Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы ИСПк- 204-52-00

Никулин Арсений Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

г. Киров

2023

1. **Цель работы**: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.
2. **Задание:**
3. Реализовать сортировку данных с помощью вставок.
4. Реализовать сортировку данных с помощью быстрого алгоритма.
5. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
6. Считывание и вывод данных необходимо производить из текстового файла.
7. Для демонстрации работы подпрограммы реализаций самостоятельно подготовить варианты входных данных (при этом объем тестовых файлов должен позволять оценить скорость работы программ).
8. **Описание алгоритма:**

**1ая программа:**

Программа **‘DKR\_5.1’** выполняет сортировку массива целых чисел методом сортировки вставками. В программе определены две функции сравнения (**Greater** и **Less**), а также процедуры **Swap** для обмена значениями и **InsertionSort** для сортировки методом вставки. Затем программа считывает данные из файла, выбирает тип сортировки (по возрастанию или убыванию), применяет соответствующую функцию сравнения и выполняет сортировку. Результат записывается в файл output.txt.

**2ая программа:**

Программа **‘DKR\_5.2’** реализует сортировку массива целых чисел методом быстрой сортировки (**Merge**). Как и в первом коде, определены функции сравнения (**Greater** и **Less**), процедура **Swap** для обмена значениями и **Merge** для рекурсивной сортировки. После считывания данных из файла и выбора типа сортировки, программа применяет соответствующую функцию сравнения и вызывает процедуру **Merge**. Результат также записывается в файл output.txt.

