

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
федеральное государственное автономное образовательное  
учреждение высшего образования  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники  
Отделение автоматизации и робототехники  
Направление мехатроника и робототехника

**Отчет**  
по лабораторной работе №2

по дисциплине  
«Основы программирования и алгоритмизации»

**Сортировки в C++**

Выполнил:

Студент группы 8E21

\_\_\_\_\_

Н.С. Моисеев

Проверил:

Ассистент ОАР ИШИТР

\_\_\_\_\_

Я.О. Кургинов

Томск 2023

## Вариант 10

### Цель работы:

Познакомиться с сортировками, используя C++.

### Задание 1

Реализовать алгоритм сортировки выбором для одномерного массива заполненного типом данных **float**.

#### Блок схема (Рисунок 1)

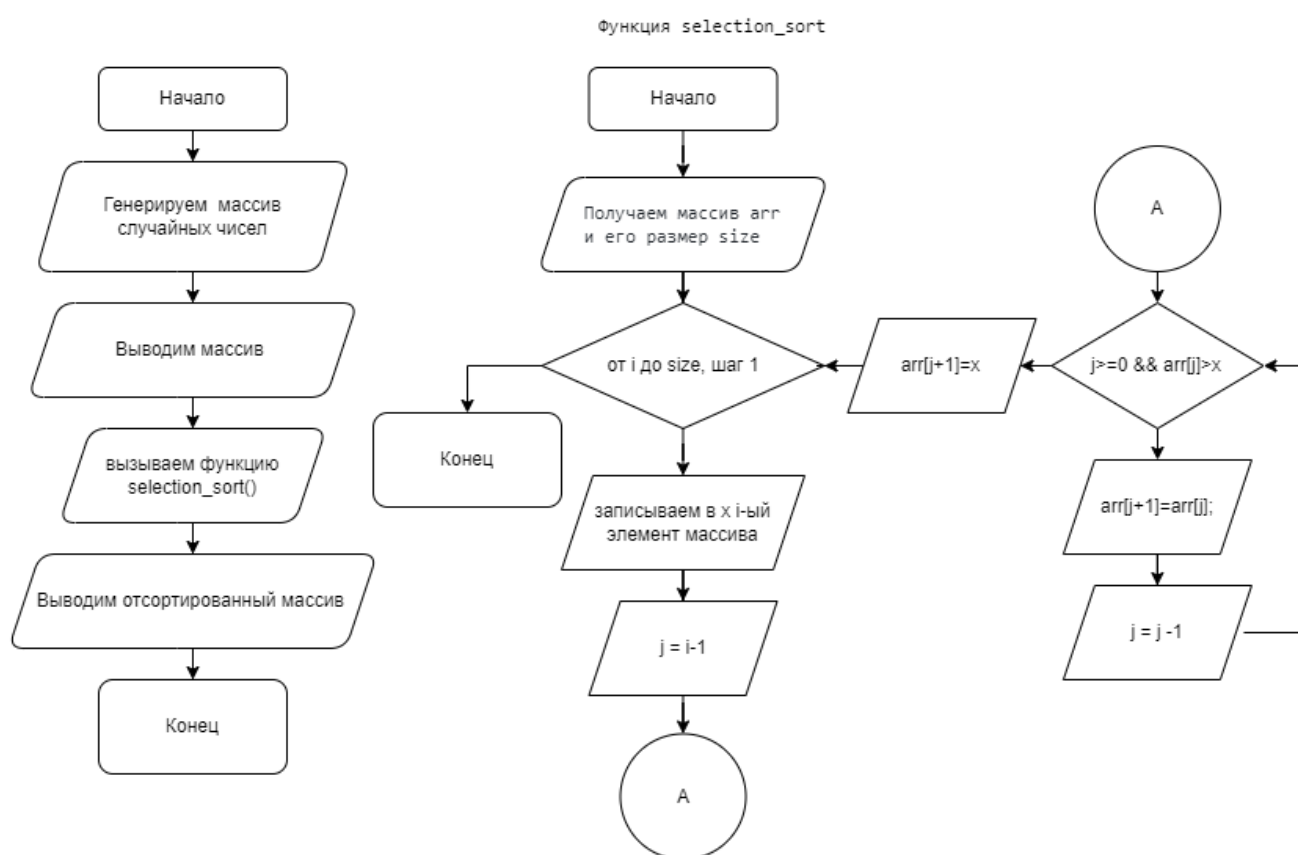


Рисунок 1 - Блок-схема к заданию 1.

