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-205-52-00

Веселов Алексей Денисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1 Цель работы

Цель работы: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

2 Формулировка задания (с вариантом)

Вариант: 4

Задание:

1. Реализовать сортировку данных с помощью алгоритма выбором.

2. Реализовать сортировку данных с помощью быстрого 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 тестовых файлов должен позволять оценить скорость работы программ).

3 Описание алгоритма

В данной задаче требуется знать и понимать принципы сортировки.

В первом случае дана сортировка выбором, где максимальный элемент меняется с первым или последним элементом, в зависимости от компаратора.

Во втором случае используется быстрая сортировка, где выбираться опорное число по позиции (как правило центральное), и все элементы что больше переноситься вправо, те что меньше влево. Затем повторить для левой и правой части повторить эти действия пока левая и правая части не станут едины.

Компараторы зависят от знака.

4. Схема алгоритма с комментарием

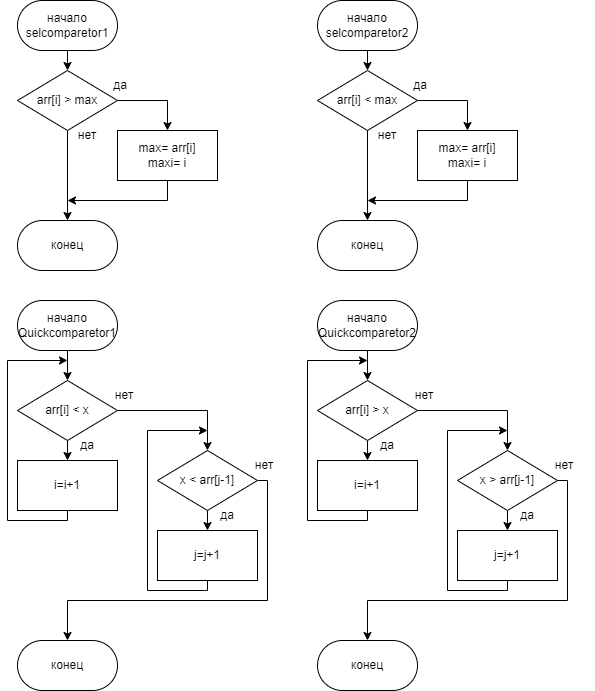


Рисунок 1 - компараторы

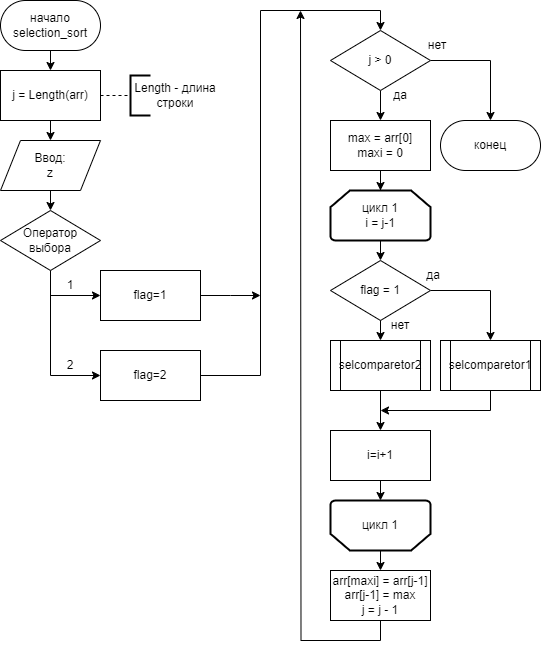


Рисунок 2 - сортировка выбором

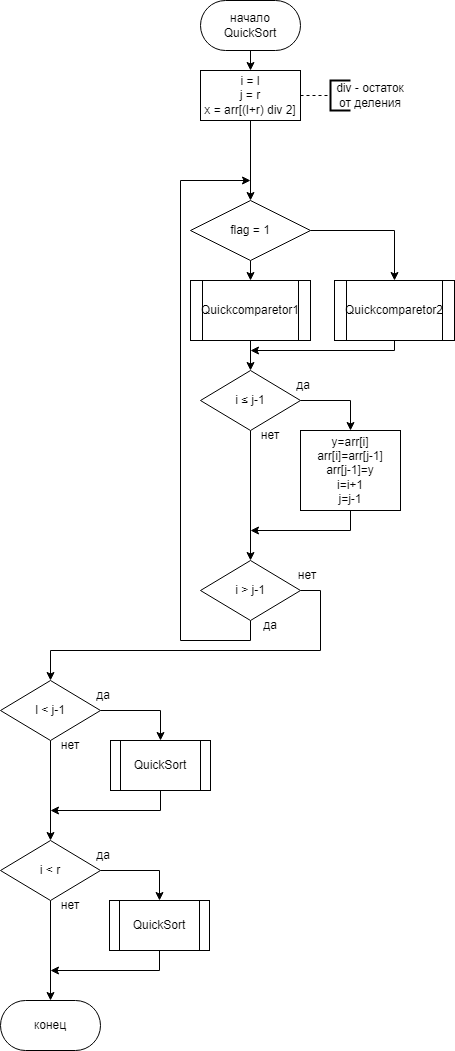


Рисунок 3 -быстрая сортировка

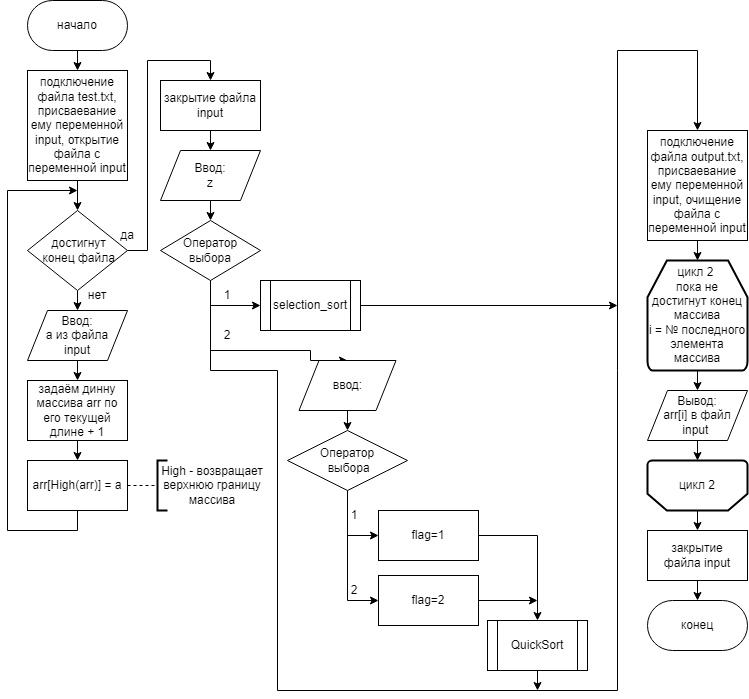


Рисунок 4 - подключение файлов и тело программы

5 Код программы

**program** z501;

**var** arr: **array of** Integer;

z,a, i, j, n,max, maxi : integer;

input,output: TextFile;

**procedure** selcomparetor1(**var** max, maxi : integer; i : integer);

**begin**

**if** arr[i] > max **then** // Зависит от знака

**begin**

max := arr[i];

maxi := i

**end**;

**end**;

**procedure** selcomparetor2(**var** max, maxi : integer; i : integer);

**begin**

**if** arr[i] < max **then** // Зависит от знака

**begin**

max := arr[i];

maxi := i

**end**;

**end**;

**procedure** Quickcomparetor1(**var** i,j: integer; x:integer);

**begin**

**while** (arr[i] < x) **do** i+=1;

**while** (x < arr[j-1]) **do** j-=1;

**end**;

**procedure** Quickcomparetor2(**var** i,j: integer; x:integer);

**begin**

**while** (arr[i] > x) **do** i+=1;

**while** (x > arr[j-1]) **do** j-=1;

**end**;

**procedure** selection\_sort(**var** arr: **array of** Integer);

**var** i,j, max, maxi,flag : integer;

**begin**

j := Length(arr);

