Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №7 з дисципліни

«Основи програмування»

«Одновимірні масиви»

Варіант 34

Виконав студент ІП-1134 Шамков Іван Дмитрович

( прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №7

Одновимірні масиви

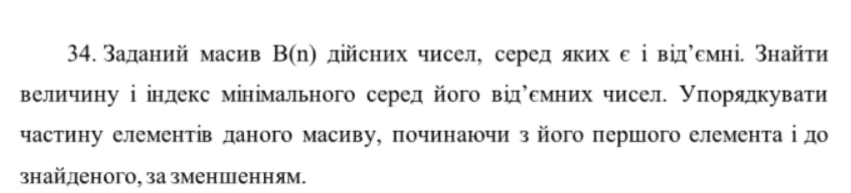
Лабораторна робота 7

Одновимірні масиви

*Мета* – вивчити особливості обробки одновимірних масивів.

Варіант: 34

*Умова задачі:*



*Математична модель:*

| Змінна | Тип | Ім’я | Призначення |
| --- | --- | --- | --- |
| Кількість елементів масиву | Цілий | num | Початкове дане |
| Масив | Дійсний | arrB | Початкове дане та Результат |
| Номер мінімального за модулем елементу масиву | Цілий | minid | Проміжне значення |
| Мінімальний елемент масиву | Дійсний | minelem | Результат |
| Змінна для збереження тимчасового значення під час сортування | Дійсний | tmp | Проміжне значення |
| Лічильник | Цілий | i | Проміжне значення |
| Лічильник | Цілий | k | Проміжне значення |

Постановка задачі:

Отже, математичне формулювання нашої задачі полягає в тому, щоб створити масив B із заданим розміром n, який складатиметься з дійсних чисел. У масиві шукаємо мінімальний за модулем елемент, тобто такий, що є найближчим до нуля. Після цього запам’ятовуємо його індекс. Проводимо сортування від нульового елементу масиву і до мінімального елементу за модулем, індекс якого до цього знайшли.У ході задачі виводимо на екран спочатку утворений масив, потім знайдений мінімальний за модулем елемент та його номер, а в кінці відсортований певним чином масив.

*Текст:*

*#include <iostream>*

*#include <ctime>*

*using namespace std;*

*void CreateArray(int, double\*);*

*int FindMin(double\*, int);*

*void SortBubble(double\*, int);*

*void CoutArr(double\*, int);*

*int main()*

*{*

*int num;*

*double minelem;*

*srand(time(NULL));*

*cout << "Enter the quantity of numbers: ";*

*cin >> num;*

*double\* arrB = new double[num];*

*CreateArray(num, arrB);*

*CoutArr(arrB, num);*

*int minid=FindMin(arrB, num);*

*SortBubble(arrB, minid+1);*

*CoutArr(arrB, num);*

*delete[] arrB;*

*return 0;*

*}*

*int FindMin(double\* arrB, int num)*

*{*

*int minid=0;*

*double minelem=arrB[0];*

*for (int i = 0; i < num; i++)*

*{*

*if (fabs(minelem) >= fabs(arrB[i]))*

*{*

*minid = i;*

*minelem = arrB[i];*

*}*

*}*

*cout << "\n\t\tMinimum element is " << minelem << " and it's id is: " << minid;*