Текст программы (Листинг 1, Приложение А)

Результаты работы (Рисунок 2)

Рисунок 2 - Скриншот результата работы программы к заданию 1 на консоли.

```
Введите размер массива: 7
Не отсортированный массив:
0.32 -0.32 0.54 -0.12 0.52 -0.56 0.08
Отсортированный массив:
-0.56 -0.32 -0.12 0.08 0.32 0.52 0.54
```

## Задание 2:

Реализовать алгоритм сортировки вставкой для одномерного массива заполненного типом данных **float**.

### Блок схема (Рисунок 3)

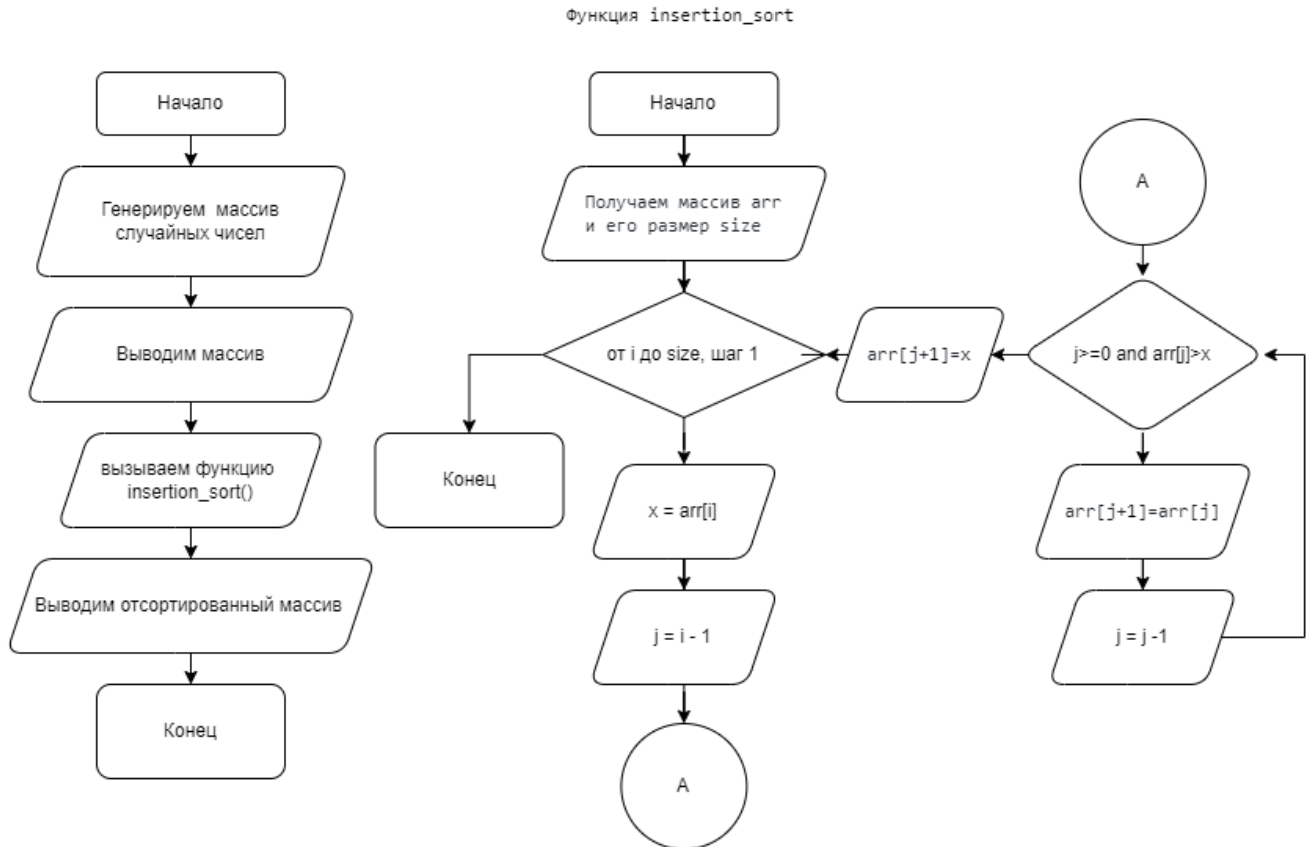


Рисунок 3 - Блок-схема к заданию 2.

