МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Исследование алгоритмов сортировки

Отчет

Лабораторная работа №5 по дисциплине

«Программирование»

Выполнил студент группы ИВТ-11 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

Киров 2020

**Цель работы**

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

**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 вводных данных (при этом объём вводных данных должен позволять оценить скорость работы программы).

**Программа**

**1. Сортировка выбором возрастающая**

program kek;

uses crt;

type TBlack = function(a,b:longint):boolean;

var f,g:text;

arr:array [1..100005] of longint;

n,i,j,x,d,max,kk:longint;

function comparator(a,b:longint):boolean;

begin

comparator:=(a>b);

end;

procedure sortirovka(d,j:longint;gg:TBlack);

begin

d:= arr[1]+1;

j := d;

while j > 1 do

begin

max := arr[2];

kk := 2;

for i := 3 to j do

if comparator(arr[i],max) then

begin

max := arr[i];

kk := i

end;

arr[kk] := arr[j];

arr[j] := max;

j := j - 1

end;

end;

begin

clrscr;

assign(f,'C:/kek.txt');

reset(f);

assign(g,'C:/shrek.txt');

rewrite(g);

n:= 0;

while not eof(f) do

begin

n:= n+1;

read(f,arr[n]);

end;

sortirovka(d,j,@comparator);

for i:=2 to n do

write(g,arr[i],' ');

close(f);

close(g);

end.

**2. Сортировка выбором убывающая**

program kek;

uses crt;

type TBlack = function(a,b:longint):boolean;

var f,g:text;

arr:array [1..100005] of longint;

n,i,j,x,d,max,kk:longint;

function comparator(a,b:longint):boolean;

begin

comparator:=(a<b);

end;

procedure sortirovka(d,j:longint;gg:TBlack);

begin

d:= arr[1]+1;

j := d;

while j > 1 do

begin

max := arr[2];

kk := 2;

for i := 3 to j do

if comparator(arr[i],max) then

begin

max := arr[i];

kk := i

end;

arr[kk] := arr[j];

arr[j] := max;

j := j - 1

end;

end;

begin

clrscr;

assign(f,'C:/kek.txt');

reset(f);

assign(g,'C:/shrek.txt');

rewrite(g);

n:= 0;

while not eof(f) do

begin

n:= n+1;

read(f,arr[n]);

end;

sortirovka(d,j,@comparator);

for i:=2 to n do

write(g,arr[i],' ');

close(f);

close(g);

end.

**3.Быстрая сортировка возрастающая**

program kek;

uses crt;

type TBlack = function(a,b:longint):boolean;

var f,g:text;

arr:array [1..1000005] of longint;

n,i,j,x,d,max,kk:longint;

function comparator(a,b:longint):boolean;

begin

comparator:=(a<b);

end;

procedure sortirovka(n,k:longint; pp:TBlack);

var

i,j,x1,y1,m:longint;

begin

i:=n;

j:=k;

m:=round ((n+k)/2);

x1:=arr[m];

repeat

while pp(arr[i],x1) do inc(i);

while pp(x1,arr[j]) do dec(j);

if i<=j then

begin

y1:=arr[i];

arr[i]:=arr[j];

arr[j]:=y1;

inc(i);

dec(j);

end;

until i>j;

if n<j then sortirovka(n,j,pp);

if i<k then sortirovka(i,k,pp);

end;

begin

clrscr;

assign(f,'C:/gg.txt');

reset(f);

assign(g,'C:/shrek.txt');

rewrite(g);

n:= 0;

while not eof(f) do

begin

n:= n+1;

read(f,arr[n]);

end;

d:= arr[1]+1;

sortirovka(2,d, @comparator);

for i:=2 to n do

write(g,arr[i],' ');

close(f);

close(g);

end.

