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-203-52-00

Корюков Павел Дмитриевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:**

Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

1. **Формулировка задания:**

Вариант: 14

1. Реализовать сортировку данных с помощью вставок.
2. Реализовать сортировку данных с помощью пирамидного алгоритма.
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
4. Считывание и вывод данных необходимо производить из текстового файла.
5. Для демонстрации работы программных реализаций самостоятельно подготовить варианты входных данных (при этом объем текстовых файлов должен позволять оценить скорость работы программ).
6. **Описание алгоритма:**

**Задание 1.**

Процедура insertionSort принимает на вход ссылку на массив arr и функцию compare, которая определяет порядок сортировки.

Основная идея алгоритма заключается в том, что массив делится на две части: отсортированную и неотсортированную. Изначально отсортированная часть состоит из первого элемента (arr[0]), а неотсортированная - из всех остальных элементов.

Алгоритм проходит по неотсортированной части массива, выбирает текущий элемент (key) и сравнивает его со всеми предыдущими элементами отсортированной части массива. Если найден элемент, который больше (или меньше, в зависимости от выбранной функции сравнения) текущего, то он сдвигается вправо, а текущий элемент вставляется на его место. Процесс повторяется до тех пор, пока не будет пройдена вся неотсортированная часть массива.

После завершения сортировки, с помощью функции compareAscending массив data сортируется по возрастанию. Далее, отсортированный массив выводится в текстовый файл output1.txt.

Последняя строка кода выводит сообщение о завершении сортировки и записи результата в файл.

**Задание 2.**

В процедуре Radix Sort происходит сортировка значение по методу поразрядного алгоритма. Считывается минимальный индекс массива и с помощью цикла перебирается до максимального индекса. Входные данные берутся из текстового файл, соответственно результат выводится так же в другой текстовый файл. Далее следует вычислению с помощью метода.

Во второй части кода происходит записывание результата в текстовый файл. Открывается текстовый файл, с помощью цикла данные из файла перебираются и считываются программой. После для сохранности первый файл закрывается и отрывается второй файл куда заносится результат. После для сохранения данных файл так же закрывается и результат сохраняется. Программа выводит сообщение о успешной работе программы.

1. **Схемы алгоритмов:**

**Задание 1.**

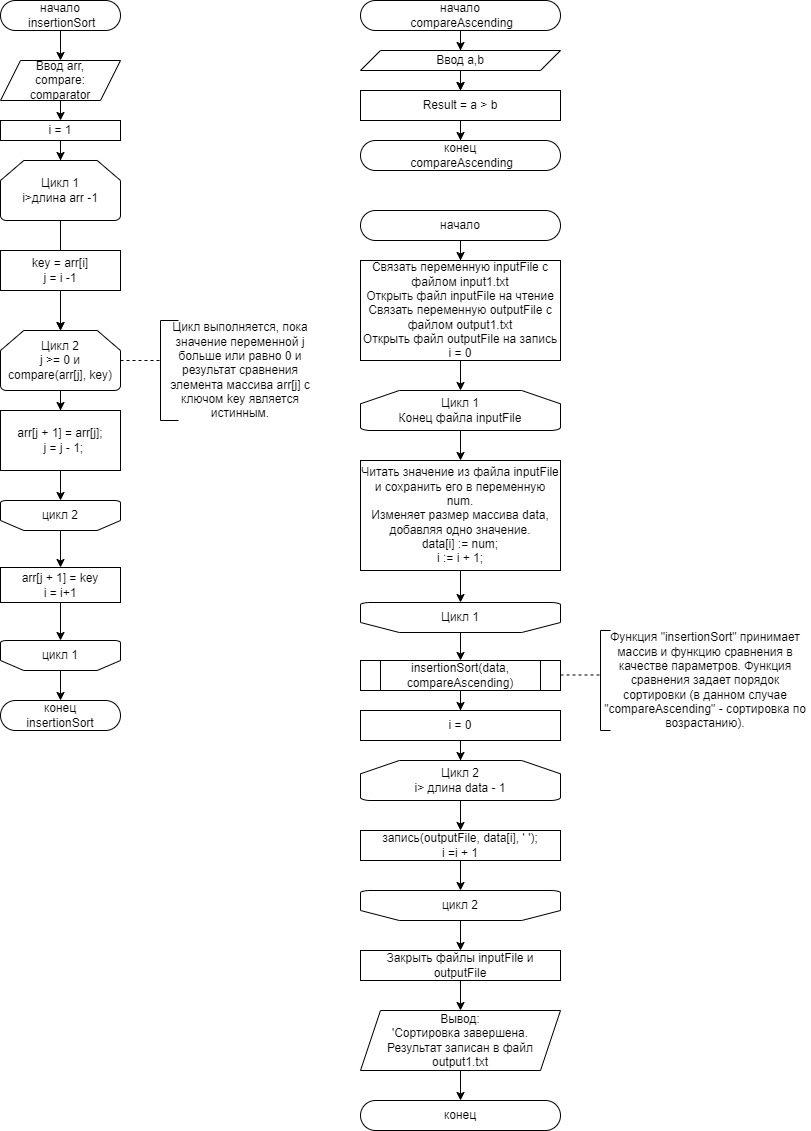
****

Рисунок 1 – Схема Алгоритма.