1. **Схемы алгоритмов:**

**Для DKR\_5.1**

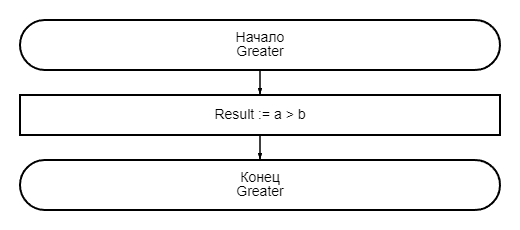


Рисунок 1 – Функция Greater

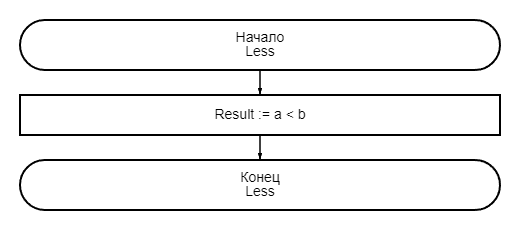


Рисунок 2 – Функция Less

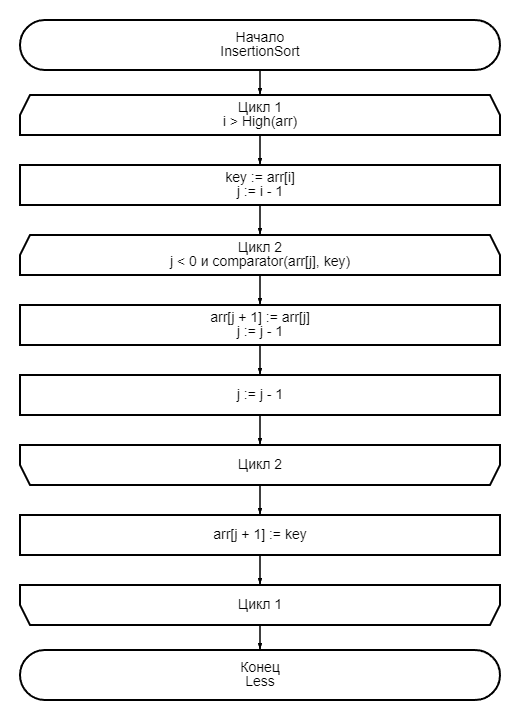


Рисунок 3 – Процедура InsertionSort

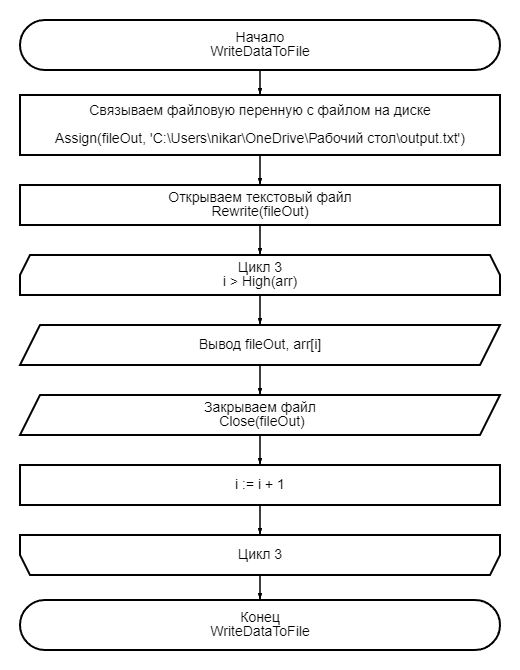


Рисунок 4 – Процедура WriteDataToFile

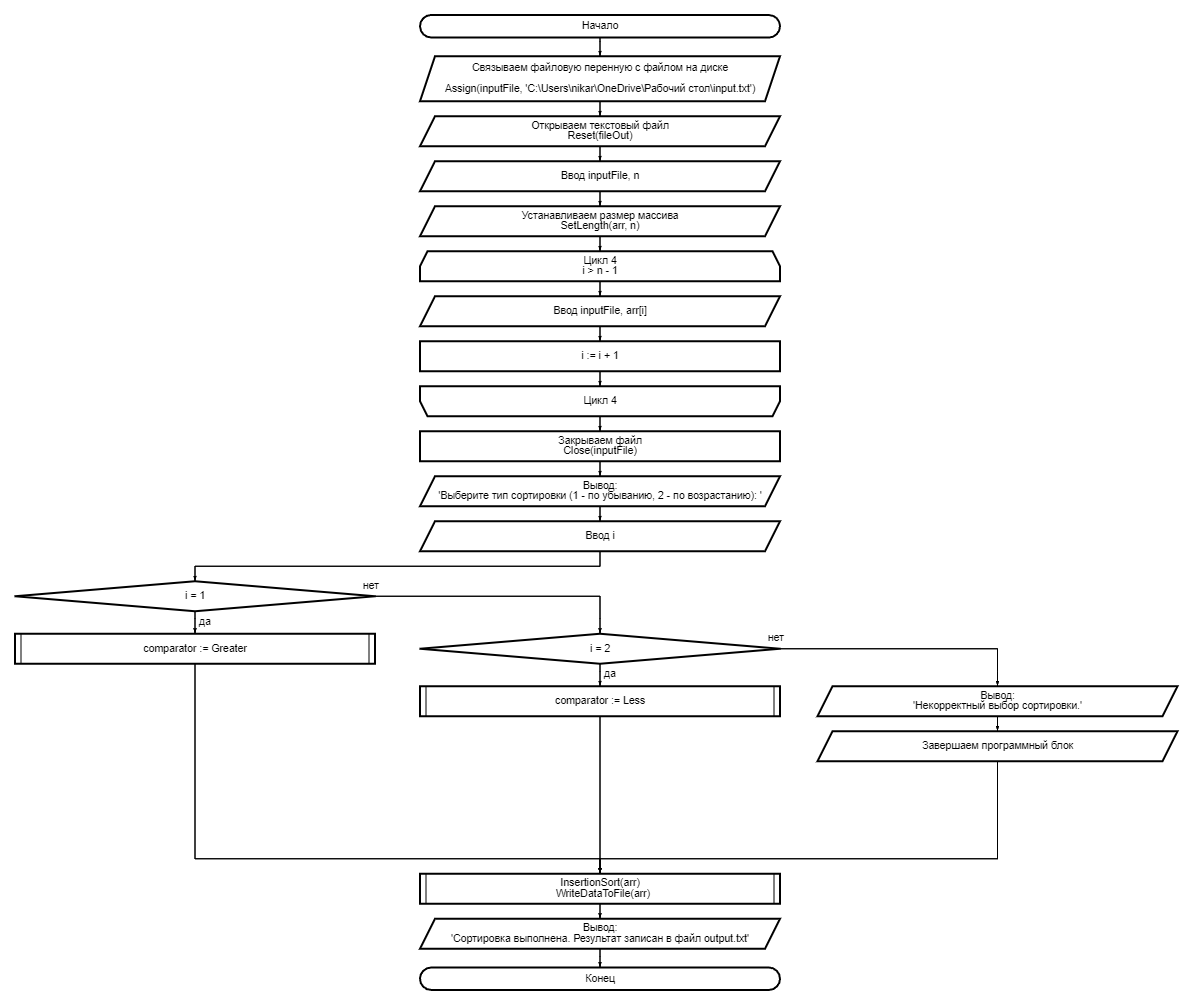


Рисунок 5 – Тело программы

**Для DKR\_5.2**

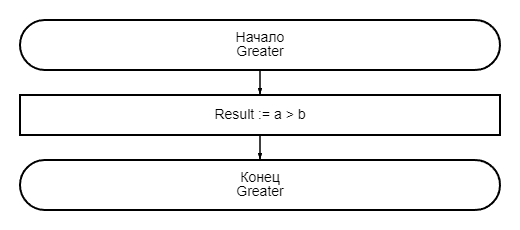


