

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа № 14
по дисциплине
«Информатика»
на тему
«Сортировка Хоара»**

Выполнил студент гр. ИВТ-23-16

Южаков Федор Алексеевич

Проверил:

доцент кафедры ИТАС

Денис Владимирович Яруллин

(оценка)

(подпись)

(дата)

г. Пермь, 2024

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1 Вариант задания

Постановка задачи.

Выполнить сортировку массива сортировкой Хоара;

Массив статический и содержит 25 элементов, заполнение массива организовать любым удобным способом.

2 Анализ задачи

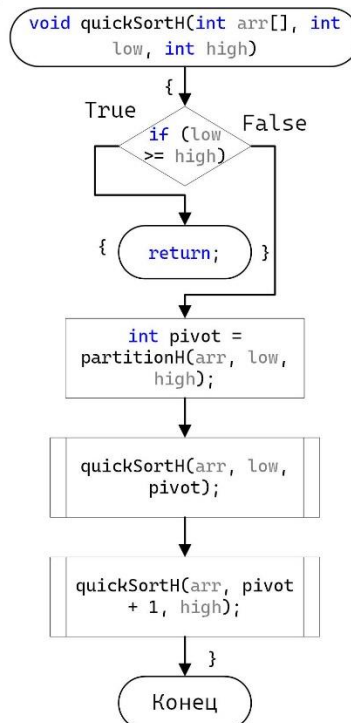
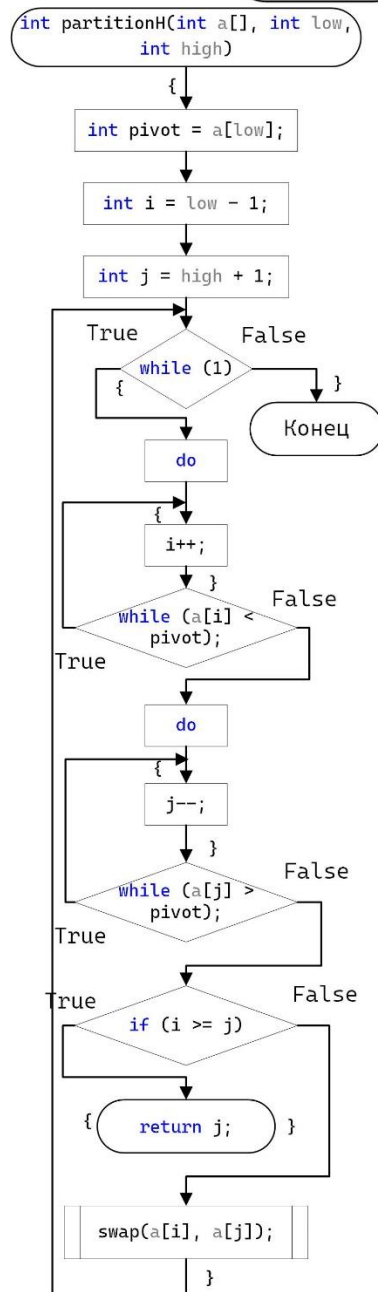
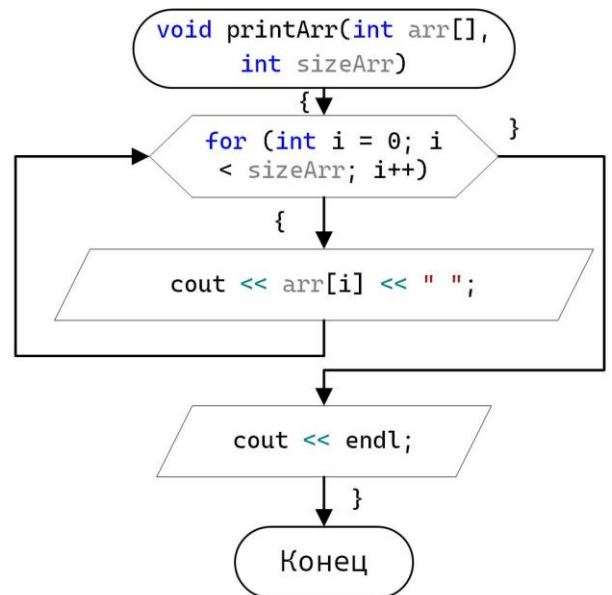
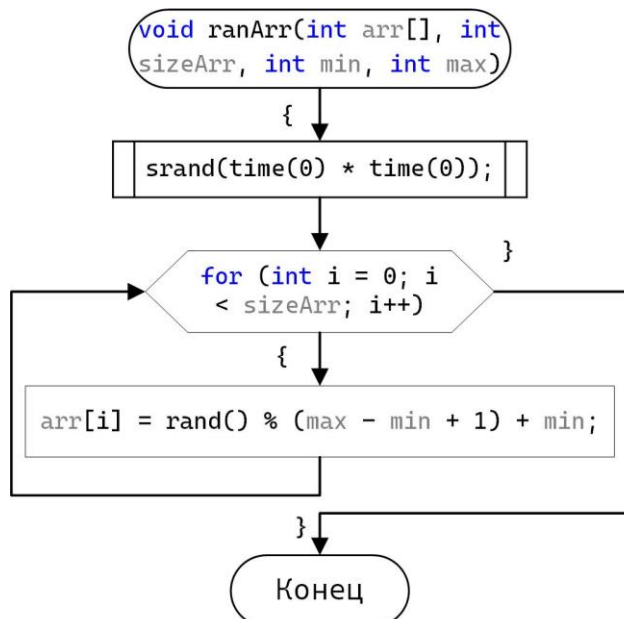
В данной задаче получаем от пользователя границы случайных чисел, запускаем функцию заполнения массива случайными числами в указанных границах. После работы этой функции выводим массив. Затем вызываем необходимую сортировку и снова выводим массив, на этот раз отсортированный.