**Задание 2.**

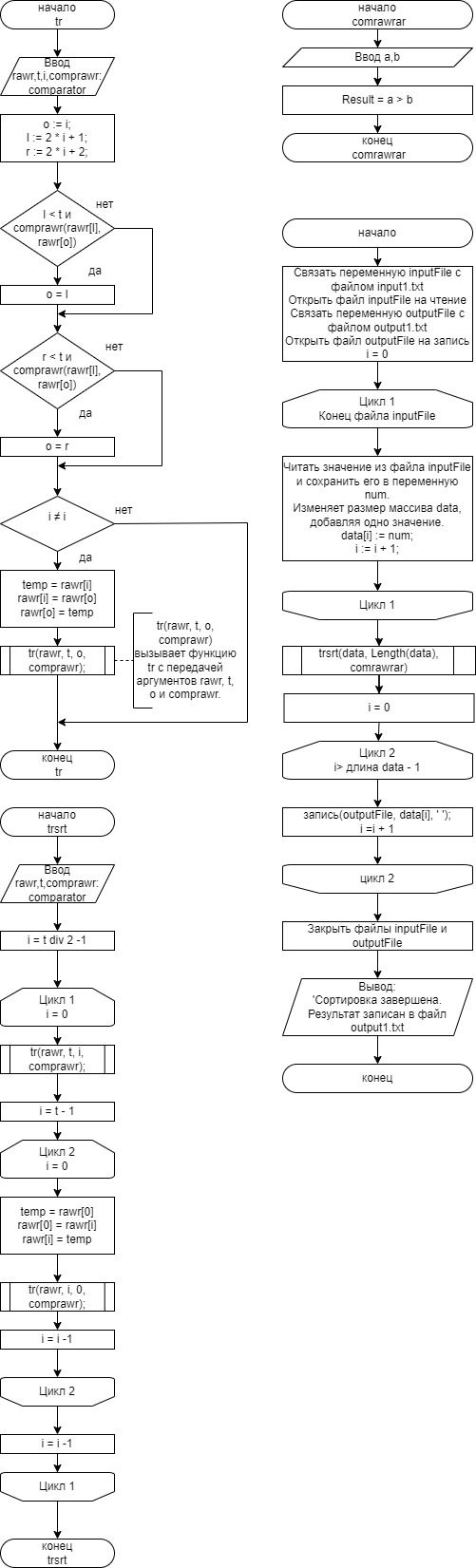
****

Рисунок 2 – Схема Алгоритма.

1. **Код программы:**

**Задание 1.**

**type**

comparator = **function**(a, b: integer): boolean;

**procedure** insertionSort(**var** arr: **array of** integer; compare: comparator);

**var**

i, j, key: integer;

**begin**

**for** i := 1 **to** Length(arr) - 1 **do**

**begin**

key := arr[i];

j := i - 1;

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

**begin**

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

j := j - 1;

**end**;

arr[j + 1] := key;

**end**;

**end**;

**function** compareAscending(a, b: integer): boolean;

**begin**

Result := a > b;

**end**;

**var**

inputFile, outputFile: text;

data: **array of** integer;

num, i: integer;

**begin**

Assign(inputFile, 'input1.txt');

Reset(inputFile);

Assign(outputFile, 'output1.txt');

Rewrite(outputFile);

i := 0;

**while not** Eof(inputFile) **do**

**begin**

read(inputFile, num);

SetLength(data, i + 1);

data[i] := num;

i := i + 1;

**end**;

insertionSort(data, compareAscending);

**for** i := 0 **to** Length(data) - 1 **do**

Write(outputFile, data[i], ' ');

Close(inputFile);

Close(outputFile);

writeln('Сортировка завершена. Результат записан в файл output1.txt');

**end**.

**Задание 2.**

**type**

comparator = **function**(a, b: integer): boolean;

**procedure** tr (**var** rawr: **array of** integer; t, i: integer; comprawr: comparator);

**var**

o, l, r, temp: integer;

**begin**

o := i;

l := 2 \* i + 1;

r := 2 \* i + 2;

**if** (l < t) **and** comprawr(rawr[l], rawr[o]) **then**

o := l;

**if** (r < t) **and** comprawr(rawr[r], rawr[o]) **then**

o := r;