Рисунок 6 – Функция Greater

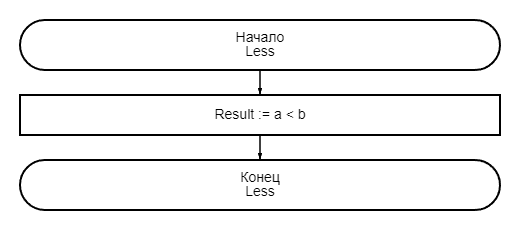


Рисунок 7 – Функция Less

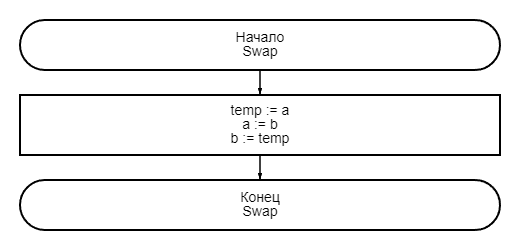


Рисунок 8 – Процедура Swap

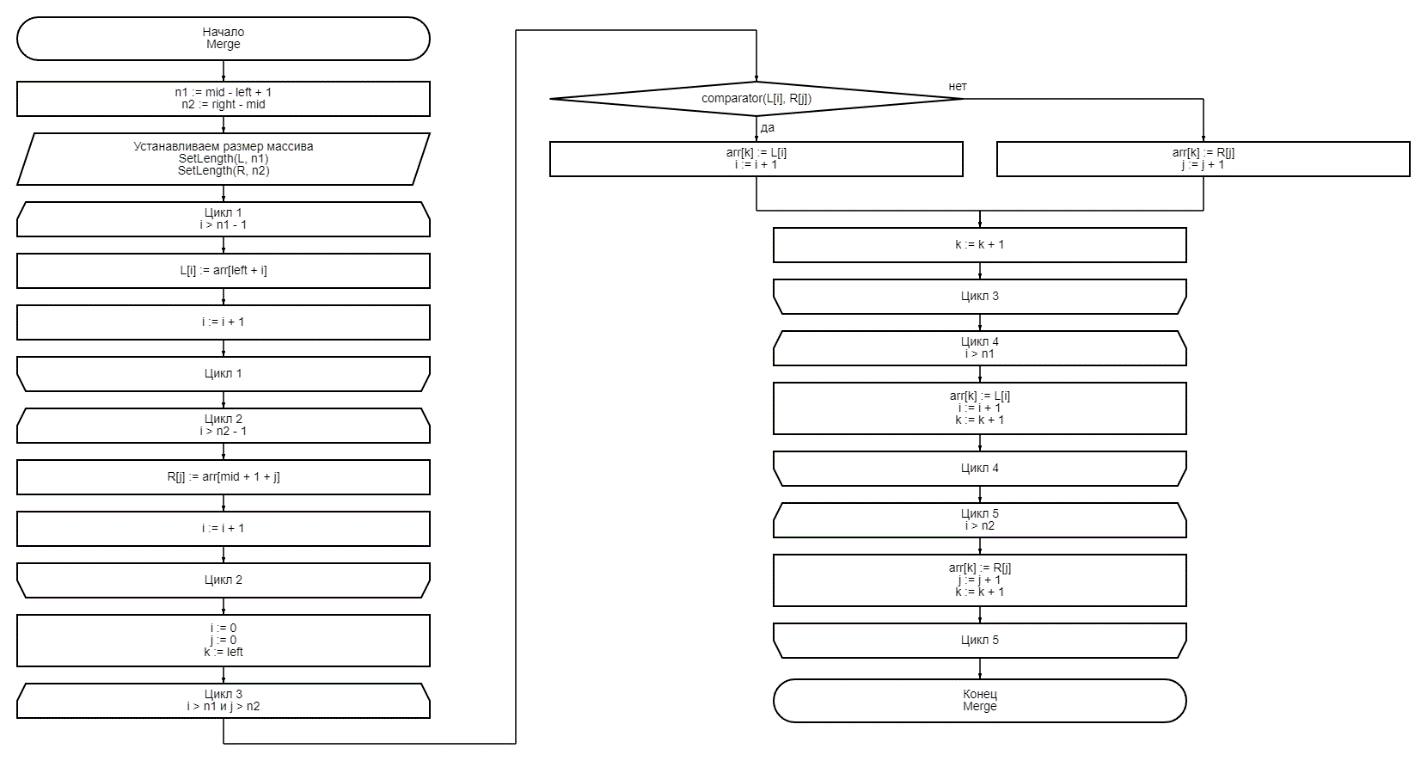


Рисунок 9 – Процедура Merge

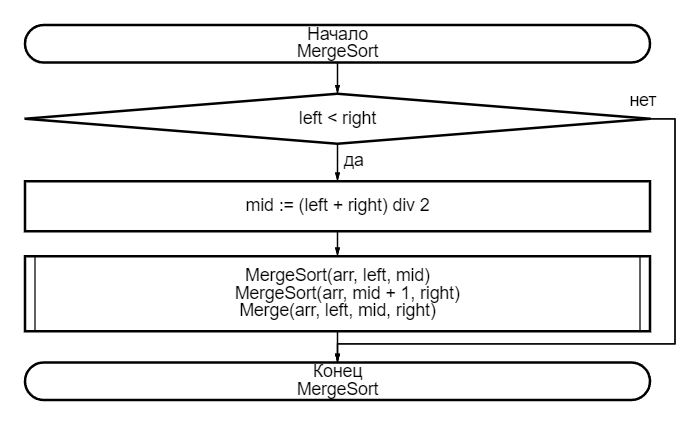


Рисунок 10 – Процедура MergeSort

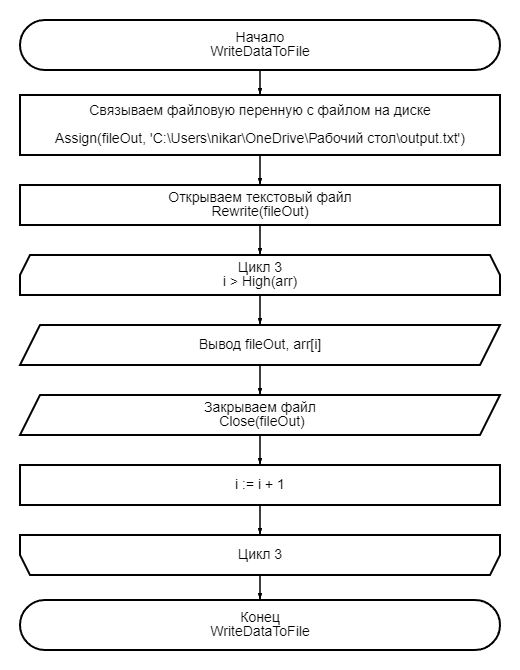


Рисунок 11 – Процедура WriteDataToFile

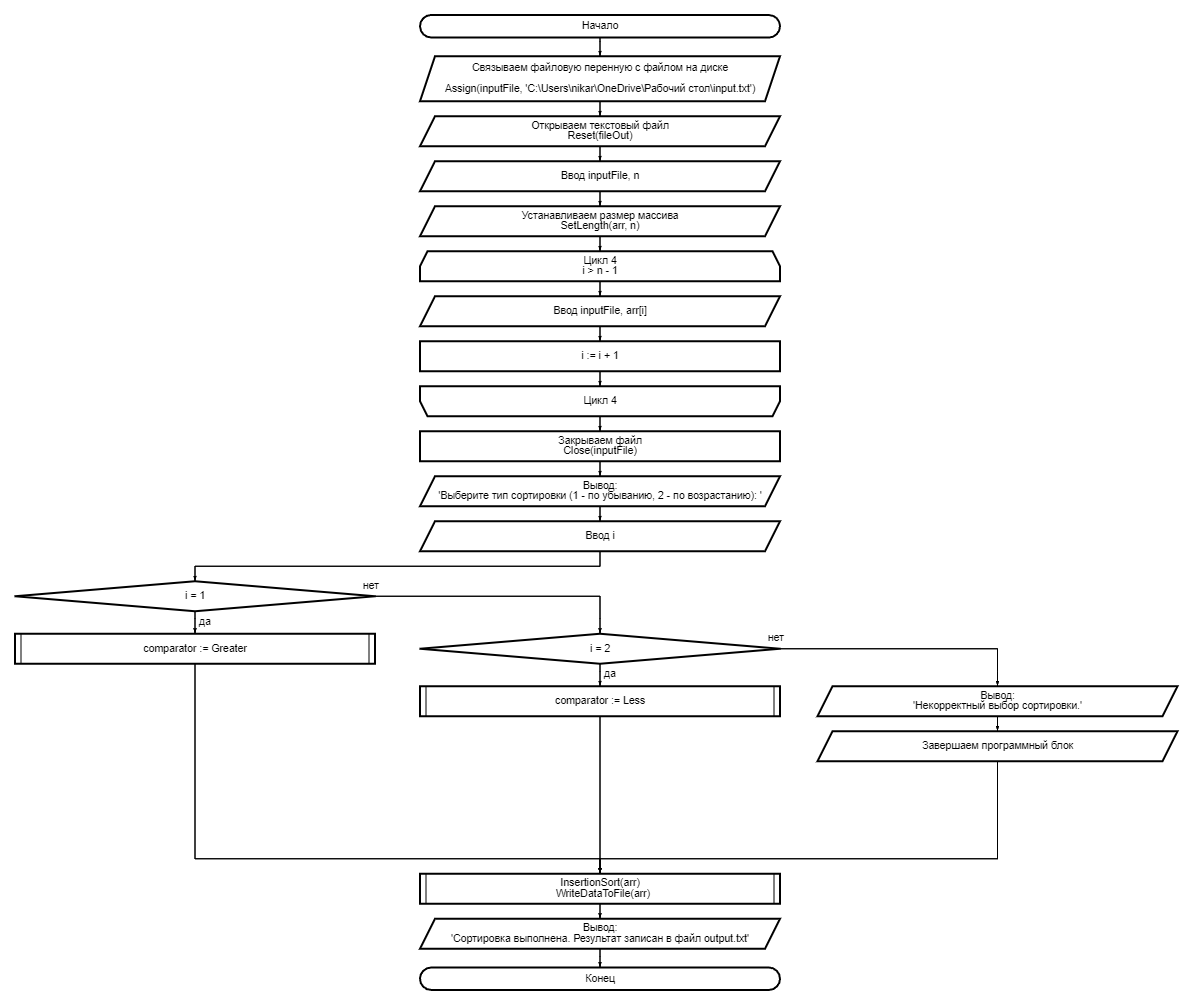


Рисунок 12 – Тело программы

**Код программ:**

**1) program** dkr5kod1;

**type**

TComparator = **function**(a, b: Integer): Boolean;

**var**

comparator: TComparator;

**function** Greater(a, b: Integer): Boolean;

**begin**

Result := a > b;

**end**;

**function** Less(a, b: Integer): Boolean;

**begin**

Result := a < b;

**end**;

**procedure** Swap(**var** a, b: Integer);

**var**

temp: Integer;

**begin**

temp := a;

a := b;

b := temp;

**end**;

**procedure** InsertionSort(**var** arr: **array of** Integer);

**var**

i, j, key: Integer;

**begin**

**for** i := 1 **to** High(arr) **do**

**begin**

key := arr[i];

j := i - 1;

**while** (j >= 0) **and** comparator(arr[j], key) **do**

**begin**

arr[j + 1] := arr[j];

j := j - 1;

**end**;

arr[j + 1] := key;

**end**;

**end**;

**procedure** WriteDataToFile(arr: **array of** Integer);

**var**

fileOut: Text;

i: Integer;

**begin**

Assign(fileOut, 'C:\Users\nikar\OneDrive\Рабочий стол\output.txt');

Rewrite(fileOut);

