Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

Звіт

з лабораторної роботи № 7 з дисципліни «Основи програмування-1»

«Одновимірні масиви»

Варіант 8

Виконав студент ІП-02 Глазунова Поліна Андріївна

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2020

**Лабораторна робота 7**

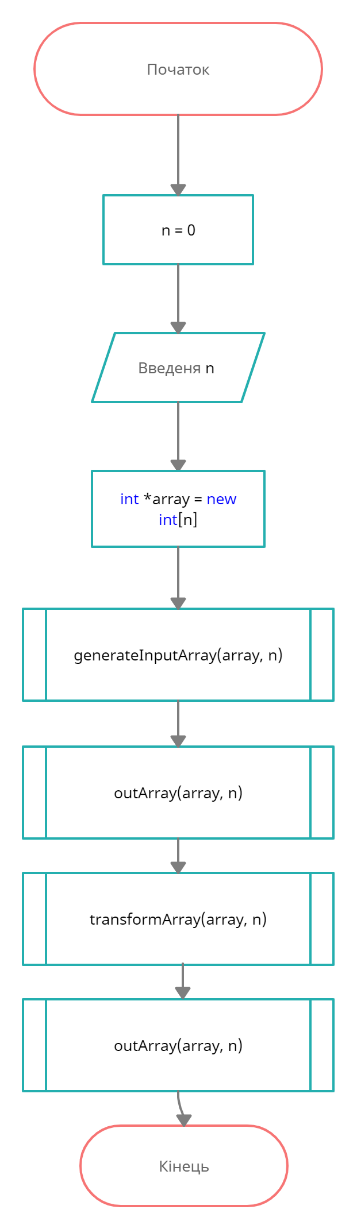