**if** o <> i **then**

**begin**

temp := rawr[i];

rawr[i] := rawr[o];

rawr[o] := temp;

tr(rawr, t, o, comprawr);

**end**;

**end**;

**procedure** trsrt(**var** rawr: **array of** integer; t: integer; comprawr: comparator);

**var**

i, temp: integer;

**begin**

**for** i := t **div** 2 - 1 **downto** 0 **do**

tr(rawr, t, i, comprawr);

**for** i := t - 1 **downto** 0 **do**

**begin**

temp := rawr[0];

rawr[0] := rawr[i];

rawr[i] := temp;

tr(rawr, i, 0, comprawr);

**end**;

**end**;

**function** comrawrar(a, b: integer): boolean;

**begin**

Result := a > b; // для сортировки по возрастанию

**end**;

**var**

inputFile, outputFile: text;

data: **array of** integer;

num, i: integer;

**begin**

Assign(inputFile, 'input1.txt');

Reset(inputFile);

Assign(outputFile, 'output1.txt');

Rewrite(outputFile);

i := 0;

**while not** Eof(inputFile) **do**

**begin**

read(inputFile, num);

SetLength(data, i + 1);

data[i] := num;

i := i + 1;

**end**;

trsrt(data, Length(data), comrawrar);

**for** i := 0 **to** Length(data) - 1 **do**

Write(outputFile, data[i], ' ');

Close(inputFile);

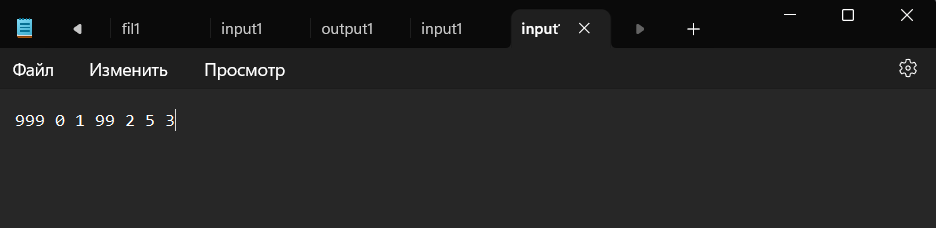
Close(outputFile);

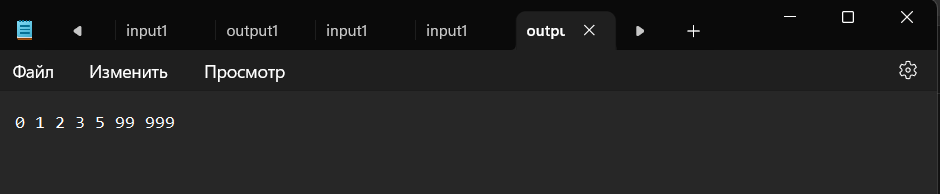
writeln('Сортировка завершена. Результат записан в файл output1.txt');

**end**.

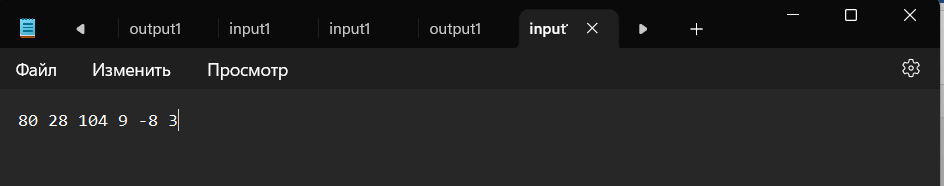
1. **Результат выполнения программы:**

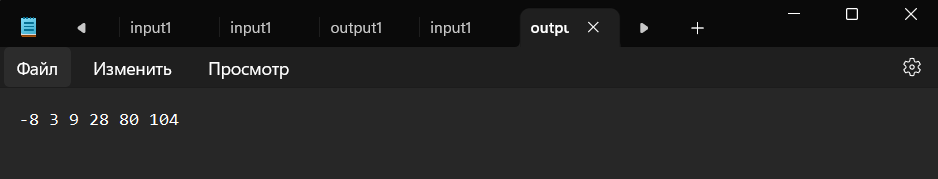
**Задание 1.**

****

****

**Задание 2.**

****

****

1. **Вывод.**

Данная работа прошла успешно. В работе были рассмотрены операции с сортировками чисел на языке программирования Pascal. А также работа с текстовыми файлами, а также взятие из них входных данных и вывода готовых данных в файл.

Трудности в работе возникли в построении схемы алгоритма и соединении связей в схеме алгоритма. Также были трудности с файлами и небольшие проблемы с пирамидным алгоритмом, но они вскоре решились.

Таким образом, в данной работе были операции исследованием алгоритмов сортировки, построение схемы алгоритма, был описан каждой алгоритм, и проверка данной программы на работоспособность и анализ результата по ранее написанному коду.