**for** i := 0 **to** High(arr) **do**

Writeln(fileOut, arr[i]);

Close(fileOut);

**end**;

**var**

arr: **array of** Integer;

n, i: Integer;

inputFile: Text;

**begin**

Assign(inputFile, 'C:\Users\nikar\OneDrive\Рабочий стол\input.txt');

Reset(inputFile);

ReadLn(inputFile, n);

SetLength(arr, n);

**for** i := 0 **to** n - 1 **do**

**begin**

ReadLn(inputFile, arr[i]);

**end**;

Close(inputFile);

Write('Выберите тип сортировки (1 - по убыванию, 2 - по возрастанию): ');

ReadLn(i);

**if** i = 1 **then**

comparator := Greater

**else if** i = 2 **then**

comparator := Less

**else**

**begin**

Writeln('Некорректный выбор сортировки.');

**Exit**;

**end**;

InsertionSort(arr);

WriteDataToFile(arr);

Writeln('Сортировка выполнена. Результат записан в файл output.txt');

**end**.

**2)** Merge(arr, left, mid, right);

**end**;

**end**;

**procedure** WriteDataToFile(arr: **array of** Integer);

**var**

fileOut: Text;

i: Integer;

**begin**

Assign(fileOut, 'C:\Users\nikar\OneDrive\Рабочий стол\output.txt');

Rewrite(fileOut);

**for** i := 0 **to** High(arr) **do**

Writeln(fileOut, arr[i]);

Close(fileOut);

**end**;

**var**

arr: **array of** Integer;

n, i: Integer;

inputFile: Text;

**begin**

Assign(inputFile, 'C:\Users\nikar\OneDrive\Рабочий стол\input.txt');

Reset(inputFile);

ReadLn(inputFile, n);

SetLength(arr, n);

**for** i := 0 **to** n - 1 **do**

**begin**

ReadLn(inputFile, arr[i]);

**end**;

Close(inputFile);

Write('Выберите тип сортировки (1 - по убыванию, 2 - по возрастанию): ');

ReadLn(i);

**if** i = 1 **then**

comparator := Greater

**else if** i = 2 **then**

comparator := Less

**else**

**begin**

Writeln('Некорректный выбор сортировки. Программа завершена.');

**Exit**;

**end**;

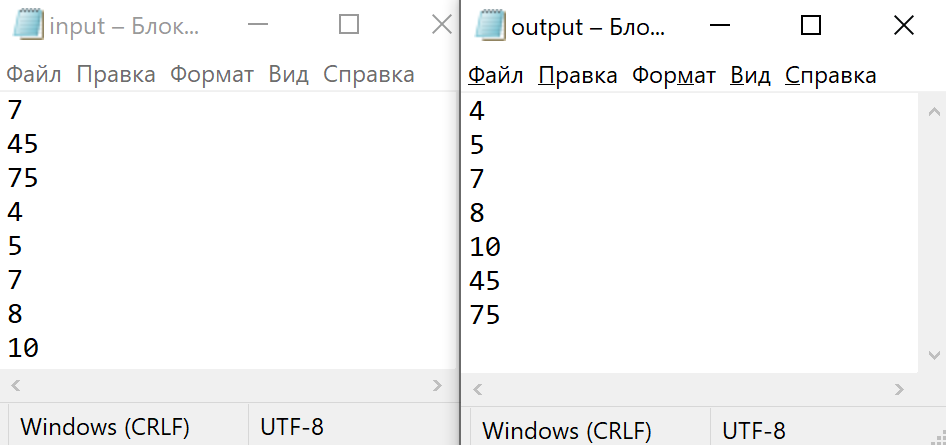
MergeSort(arr, 0, High(arr));

WriteDataToFile(arr);

Writeln('Сортировка выполнена. Результат записан в файл output.txt');

**end**.

1. **Результат выполнения программ:**

****

1. **Вывод:**

В результате выполнения лабораторной работы были получены базовые знания о наиболее известных алгоритмах сортировки, а также изучены принципы работы с текстовыми файлами. При реализации алгоритмов сортировки с помощью вставок и алгоритма слияния были учтены требования задания, в том числе возможность изменения компаратора путем передачи его в подпрограмму. В ходе работы было отработано считывание и вывод данных из текстового файла, а также подготовлены варианты входных данных для демонстрации работы подпрограммы реализаций.