Текст программы (Листинг 2, Приложение А)

Результаты работы (Рисунок 4)

```
Введите размер массива: 7
Не отсортированный массив:
8.97 -8.02 7.65 -9.92 0.01 -5.21 2.2
Отсортированный массив:
-9.92 -8.02 -5.21 0.01 2.2 7.65 8.97
```

Рисунок 4 - Скриншот результата работы программы к заданию 2 на консоли.

### Задание 3

Реализовать алгоритм быстрой сортировки для одномерного массива заполненного типом данных **float**.

#### Блок схема (Рисунок 5)

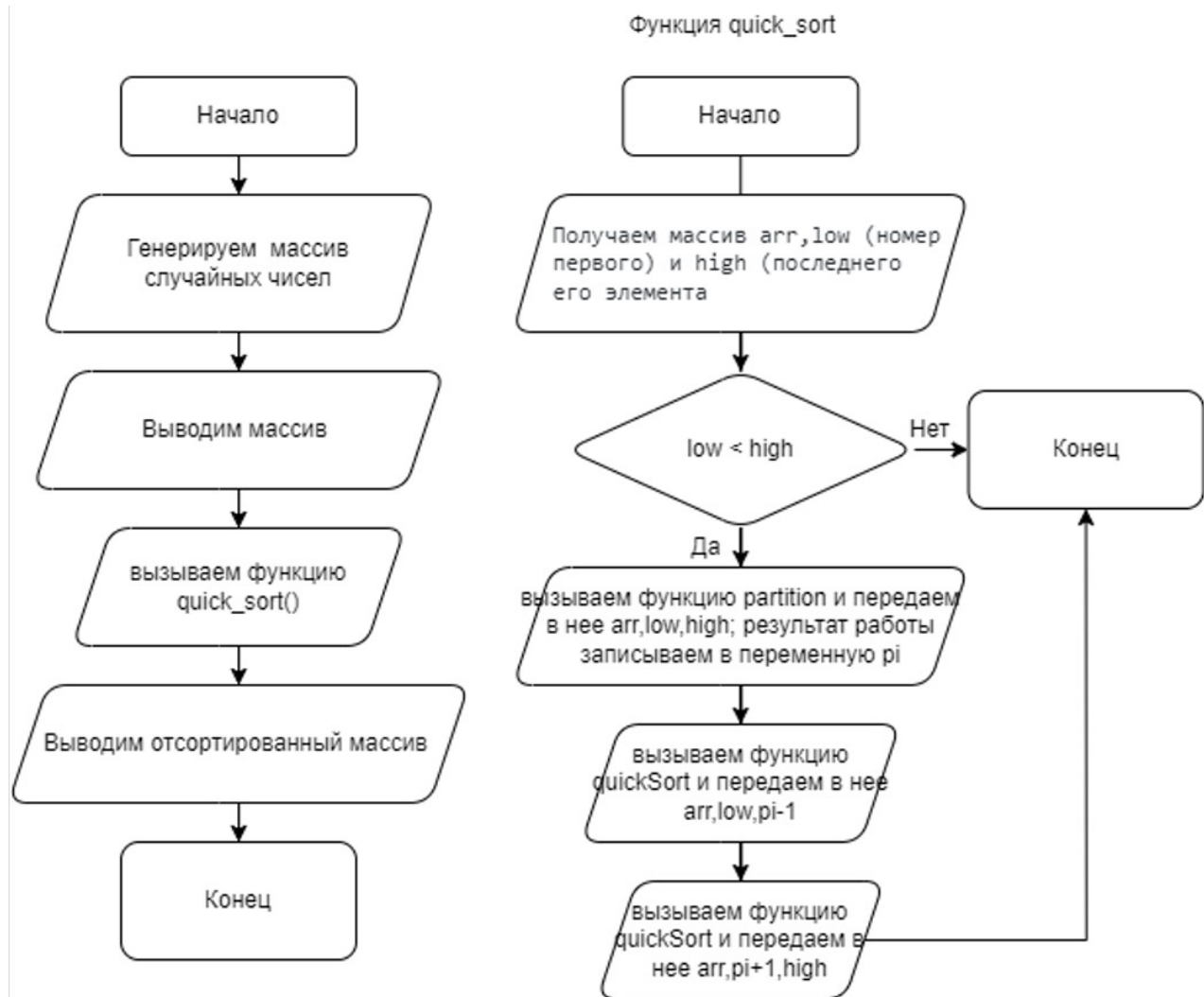


Рисунок 5 - Блок-схема к заданию 3.