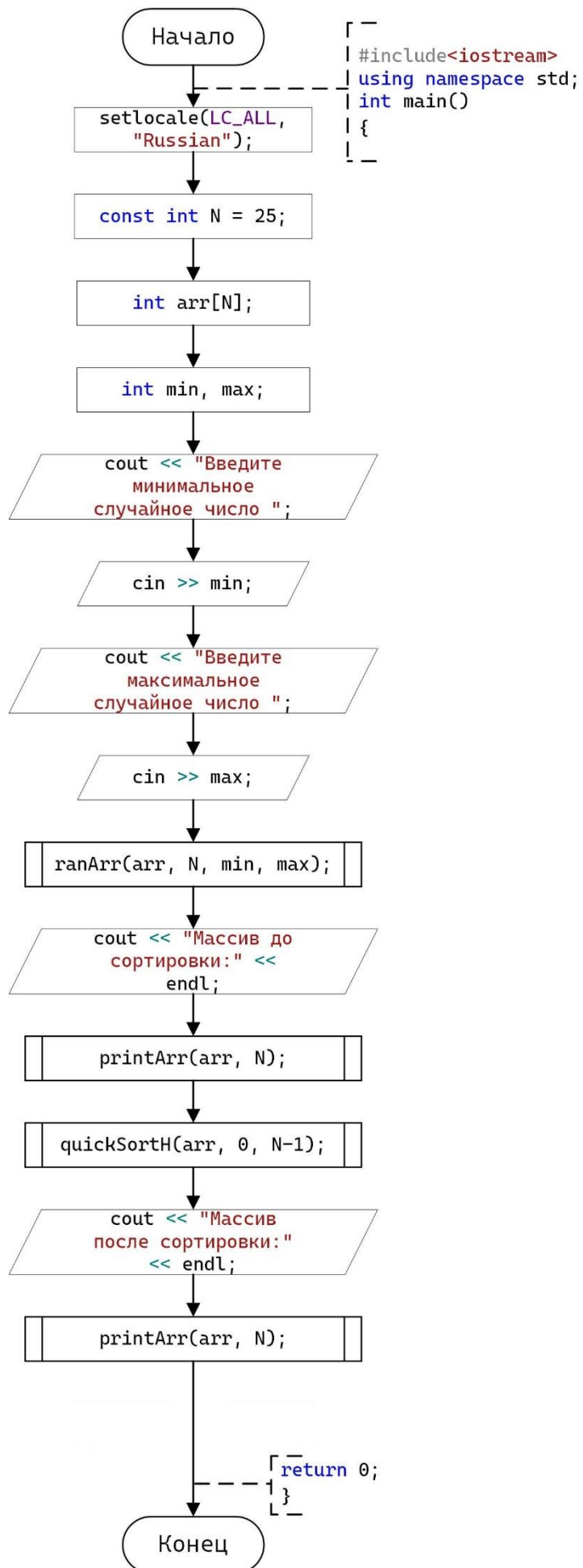
Общий алгоритм быстрых сортировок:

