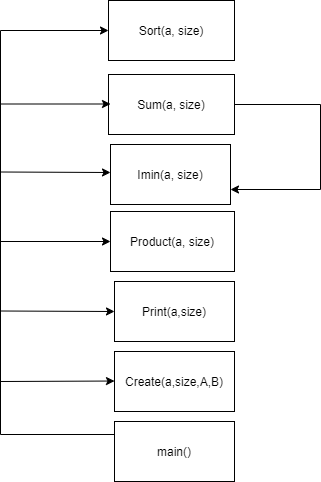
1. Всі вказані дії необхідно реалізувати за допомогою окремих функцій, що використовують ітераційні алгоритми. Спосіб 2. Всі вказані дії необхідно реалізувати за допомогою окремих рекурсивних функцій. Інформацію у функції слід передавати лише за допомогою параметрів. Використання глобальних змінних – 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:



**Ітераційний спосіб:**

# Текст програми:

#include <iostream> #include <iomanip>

using namespace std;

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

{

a = new int[size];

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

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

}

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

{

for (int i = 0; i < size\_a; i++) cout << setw(4) << a[i];

cout << endl;

}

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

{

int P = 1;

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

{

if (a[i] > 0)

P \*= a[i];

}

return P;

}

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

{

int min = a[0]; int imin{};

for (int i = 1; i < size; i++) if (a[i] < min)

{

}

return imin;

}

min = a[i]; imin = i;

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

{

int imin = Imin(a, size); int S = 0;

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

{

if (imin != 0)

S += a[i];

}

if (imin == 0)

{

cerr << " Sumarry can`t be found" << endl; return 0;

}

return S;

}

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

{

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

{

int k = 0;

for (int j = 0; j < size - 2; j++)

if (a[j] > a[j + 2])

{

}

}

if (k == 0)

return;

int tmp = a[j]; a[j] = a[j + 2]; a[j + 2] = tmp; k = 1;

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

{

int k = 0;

for (int j = 1; j < size - 2; j++)

if (a[j] > a[j + 2])

{

}

if (k == 0)

int tmp = a[j]; a[j] = a[j + 2]; a[j + 2] = tmp; k = 1;

return;

}

}

int main()

{

srand((unsigned)time(NULL)); int n;

cout << " n = "; cin >> n; int\* a;

int Low = -17; int High = 30;

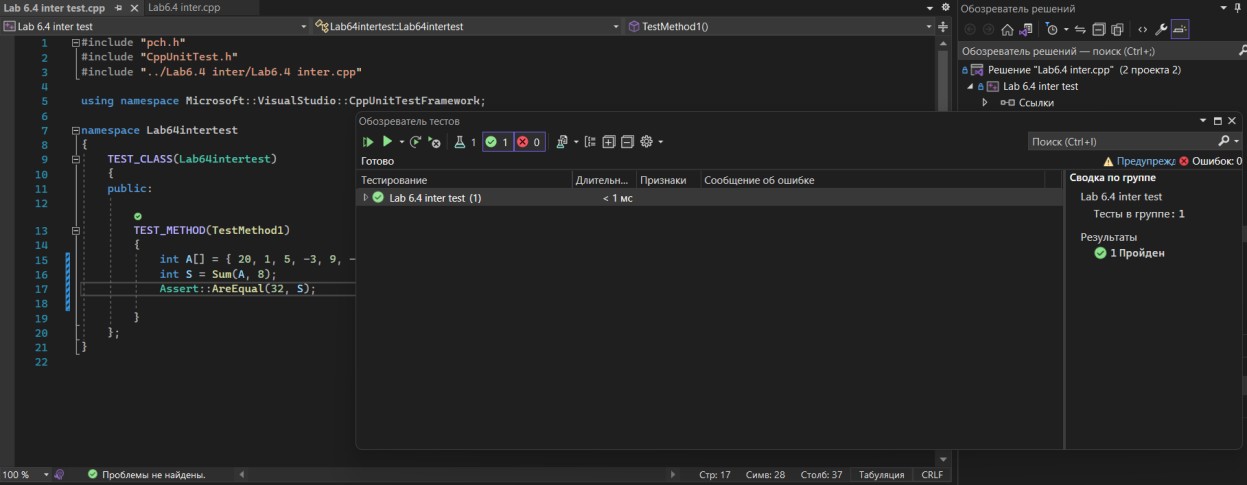
Create(a, n, Low, High); Print(a, n);

cout << " Product of positive elements = " << Product(a, n) << endl; cout << " Summary of elements before minimal = " << Sum(a, n) << endl; Sort(a, n);

Print(a, n); return 0;

}

# Результати unit-тесту:



**Текст unit-тесту:**

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

#include "../Lab6.4 inter/Lab6.4 inter.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework; namespace Lab64intertest

{

TEST\_CLASS(Lab64intertest)

{

public:

TEST\_METHOD(TestMethod1)

{

int A[] = { 20, 1, 5, -3, 9, -6, 15, 24 };

int S = Sum(A, 8); Assert::AreEqual(32, S);

}

};

}

**Посилання**: <https://github.com/milianets20/Lab-6.4-iter>

**Висновок**: виконуючи лабораторну роботу я навчився опрацьовувати одновимірні динамічні масиви