**Одновимірні масиви**

**Мета** – вивчити особливості обробки одновимірних масивів.

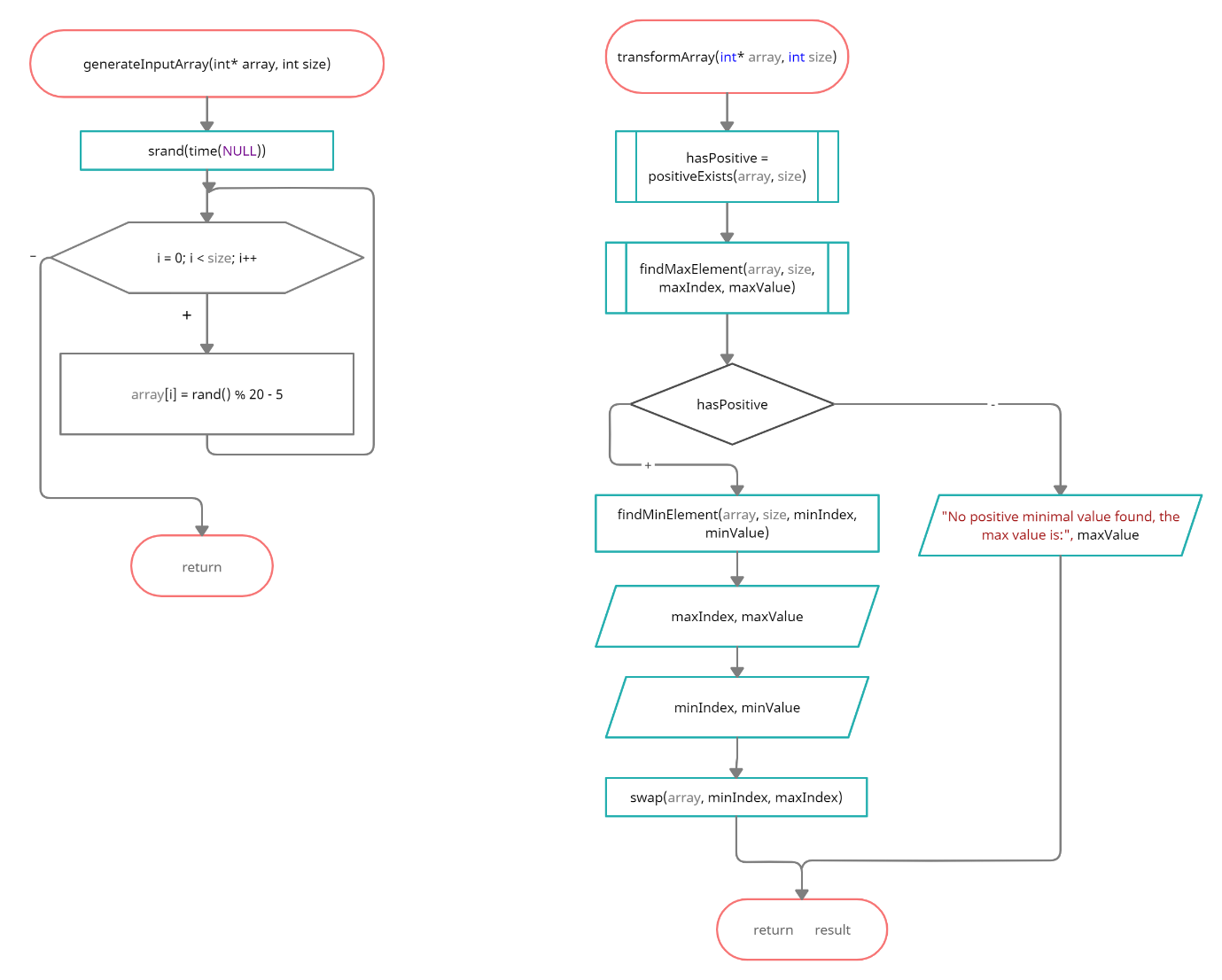
**Варіант 8**

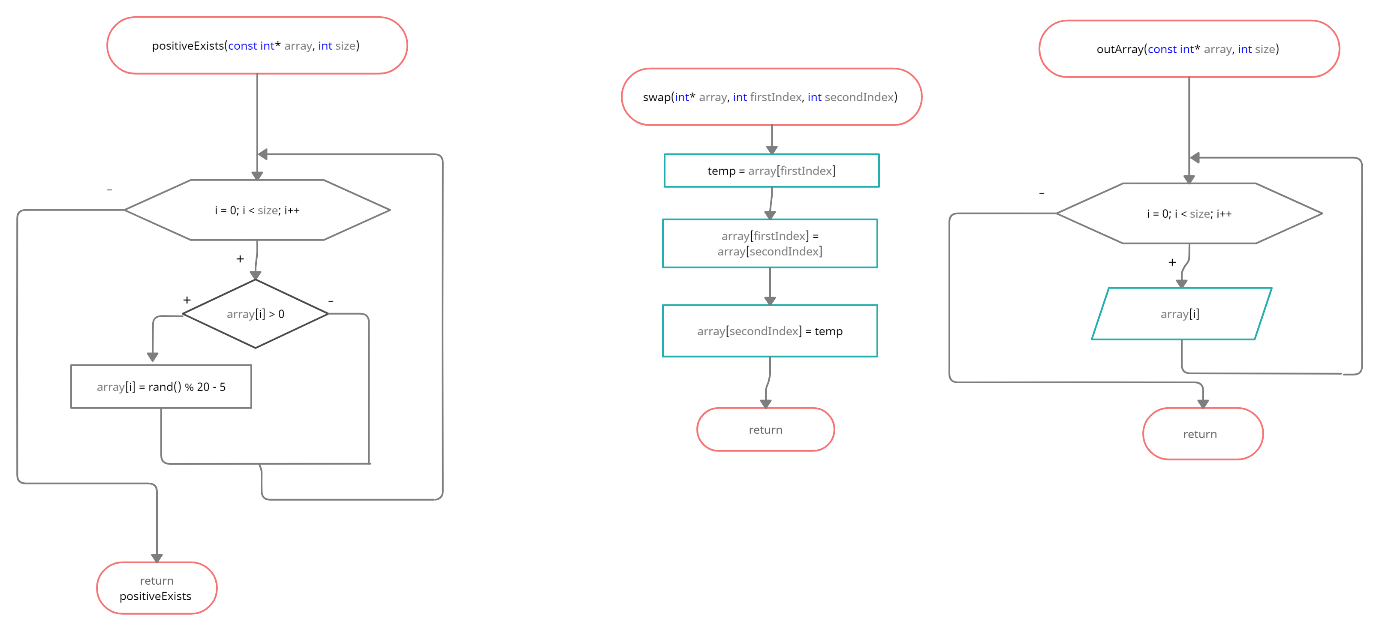
*Для заданого масиву цілих чисел D(n), серед яких є і від’ємні, знайти величину і індекс найменшого серед його додатних чисел. Поміняти місцями знайдений і максимальний елемент масиву (перше входження).*

