ЗВІТ

про виконання лабораторної роботи № < 6.4 >

*«Опрацювання та впорядкування*

*одновимірних динамічних масивів»*

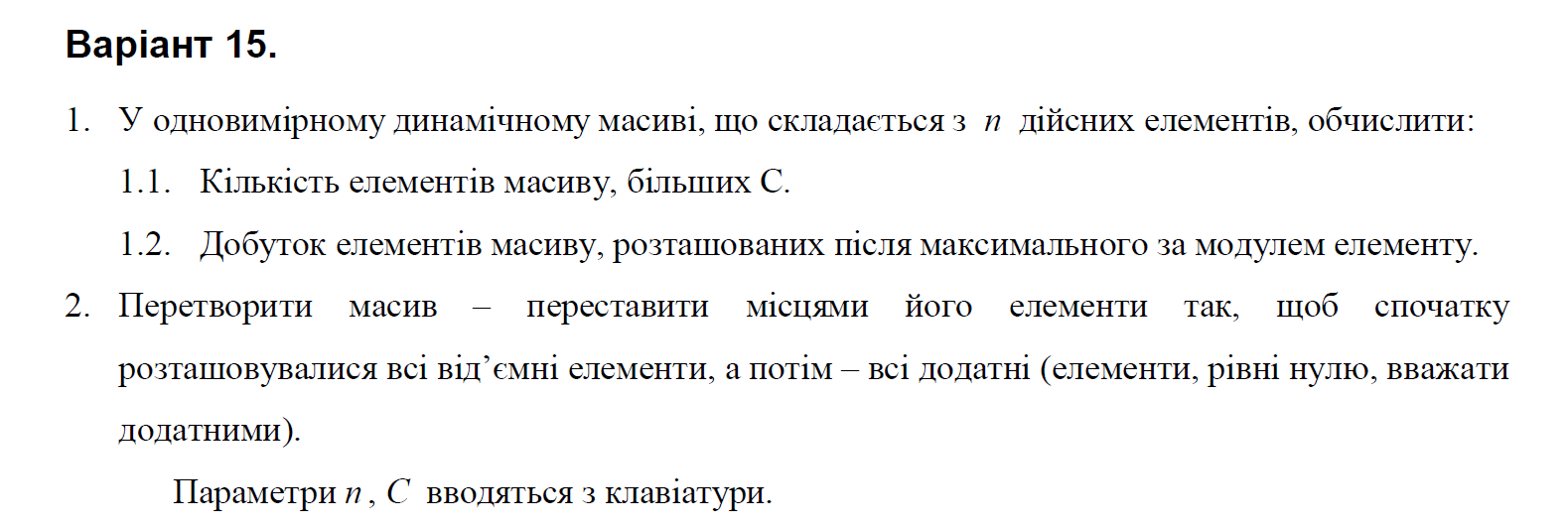
з дисципліни

«Алгоритмізація та програмування»

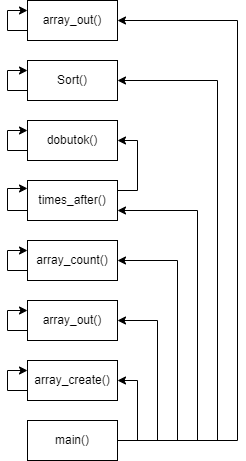
студента групи ІК-11

< *Снігура Стефана Андрійовича* >

**Умова завдання:**

****

**Структурна схема програми**

****

**Текст програми:**

#include <iostream>

#include <iomanip>

using namespace std;

void array\_create(int\* arr, const int min, const int max, const int size, int i) {

arr[i] = min + rand() % (max - min + 1);

if (i < size - 1)

array\_create(arr, min, max, size, i + 1);

}

void array\_out(int\* arr, const int size, int i)

{

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

if (i < size - 1)

array\_out(arr, size, i + 1);

}

int array\_count(int\* arr, int const size, int count, int i, int C)

{

if (i <= size - 1) {

if (arr[i] > C)

count++;

array\_count(arr, size, count, i + 1, C);

}

else

return count;

}

int dobutok(int\* arr, int const size, int k, int& dob) {

if (k + 1 <= size) {

dob \*= arr[k];

dobutok(arr, size, k + 1, dob);

}

else

return (arr, size, k , dob);

}

int times\_after(int\* arr, int const size, int i, int k, int max\_abs )

{

int dob = 1;

if (i < size) {

if (abs(arr[i]) >= max\_abs) {

max\_abs = abs(arr[i]);

k = ++i;

}

times\_after(arr, size, i + 1, k, max\_abs);

}

else

return dobutok(arr, size, k,dob);

}

void Find(int\* arr, int tmp, int& j)

{

if (tmp > arr[j])

{

j++;

Find(arr, tmp, j);

}

}

void Shift(int\* arr, int j, int k)

{

arr[k + 1] = arr[k];

if (k > j)

Shift(arr, j, k - 1);

}

void Sort(int\* arr, const int size, int i)

{

int tmp = arr[i];

int j = 0;

Find(arr, tmp, j);

Shift(arr, j, i - 1);

arr[j] = tmp;

if (i < size - 1)

Sort(arr, size, i + 1);

}

int main()

{

srand((unsigned)time(NULL));

int n, C;

int\* mass;

cout << "Write the number in array "; cin >> n;

cout << endl;

cout << "Write the value of C "; cin >> C;

cout << endl;

mass = new int[n];

int count = 0;

const int max = 40;

const int min = -40;

array\_create(mass, min, max, n, 0);

cout << "Created array "; array\_out(mass, n, 0);

cout << endl;

cout << endl;

array\_count(mass, n, count, 0, C);

int num = array\_count(mass, n, count, 0, C);

cout << "The number of numbers that are bigger than C = " << num << endl;;

cout << endl;

times\_after(mass, n, 0,0,0);

int product\_after = times\_after(mass, n,0,0,0);

cout << "The product of the numbers after the largest modulo number = " << product\_after;

cout << endl;

Sort(mass, n, 1);

cout << endl;

cout << "Sorted array "; array\_out(mass, n, 0);

cout << endl;

}

**UNIT-test:**

#include "pch.h"

#include "CppUnitTest.h"

#include "C:\Users\User\source\repos\lab6.4(recursion)\Source.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(TestMethod\_dobutok)

{

const int n = 25;

int arr[n] = { 1,2,3,4,5,6,7,8,40,10,11,12,13,4,15,16,17,18,19,20,-100,44,20,43,41 };

int dobutok = times\_after(arr, n, 0, 0,0);

Assert::AreEqual(dobutok, 1551440);

}

};

}

**Посилання на git-репозиторій з проектом:**

<https://github.com/BigTrouble-Git/ashtray.git>

**Висновки**:

Виконавши цю лабораторну роботу я навчився опрацьовувати одновимірні динамічні масиви .