**4. Быстрая сортировка убывающая**

program kek;

uses crt;

type TWhite = function(a,b:longint):boolean;

var f,g:text;

arr:array [1..1000005] of longint;

n,i,j,x,d,max,kk:longint;

function comparator(a,b:longint):boolean;

begin

comparator:=(a>b);

end;

procedure sortirovka(n,k:longint; gg:TWhite);

var

i,j,x1,y1,m:longint;

begin

i:=n;

j:=k;

m:=round ((n+k)/2);

x1:=arr[m];

repeat

while gg(arr[i],x1) do inc(i);

while gg(x1,arr[j]) do dec(j);

if i<=j then

begin

y1:=arr[i];

arr[i]:=arr[j];

arr[j]:=y1;

inc(i);

dec(j);

end;

until i>j;

if n<j then sortirovka(n,j,gg);

if i<k then sortirovka(i,k,gg);

end;

begin

clrscr;

assign(f,'C:/kek.txt');

reset(f);

assign(g,'C:/shrek.txt');

rewrite(g);

n:= 0;

while not eof(f) do

begin

n:= n+1;

read(f,arr[n]);

end;

d:= arr[1]+1;

sortirovka(2,d,@comparator);

for i:=2 to n do

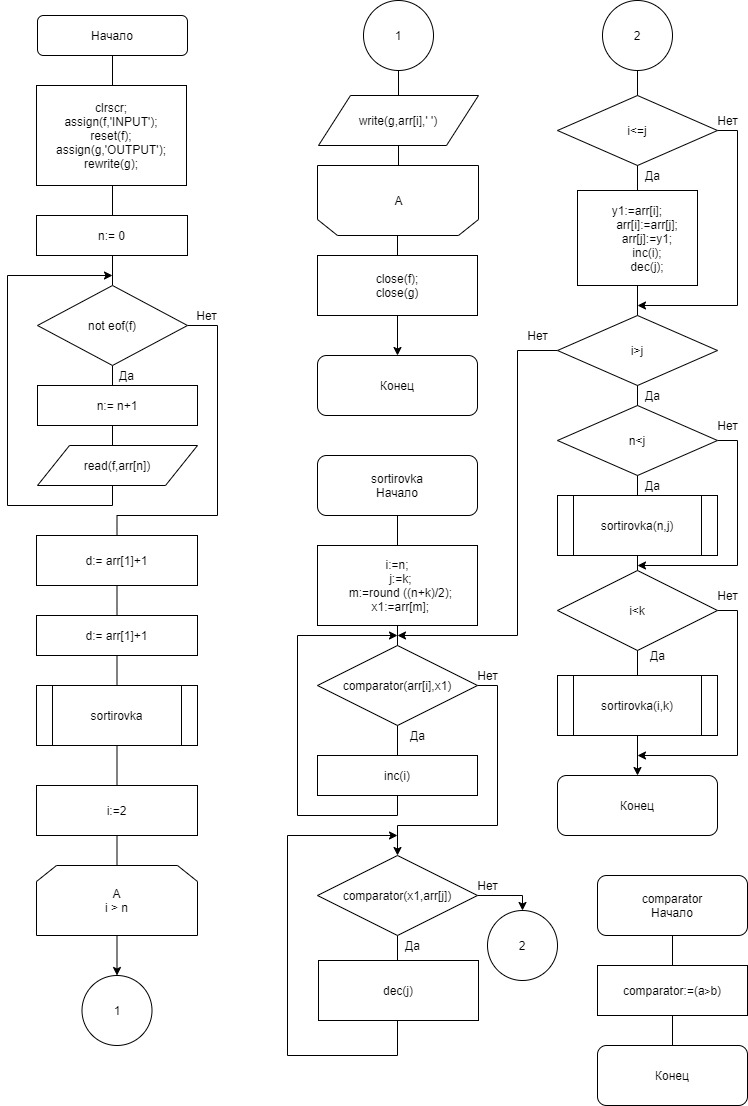
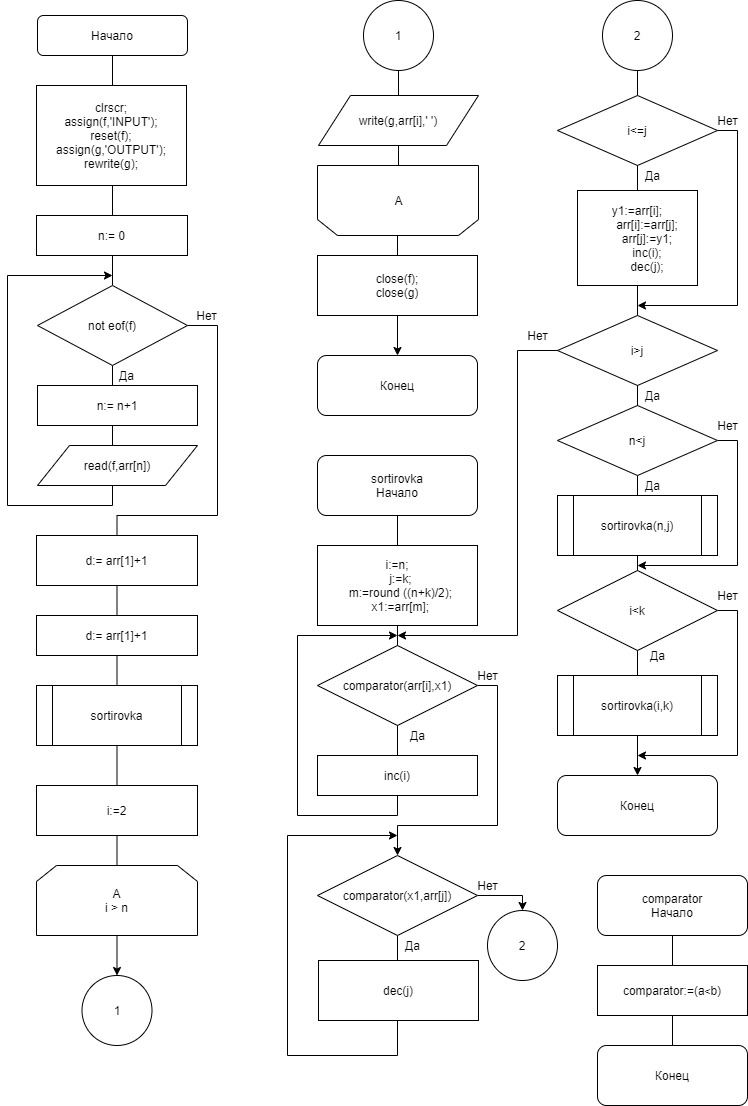
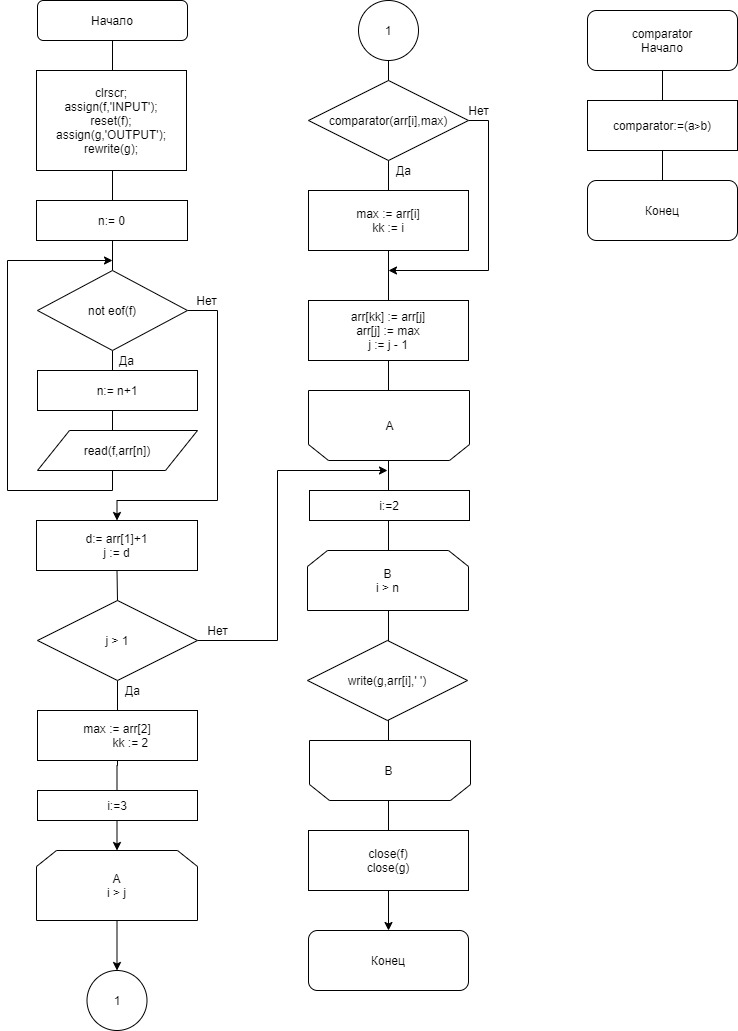
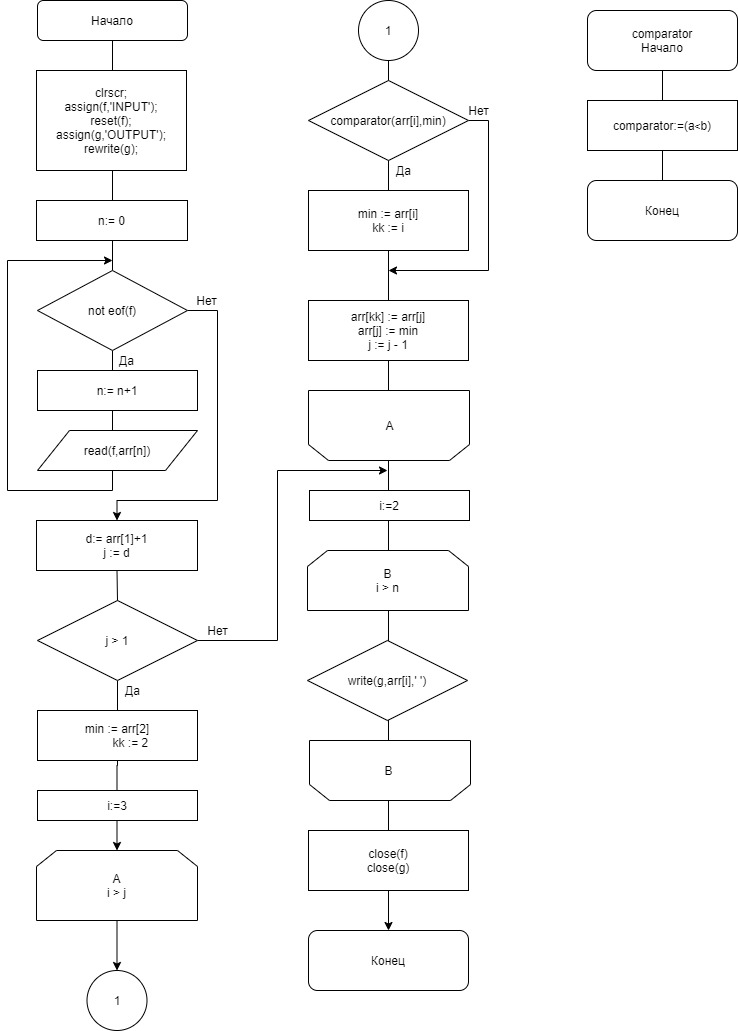
write(g,arr[i],' ');

