```cpp
#pragma once
#include <vector>
#include <time.h>
#include <fstream>
#include <iostream>
using namespace std;

template<typename T>
void fill_random(vector<T>& arr, int left, int right);      //Заполнение массива случайными числами

template<typename T>
void print_arr(vector<T>& arr);                             //Вывод массива на экран

template<typename T>
int count_in_range(vector<T>& arr, T left, T right);        //Кол-во элементов массива находящиеся в данном диапозоне

template<typename T>
int find_left(vector<T>& arr, T left_value);                //Бинарный поиск левого индекса

template<typename T>
int find_right(vector<T>& arr, T right_value);              //Бинарный поиск правого индекса

template<typename T>
T sum_after_value(vector<T>& arr, T value);                 //Сумма элементов массива после заданного значения

template<typename T>
void sort_decrease_abs(vector<T>& arr);                     //Сортировка массива по убыванию модулей

template<typename T>
void write_tofile(vector<T>& arr, ofstream& f);     //запись массива в файл

////////////////////////////////////  Реализация  \\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

template<typename T>
void fill_random(vector<T>& arr, int left, int right)   //Заполнение массива случайными числами
{
    srand(time(NULL));
    for (int i = 0; i < arr.size(); i++)
    {
        float n = rand() % (right * 100);

        if (left < 0)                                   //Установка левой границы
```

```
43          {
44              arr[i] = 0.01 * n - rand() % left;          //Если граница меньше 0, то сдвигаем влево
45          }
46          else
47          {
48              arr[i] = 0.01 * n + rand() % left;          //Если больше то вправо
49          }
50      }
51  }
52
53  template<typename T>
54  void print_arr(vector<T>& arr)                          //Вывод массива на экран
55  {
56      for (int i = 0; i < arr.size(); i++)
57      {
58          std::cout << arr[i] << "  ";
59          if (i % 9 == 0 && i != 0)
60          {
61              std::cout << "\n";
62          }
63      }
64  }
65
66  template<typename T>
67  int count_in_range(vector<T>& arr, T left, T right)     //Кол-во элементов массива находящиеся в данном диапозоне
68  {
69      if (left > arr[arr.size() - 1] || right < arr[0] || left > right)
70      {
71          return 0;
72      }
73      else
74      {
75          int l = find_left(arr, left);       //находим значение индекса левого границы
76          int r = find_right(arr, right);     //находим значение индекса правой границы
77
78          if (r == l && arr[r] <= right && arr[r] >= left)
79          {
80              return 1;
81          }
82          else
83          {
84              return r - l + 1;               //возвращаем их разницу
85          }
86
87      }
88  }
```

```
 89
 90 template<typename T>
 91 int find_left(vector<T>& arr, T left_value)             //Бинарный ↵
      поиск левого индекса
 92 {
 93     int left = 0, right = arr.size() - 1;              //устанавливаем ↵
          изначальные границы
 94
 95     while (left < right)                               //пока они не ↵
         пересекутся будем искать
 96     {
 97         int mid = (left + right) / 2;                  //находим центр
 98
 99         if (arr[mid] >= left_value)                    //если искомое ↵
           значение больше центрального
100         {
101             right = mid;                               //то сдвигаем ↵
                правую границу
102         }
103         else
104         {
105             left = mid + 1;                            //если нет, то ↵
                левую
106         }
107     }
108     return left;
109 }
110
111 template<typename T>
112 int find_right(vector<T>& arr, T right_value)          //Бинарный ↵
      поиск правого индекса
113 {
114     int left = 0, right = arr.size() - 1;              //устанавливаем ↵
          изначальные границы
115
116     while (left < right)                               //пока они не ↵
         пересекутся будем искать
117     {
118         int mid = (left + right + 1) / 2;              //находим центр
119
120         if (arr[mid] <= right_value)                   //если искомое ↵
           значение больше центрального
121         {
122             left = mid;                                //то сдвигаем ↵
                правую границу
123         }
124         else
125         {
126             right = mid - 1;                           //если нет, то ↵
                левую
127         }
128     }
129     return left;
```

```
130 }
131
132 template<typename T>
133 T sum_after_value(vector<T>& arr, T value)              //Сумма элементов массива после заданного значения
134 {
135     T summ = 0;
136     if (value == arr[arr.size() - 1])                   //если элемент последний, то возвращаем 0
137     {
138         return 0.00;
139     }
140     else
141     {
142         int i = 0;
143         while (arr[i] < value)                          //Доходим до нужного элемента
144         {
145             i++;
146         }
147         i++;
148         for (; i < arr.size(); i++)
149         {
150             summ += arr[i];                             //считаем сумму
151         }
152         return summ;
153     }
154
155 }
156
157
158 template<typename T>
159 void sort_decrease_abs(vector<T>& arr)                  //Сортировка массива по убыванию модулей
160 {
161
162     for (int i = 0; i < arr.size(); i++)
163     {
164         int i_max = i;
165         for (int j = i; j < arr.size(); j++)                //ищем индекс самого максимального числа
166         {
167             if (abs(arr[j]) > abs(arr[i_max]))              //если j элемент больше i_max-го элемента (по модулю), то запоминаем его
168             {
169                 i_max = j;
170             }
171         }
172         swap(arr[i], arr[i_max]);
173     }
174 }
175
```

```
176 template<typename T>
177 void write_tofile(vector<T>& arr, ofstream& f)          //запись
      массива в файл
178 {
179     for (int i = 0; i < arr.size(); i++)
180     {
181         f << arr[i] << "  ";
182         if (i % 9 == 0 && i != 0)
183         {
184             f << "\n";
185         }
186     }
187 }
188
```