1. Если диапазон менее 2-х элементов, сразу же прекращаем действия, т.к. сортировать нечего.
2. Выбираем точку (*pivot*, значение), которая встречается в диапазоне. Конкретный принцип выбора такого значения зависит от выбранного извода и может включать элементы случайности (рандомизации).
3. Разбиваем диапазон на части: меняем порядок элементов, одновременно определяя точку деления таким образом, чтобы элементы со значением меньшим, чем *pivot*, шли до точки деления, тогда как элементы с большим значением — после точки деления. Элементы, равные значению *pivot*, могут записываться в любую часть (зависит от извода). Т.к. минимум одно значение равно *pivot*, большинство изводов стремятся точку деления приравнять к *pivot*.
4. Рекурсивно применяем быструю сортировку к каждой из полученных частей (до и после точки деления), пока алгоритм это позволяет (т.е. пока он не остановится на шаге 1)

3 Блок схема

Проанализировав задачу, составим подробную блок схему основной программы и использованных функций.





ПРАКТИЧЕСКАЯ ЧАСТЬ

4 Результат решения

4.1 Готовая программа

Исходя из подробных блок схем, составим программу на языке C++.

Таблица 1 – Готовая программа задачи

```
#include <iostream>
using namespace std;
void ranArr(int arr[], int sizeArr, int min, int max)
{
    srand(time(0) * time(0));
    for (int i = 0; i < sizeArr; i++)
    {
        arr[i] = rand() % (max - min + 1) + min;
    }
}
void printArr(int arr[], int sizeArr)
{
    for (int i = 0; i < sizeArr; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
int partitionH(int a[], int low, int high)
{
    int pivot = a[low];
    int i = low - 1;
    int j = high + 1;
    while (1)
    {
        do
        {
            i++;
        } while (a[i] < pivot);

        do
        {
            j--;
        } while (a[j] > pivot);

        if (i >= j) return j;
        swap(a[i], a[j]);
    }
}
void quickSortH(int arr[], int low, int high)
{
    if (low >= high) return;
    int pivot = partitionH(arr, low, high);
    quickSortH(arr, low, pivot);
    quickSortH(arr, pivot + 1, high);
}
int main()
{
    setlocale(LC_ALL, "Russian");
    const int N = 25; // если изменить, работает и с другими размерами
    int arr[N];
    int min, max;
    cout << "Введите минимальное случайное число ";
    cin >> min;
    cout << "Введите максимальное случайное число ";
```