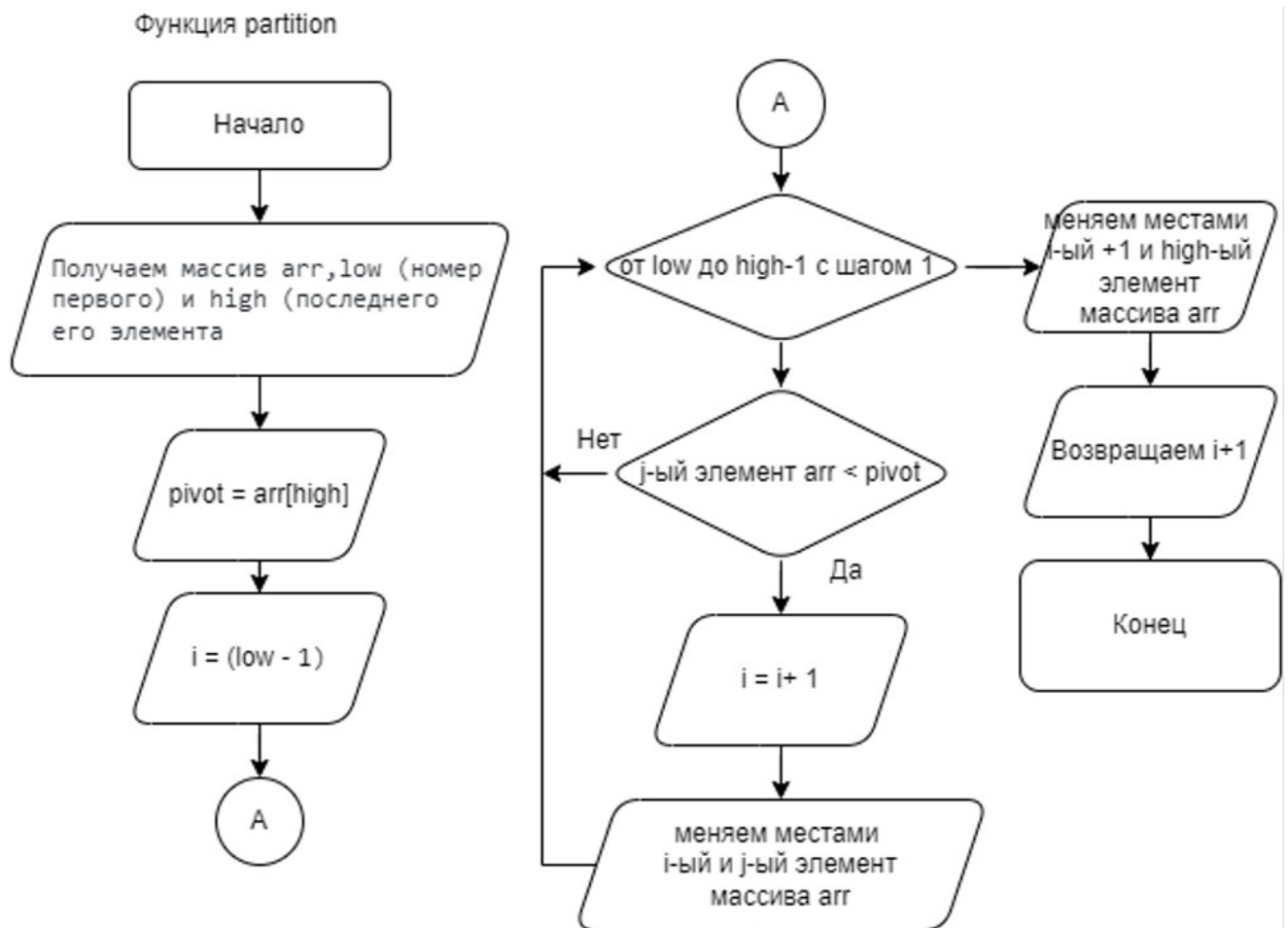


Рисунок 6 - Блок-схема к заданию 3.

Текст программы (Листинг 3, Приложение А)

**Результаты работы (Рисунок 6)**

```

Введите размер массива: 7
Не отсортированный массив:
0.32 -0.32 0.54 -0.12 0.52 -0.56 0.08
Отсортированный массив:
-0.56 -0.32 -0.12 0.08 0.32 0.52 0.54
  
```

Рисунок 7 - Скриншот результата работы программы к заданию 3 на консоли.

