**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет Информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 16-17**

**Дисциплина:**

Основы алгоритмизации и программирования

**Тема:**

Алгоритм сортировки «быстрая»

**Выполнил(а): студент(ка) группы 211-7210**

**Салов Д.К.**

(Фамилия И.О.)

**Дата, подпись** 07.12.21

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)[[1]](#endnote-1)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2021**

**Цель:**  Получить практические навыки разработке алгоритмов и их программной реализации.

**Задания:**

Необходимо выполнить и оформить описание следующих пунктов:

1. Сформулировать идею алгоритма

2. Выполнить словесное представление алгоритма

3. Выполнить полнить представление алгоритма с помощью блок схем с использованием элемента модификации и без него.

4. Выполнить программную реализацию алгоритмов на языке С с  
использованием параметрического цикла и цикла с предусловием.

**Описание:**

Вводим массив.

1. Выбирается “опорный” элемент массива. Это может быть любое число, но от выбора этого элемента сильно зависит эффективность алгоритма. Если нам известна медиана, то лучше выбирать элемент, который как можно ближе к медиане. В нашей реализации алгоритма, мы будем брать самый левый элемент, который в результате займет свое место.
2. Элементы в массиве делятся на две части: слева те, кто меньше опорного элемента, справа те, кто больше. Таким образом опорный элемент занимает свое место и больше никуда не двигается.
3. Для левого и правого массива действия повторяются рекурсивно.

Выводим массив.

**Код:**

#include <iostream>

using namespace std;

void qs(int\* arr, int L, int R) {

if (L < R) {

int left = L;

int right = R;

int middle = arr[(left + right) / 2];

do {

while (arr[left] < middle) left++; //сдвиг left вправо

while (arr[right] > middle) right--; //сдвиг right влево

if (left <= right) {

//перестановка

int tmp = arr[left];

arr[left] = arr[right];

arr[right] = tmp;

left++;

right--;

}

} while (left < right);

qs(arr, L, right);

qs(arr, left, R);

}

}

int main()

{

//заполнение массива

const int len = 10;

int arr[len];

for (int i = 0; i <= len - 1; i++)

{

cin >> arr[i];

}

qs(arr, 0, len - 1); //вызов сортировщика

// Вывод

for (int i = 0; i < len; i++) {

cout << arr[i] << " ";

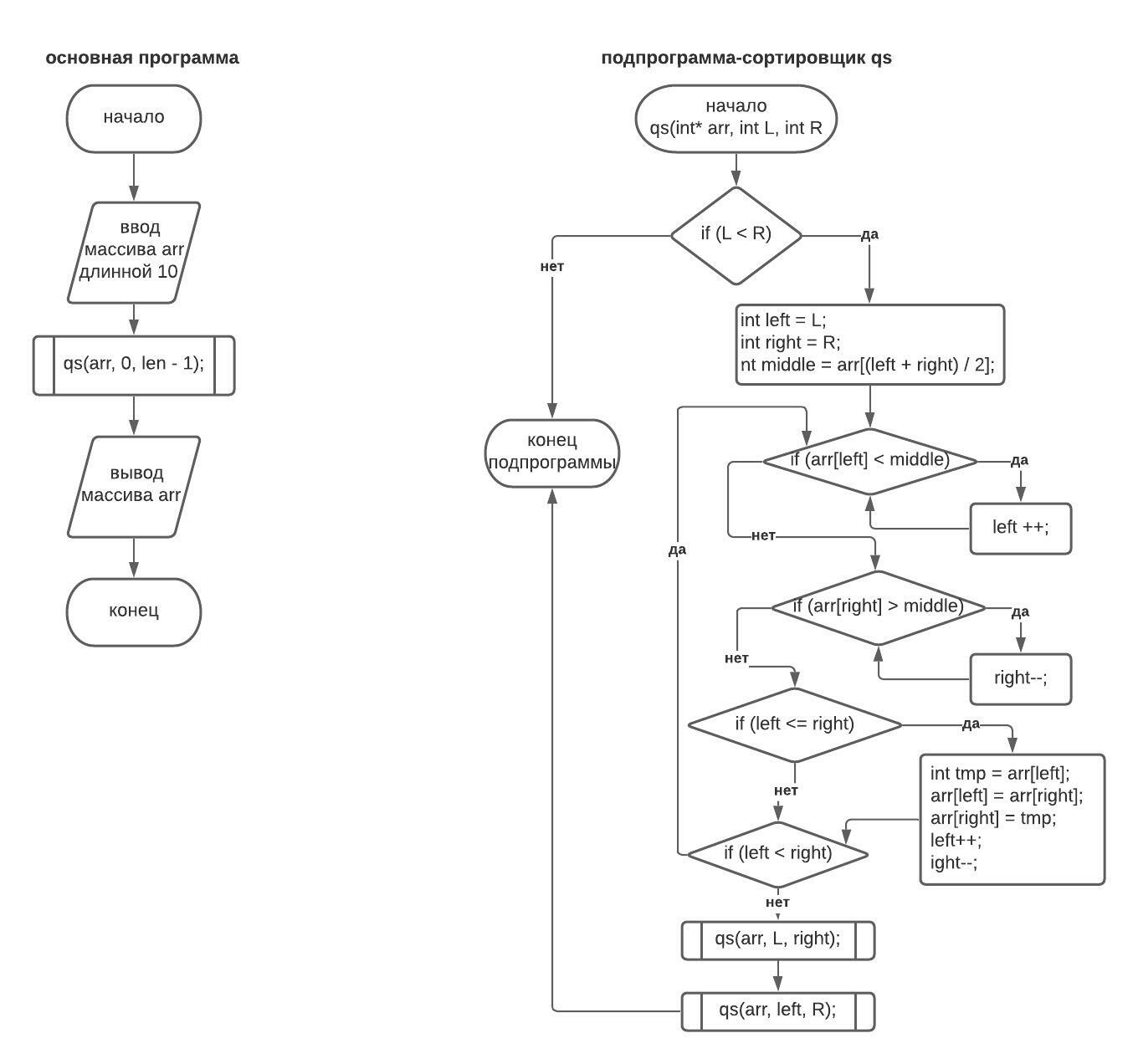
}

cout << "\n";

system("pause");

return 0;

}

**Блок-схемы:**

**Вывод:** Крутая сортировка.

1. [↑](#endnote-ref-1)