***Основна функція:***

******

***Підпрограми:***

**

**

**С++**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <iomanip>

using namespace std;

//=========Прототипи функцій============

void generateInputArray(int\* array, int size); // прототип функції, що генерує вхідний масив

void transformArray(int\* array, int size); // прототип функції, що трансформує масив згідно до умови

bool positiveExists(const int\* array, int size); // прототип функції, що перевіряє чи в масиві існує хоча б один додатній елемент

void findMinElement(const int\* array, int size, int& minIndex, int& minValue); // прототип функції, що шукає мінімальний елемент

void findMaxElement(const int\* array, int size, int& maxIndex, int& maxValue); // прототип функції, що шукає максимальний елемент

void swap(int\* array, int firstIndex, int secondIndex); // прототип функції, що переставляє місцями два елементи масиву

void outArray(const int\* array, int size); // прототип функції, що виводить масив

//=========Головна функція============

int main() {

int n = 0;

cout << "Enter size: "; // розмір масиву

cin >> n;

int \*array = new int[n]; // одомірний масив

generateInputArray(array, n);

outArray(array, n);

cout << endl;

transformArray(array, n);

outArray(array, n);

return 0;

}

//=========Визначення функцій============

void generateInputArray(int\* array, int size) {

srand(time(NULL));

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

array[i] = rand() % 20 - 5;

}

}

void transformArray(int\* array, int size) {

int minIndex, minValue;

int maxIndex, maxValue;

bool hasPositive = positiveExists(array, size);

findMaxElement(array, size, maxIndex, maxValue);

if (hasPositive) {

findMinElement(array, size, minIndex, minValue);

cout << "Max is: array[" << maxIndex << "] = " << maxValue << endl;

cout << "Min is: array[" << minIndex << "] = " << minValue << endl;

cout << "Swapping min and max values" << endl;

swap(array, minIndex, maxIndex);

}

else {

cout << "No positive minimal value found, the max value is: " << maxValue << endl;

}

}

bool positiveExists(const int\* array, int size) {

bool positiveExists;

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

if (array[i] > 0) {

positiveExists = true;

}

}

return positiveExists;

}

void findMinElement(const int\* array, int size, int& minIndex, int& minValue) {

if (array[0] > 0) minIndex = 0;

else {

bool checkMin = false;

int i = 0;

while (checkMin != true)

{

if (array[i] > 0) {

minIndex = i;

checkMin = true;

}

i++;

}

}

minValue = array[minIndex];

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

if (array[i] < minValue && array[i] > 0) {

minValue = array[i];

minIndex = i;

}

}

}

void findMaxElement(const int\* array, int size, int& maxIndex, int& maxValue) {

maxIndex = 0;

maxValue = array[maxIndex];

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

if (array[i] > maxValue) {

maxValue = array[i];

maxIndex = i;

}

}

}

void swap(int\* array, int firstIndex, int secondIndex) {

int temp = array[firstIndex];

array[firstIndex] = array[secondIndex];

array[secondIndex] = temp;

}

void outArray(const int\* array, int size) {

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

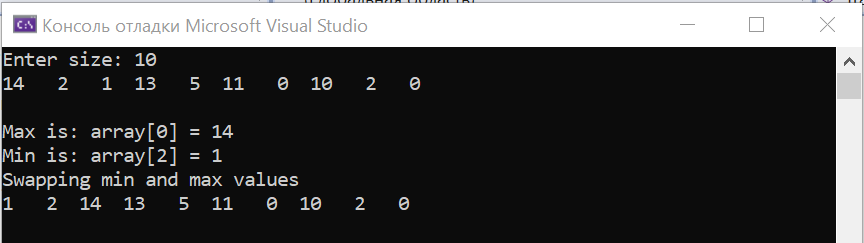
cout << array[i] << setw(4);

}

cout << endl;

}

**Відеокопія результату:**



**Висновок:** ми вивчили особливості обробки одновимірних масивів. Основною особливістю масивів є прямий доступ до елементів: всі компоненти масиву можуть вибиратись довільно і однаково доступні.