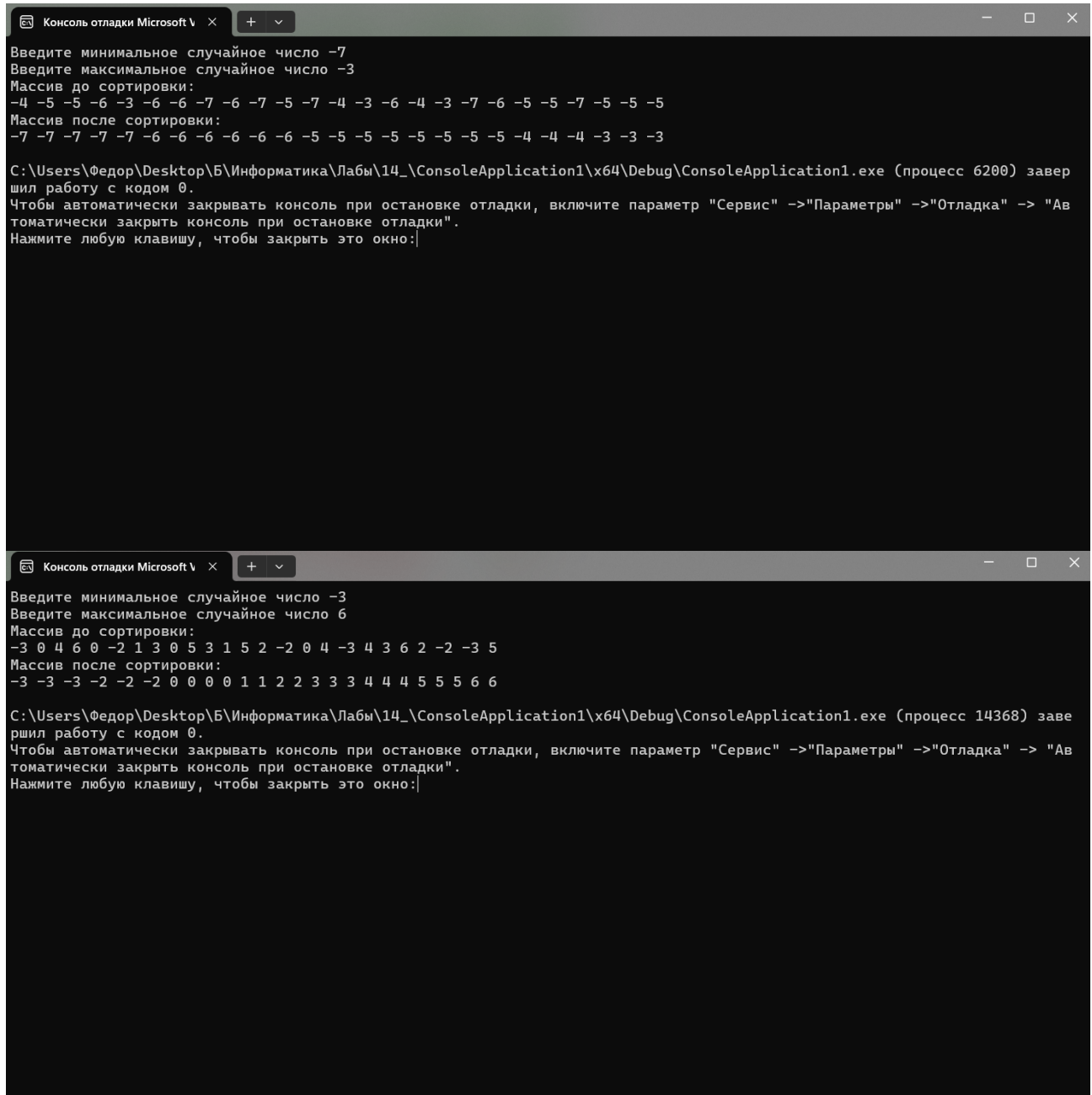
```

    cin >> max;
    ranArr(arr, N, min, max);
    cout << "Массив до сортировки:" << endl;
    printArr(arr, N);
    quickSortH(arr, 0, N-1);
    cout << "Массив после сортировки:" << endl;
    printArr(arr, N);
    return 0;
}

```

4.2 Скриншоты

Запустим программу, чтобы убедиться, что она работает верно.



```
Консоль отладки Microsoft V x + v
Введите минимальное случайное число 4
Введите максимальное случайное число 9
Массив до сортировки:
8 8 9 9 6 7 8 5 4 9 8 5 7 4 4 5 8 9 6 5 8 4 4 6 4
Массив после сортировки:
4 4 4 4 4 4 5 5 5 5 6 6 6 7 7 8 8 8 8 8 9 9 9 9
C:\Users\Федор\Desktop\Б\Информатика\Лабы\14_\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (процесс 4996) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

ВЫВОД

В итоге этой работы была составлена программа с сортировкой Хоара. Сделан вывод о полезности этой сортировки. Выявлены плюсы и минусы этой сортировки перед простыми.

Проведенная лабораторная работа была опубликована в общий доступ по адресу: https://github.com/Fedor0000/TheUltimateFolder/tree/main/Sem_2/Labs/14