

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:*

34. Заданий масив  $B(n)$  дійсних чисел, серед яких є і від'ємні. Знайти величину і індекс мінімального серед його від'ємних чисел. Упорядкувати частину елементів даного масиву, починаючи з його першого елемента і до знайденого, за зменшенням.

*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 В 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 елемента масиву і до мінімального елемента 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 проекта:

C++:

```
1  #include <iostream>
2  #include <ctime>
3  using namespace std;
4
5  void CreateArray(int, double*);
6  int FindMin(double*, int);
7
8  void SortBubble(double*, int);
9  void CoutArr(double*, int);
10
11 int main()
12 {
13     int num;
14     double minelem;
15
16     srand(time(NULL));
17
18     cout << "Enter the quantity of numbers: ";
19     cin >> num;
20
21     double* arrB = new double[num];
22
23     CreateArray(num, arrB);
24     CoutArr(arrB, num);
25     int minid=FindMin(arrB, num);
26
27     SortBubble(arrB, minid+1);
28     CoutArr(arrB, num);
29
30     delete[] arrB;
31     return 0;
32 }
33
34 int FindMin(double* arrB, int num)
35 {
36     int minid=0;
37     double minelem=arrB[0];
38     for (int i = 0; i < num; i++)
39     {
40         if (fabs(minelem) >= fabs(arrB[i]))
41         {
42             minid = i;
43             minelem = arrB[i];
44         }
45     }
46 }
```

```

34 int FindMin(double* arrB, int num)
35 {
36     int minid=0;
37     double minelem=arrB[0];
38     for (int i = 0; i < num; i++)
39     {
40         if (fabs(minelem) >= fabs(arrB[i]))
41         {
42             minid = i;
43             minelem = arrB[i];
44         }
45     }
46
47     cout << "\n\t\tMinimum element is " << minelem << " and it's id is: " << minid;
48     return minid;
49 }
50
51 void CreateArray(int num, double* arrB)
52 {
53     for (int i=0; i < num; i++)
54     {
55         arrB[i]= (double(rand())%2000) - 1000)/10.0;
56     }
57     return;
58 }
59
60 void SortBubble(double* ptrArr, int n) {
61     double tmp;
62     for (int k = 0; k < n; k++) {
63         for (int i = 0; i < n-1; i++) {
64
65             if (*(ptrArr + i) < *(ptrArr + (i + 1))) {
66                 tmp = *(ptrArr + (i + 1));
67                 *(ptrArr + (i + 1)) = *(ptrArr + i);
68                 *(ptrArr + i) = tmp;
69             }
70         }
71     }
72     return;
73 }
74

```

```

74
75 void CoutArr(double* ptrArr, int n) {
76
77     cout << "\n\n" << "Array: " << "[";
78
79     for (int i = 0; i < n; i++) {
80         cout << *(ptrArr + i) << (i==n-1 ? "" : ", ");
81     }
82
83     cout << "]" << endl;
84     return;
85 }

```



*Копії екранних форм:*

**C++:**

```
Enter the quantity of numbers: 10

Array: [-46.7, 51.9, 39.7, -73.9, -39.5, 24.7, -7.9, 30.4, 11, -97]
        Minimum element is -7.9 and it's id is: 6
Array: [51.9, 39.7, 24.7, -7.9, -39.5, -46.7, -73.9, 30.4, 11, -97]
```

```
Enter the quantity of numbers: 15

Array: [83.1, 10.5, -48.5, -52.7, 55.5, -19.7, -6.5, 36.1, -94.6, 59.6, -81.2, 82.7, 22.2, -5.2, 12.7]
        Minimum element is -5.2 and it's id is: 13
Array: [83.1, 82.7, 59.6, 55.5, 36.1, 22.2, 10.5, -5.2, -6.5, -19.7, -48.5, -52.7, -81.2, -94.6, 12.7]
```

```
Enter the quantity of numbers: 5

Array: [95.5, -67.1, 99.4, -85.7, -70.7]
        Minimum element is -67.1 and it's id is: 1
Array: [95.5, -67.1, 99.4, -85.7, -70.7]
```

### *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.