### Выводы:

В результате лабораторной работы были изучена Сортировка выбором, сортировка вставкой, сортировка быстрая.

## Приложение А

### Листинг 1 – Программа к заданию 1.

```
1. #include <iostream>
2.
3. void selection_sort(float *arr, int size)
4. {
5.     for (int i = 0; i < size - 1; i++)
6.     {
7.         int min_index = i;
8.         for (int j = i + 1; j < size; j++)
9.         {
10.            if (arr[j] < arr[min_index]) min_index = j;
11.        }
12.        if (min_index != i)
13.        {
14.            float temp = arr[i];
15.            arr[i] = arr[min_index];
16.            arr[min_index] = temp;
17.        }
18.    }
19. }
20.
21. int main(){
22.     int size; std::cout << "Введите размер массива: "; std::cin >> size;
23.     float randomDigits[size] {};
24.     for(int i = 0; i < size; i++){
25.         if(i%2 == 1){
26.             randomDigits[i] = -0.01 * (rand() % 101);
27.             continue;
```

```

28.         }
29.         randomDigits[i] = 0.01 * (rand() % 101);
30.     }
31.     std::cout << "Не отсортированный массив: " << std::endl;
32.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
33.     std::cout << std::endl;
34.     selection_sort(randomDigits, size);
35.     std::cout << "Отсортированный массив: " << std::endl;
36.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
37.     std::cout << std::endl;
38. }

```

Листинг 2 – Программа к заданию 2.

```

1. #include <iostream>
2.
3. void insertion_sort(float *arr, int size){
4.     for(int i=0; i<size; i++){
5.         float x=arr[i];
6.         int j=i-1;
7.         while(j>=0 && arr[j]>x){
8.             arr[j+1]=arr[j];
9.             j--;
10.        }
11.        arr[j+1]=x;
12.    }
13. }
14. int main(){
15.     int size; std::cout << "Введите размер массива: "; std::cin >> size;
16.     float randomDigits[size] {};
17.     for(int i = 0; i < size; i++){
18.         if(i%2 == 1){

```

```

19.         randomDigits[i] = -0.01 * (rand() % 1001);
20.         continue;
21.     }
22.         randomDigits[i] = 0.01 * (rand() % 1001);
23.     }
24.     std::cout << "Не отсортированный массив: " << std::endl;
25.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
26.     std::cout << std::endl;
27.     insertion_sort(randomDigits, size);
28.     std::cout << "Отсортированный массив: " << std::endl;
29.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
30.     std::cout << std::endl;
31. }

```

Листинг 3 – Программа к заданию 3.

```

1. #include <iostream>
2. int partition(float *arr, int low, int high){
3.     float pivot = arr[high];
4.     int i = (low - 1);
5.
6.     for (int j = low; j <= high - 1; j++) {
7.         if (arr[j] < pivot) {
8.             i++;
9.             float temp = arr[i];
10.            arr[i] = arr[j];
11.            arr[j] = temp;
12.        }
13.    }
14.    float temp = arr[i + 1];
15.    arr[i + 1] = arr[high];
16.    arr[high] = temp;

```



```

17.     return (i + 1);
18. }
19. void quickSort(float *arr, int low, int high){
20.     if (low < high) {
21.         int pi = partition(arr, low, high);
22.         quickSort(arr, low, pi - 1);
23.         quickSort(arr, pi + 1, high);
24.     }
25. }
26. int main()
27. {
28.     int size; std::cout << "Введите размер массива: "; std::cin >> size;
29.     float randomDigits[size] {};
30.     for(int i = 0; i < size; i++){
31.         if(i%2 == 1){
32.             randomDigits[i] = -0.01 * (rand() % 101);
33.             continue;
34.         }
35.         randomDigits[i] = 0.01 * (rand() % 101);
36.     }
37.     std::cout << "Не отсортированный массив: " << std::endl;
38.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
39.     std::cout << std::endl;
40.     quickSort(randomDigits, 0, size-1);
41.     std::cout << "Отсортированный массив: " << std::endl;
42.     for(int i = 0; i < size; i++) std::cout << randomDigits[i] << " ";
43.     std::cout << std::endl;
44.
45.     return 0;
46. }

```