writeln('компоратор');

read(z);

**case** z **of**

1:flag:=1;

2:flag:=2;

**end**;

**while** j > 0 **do**

**begin**

max := arr[0];

maxi := 0;

**for** i := 1 **to** j-1 **do if** flag = 1 **then** selcomparetor1(max,maxi,i)

**else** selcomparetor2(max,maxi,i);

arr[maxi] := arr[j-1];

arr[j-1] := max;

j := j - 1;

//writeln(max:4,maxi:4);

// writeln(arr);

**end**;

**end**;

**procedure** QuickSort( L, R : Integer ); { Быстрая сортировка массива A[] }

**var** i,j,x,y,flag : integer;

**begin**

i := l;

j := r;

x := arr[(l+r) **div** 2];

//writeln(arr, x);

writeln('компоратор');

read(z);

**case** z **of**

1:flag:=1;

2:flag:=2;

**end**;

**repeat**

**if** flag = 1 **then** Quickcomparetor1(i,j,x)

**else** Quickcomparetor2(i,j,x);

**if** i <= j-1 **then**

**begin**

y:=arr[i];

arr[i]:=arr[j-1];

arr[j-1]:=y;

i+=1;

j-=1;

**end**;

**until** i > j-1;

//writeln(arr, x);

**if** l < j-1 **then** QuickSort(l,j);

**if** i < r **then** QuickSort(i,r);

**end**;

**begin**

AssignFile(input, 'test.txt');

Reset(input);

**while not** Eof(input) **do**

**begin**

ReadLn(input, a);

SetLength(arr, Length(arr) + 1);

arr[High(arr)] := a;

**end**;

CloseFile(input);

writeln('метод');

read(z);

**case** z **of**

1:selection\_sort(arr);

2:QuickSort(0,Length(arr));

**end**;

AssignFile(input,'output.txt');

Rewrite(input);

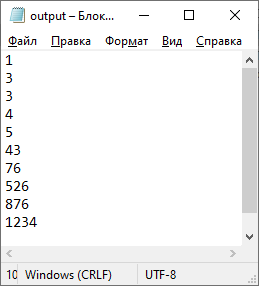
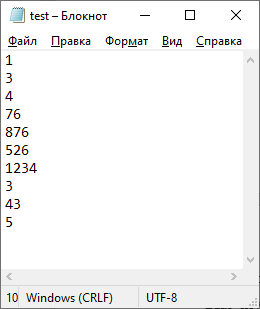
**for** i := Low(arr) **to** High(arr) **do**

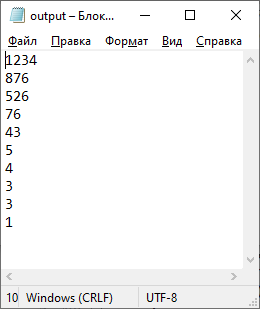
WriteLn(input, arr[i]);

CloseFile(input);

**end**.

6 Результат выполнения программы





7 Вывод

Данная работа не сложной.

Были изучены методы сортировки.

Возникли небольшие трудности с подключением подпрограмм, но данная проблема достаточно быстро была решена.

Так же были придуманы наборы тестов.

Были изучены сложности сортировки.

Далее был изучен бумажный источник информации до 60 стр., носящий название «математика и дурацкими рисунками». Достаточно полезный источник информации т.к. расширяет кругозор в области математики, программирования и способов мышления.

Само собой цель работы была достигнута.