


```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>           //Для быстрой сортировки
4  #include <chrono>             //для замера времени
5  #include <fstream>            //файлы
6  #include "Array.h"
7
8  using namespace std;
9
10
11 int main()
12 {
13     auto start_program = chrono::high_resolution_clock::now();    // время начала работы программы
14
15     ofstream write("Result.txt");    //файлик с результатом
16
17
18
19     system("chcp 1251");    //русский язык
20
21     cout << "Введите размер массива:";
22     write << "Введите размер массива:";
23     int size;
24     cin >> size;
25     write << size << "\n";
26     vector<float> arr(size);    //создаем массив на size элементов
27
28     fill_random(arr, -5, 10);    //заполняем случайными числами
29     print_arr(arr);    //выводим на экран
30     write_tofile(arr, write);    //и в файл
31
32     vector<float> arr_copy(size);    //создаем копию массива
33     arr_copy = arr;
34     cout << "\nОтсортированный массив:\n";
35     write << "\nОтсортированный массив:\n";
36     sort(arr.begin(), arr.end());    //сортируем
37
38     print_arr(arr);    //выводим на экран
39     write_tofile(arr, write);    //и в файл
40
41     //////////////////////////////////// Поиск в диапазоне ////////////////////////////////////
42
43     cout << "\n\nВведите диапазон поиска\nВведите левую границу поиска:";
44     write << "\n\nВведите диапазон поиска\nВведите левую границу поиска:";
45     float left;
46     cin >> left;
47     write << left << "\n";
48     cout << "Введите правую границу поиска:";
```

```

49     write << "Введите правую границу поиска:";
50     float right;
51     cin >> right;
52     write << right << "\n";
53
54     cout << "\nКол-во элементов находящихся в данном диапазоне:";
55     write << "\nКол-во элементов находящихся в данном диапазоне:";
56
57     cout << "\nИндекс левой границы:" << find_left(arr, left);
58     cout << "\nИндекс правой границы:" << find_right(arr, right);
59     cout << "\nКол-во элементов внутри заданного диапазона:" <<      ↗
        count_in_range(arr, left, right);
60
61     write << "\nИндекс левой границы:" << find_left(arr, left);
62     write << "\nИндекс правой границы:" << find_right(arr, right);
63     write << "\nКол-во элементов внутри заданного диапазона:" <<      ↗
        count_in_range(arr, left, right);
64
65     ////////////////////////////////////// Сумма эл-в после максимального  \\\\\\\ ↗
        \\\\\\\
66
67     cout << "\n\nТ.к. массив отсортирован, то сумма элементов после      ↗
        максимального равна 0,\n" <<
68         "но если брать массив до сортировки, то вот:\n";
69     write << "\n\nТ.к. массив отсортирован, то сумма элементов после      ↗
        максимального равна 0,\n" <<
70         "но если брать массив до сортировки, то вот:\n";
71
72     auto start = chrono::high_resolution_clock::now      ↗
        ();          //замеряем время выполнения функции      ↗
        суммирования(начало работы ф-ии
73     float summ = sum_after_value(arr_copy, arr[arr.size() -      ↗
        1]);          //суммируем
74     auto end = chrono::high_resolution_clock::now();      // ↗
        конец работы ф-ии
75
76     chrono::duration<float> duration = end - start;
77     cout << "Сумма элементов после максимального значения:" <<      ↗
        summ ; //передаем неотсортированный массив и значение последнего      ↗
        элемента в отсортированном массиве, тк оно максимальное
78     write << "Сумма элементов после максимального значения:" << summ;
79     printf_s("\nВремя работы ф - ии суммирования :%.7f", duration.count      ↗
        ());
80     write << "\nВремя работы ф - ии суммирования :" << duration.count();
81
82     cout << "\n\nСортировка по убыванию модулей:\n";
83     write << "\n\nСортировка по убыванию модулей:\n";
84     sort_decrease_abs(arr);
85     print_arr(arr);
86     write_to_file(arr, write);
87     auto end_program = chrono::high_resolution_clock::now      ↗
        ();          //время конца работы программы
88     duration = end_program -      ↗

```

```
        start_program;                                //вычитаем из времени 
        конца работы программы, начало, таким образом находим время работы
89     cout << "\nВремя работы:" << duration.count();
90     write << "\nВремя работы:" << duration.count();
91 }
92
93
94
```