*return minid;*

*}*

*void CreateArray(int num, double\* arrB)*

*{*

*for (int i=0; i < num; i++)*

*{*

*arrB[i]= (double(rand()%2000) - 1000)/10.0;*

*}*

*return;*

*}*

*void SortBubble(double\* ptrArr, int n) {*

*double tmp;*

*for (int k = 0; k < n; k++) {*

*for (int i = 0; i < n-1; i++) {*

*if (\*(ptrArr + i) < \*(ptrArr + (i + 1))) {*

*tmp = \*(ptrArr + (i + 1));*

*\*(ptrArr + (i + 1)) = \*(ptrArr + i);*

*\*(ptrArr + i) = tmp;*

*}*

*}*

*}*

*return;*

*}*

*void CoutArr(double\* ptrArr, int n) {*

*cout << "\n\n"<<"Array: "<< "[";*

*for (int i = 0; i < n; i++) {*

*cout << \*(ptrArr + i) << (i==n-1 ? "" : ", ");*

*}*

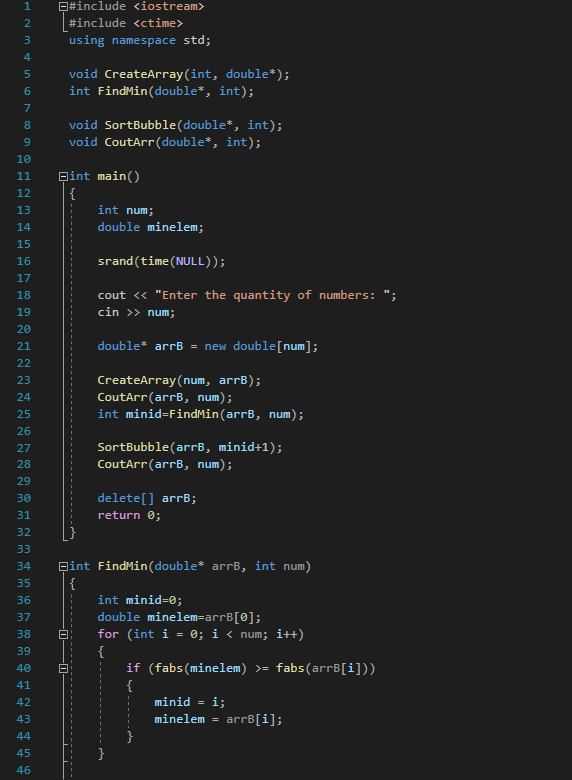
*cout << "]" << endl;*

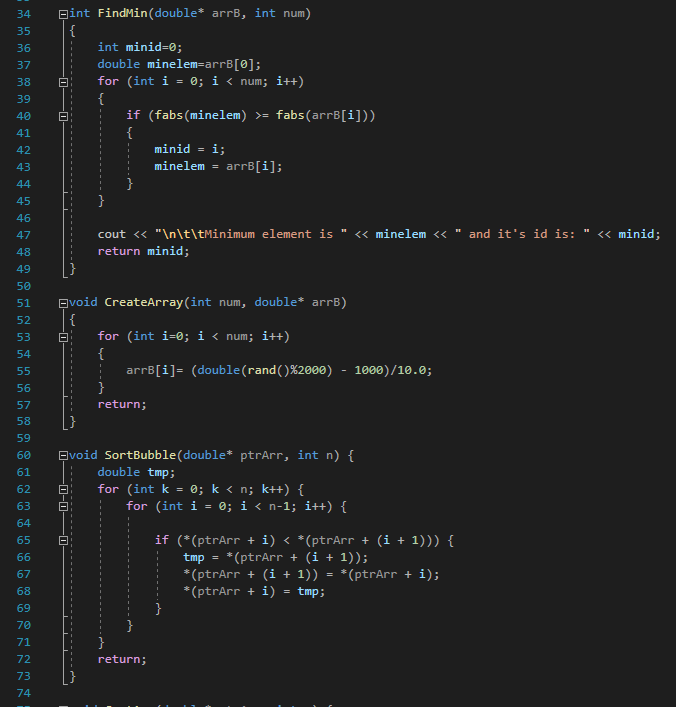
*return;*

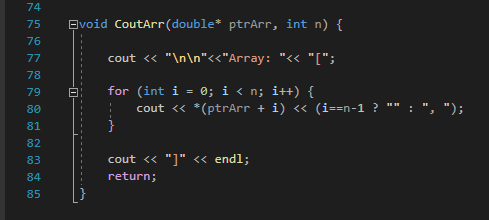
*}*

*Текст файла проекту:*

**C++:**

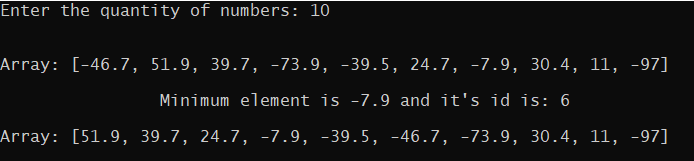


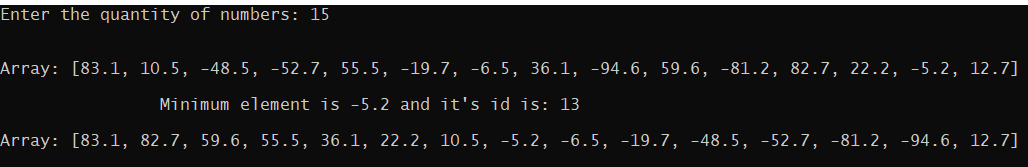


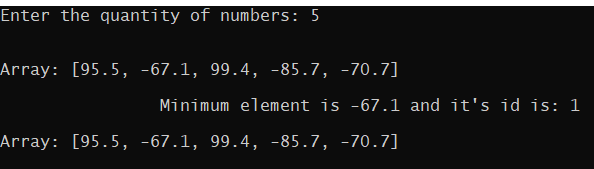


*Копії екранних форм:*

**C++:**







*Висновок*

Отже, виконавши цю лабораторну роботу, ми навчилися працювати з одновимірними масивами. Проекти, на мою думку, розроблені коректно, адже заплановані елементи працюють, а саме: отримання даних від користувача, виклик функцій пошуку найменшого елемента масива та його індекса, передача цього значення іншій функції для сортування масиву методом бульбашки, а також виведення усіх цих значень на екран. Масив створюється автоматично через підпрограму CreateArray(int, double\*), яка отримує значення довжини масива, а також посилання на його перший елемент, що записується в покажчик. Наповнюємо масив елементами з діапазону від -1000 до 1000, що ділимо на 10, щоб отримувати не цілі числа.