close(f);

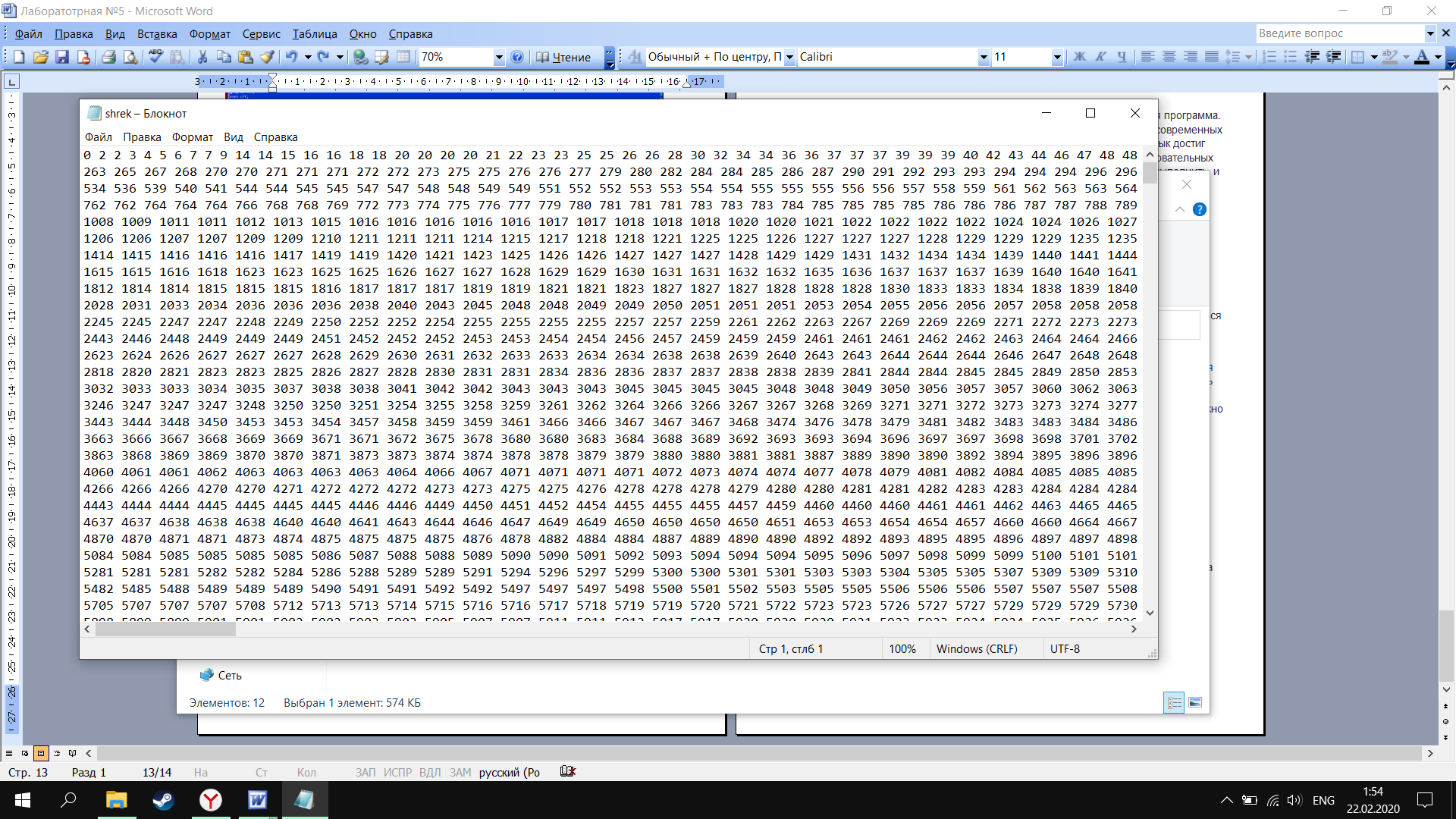
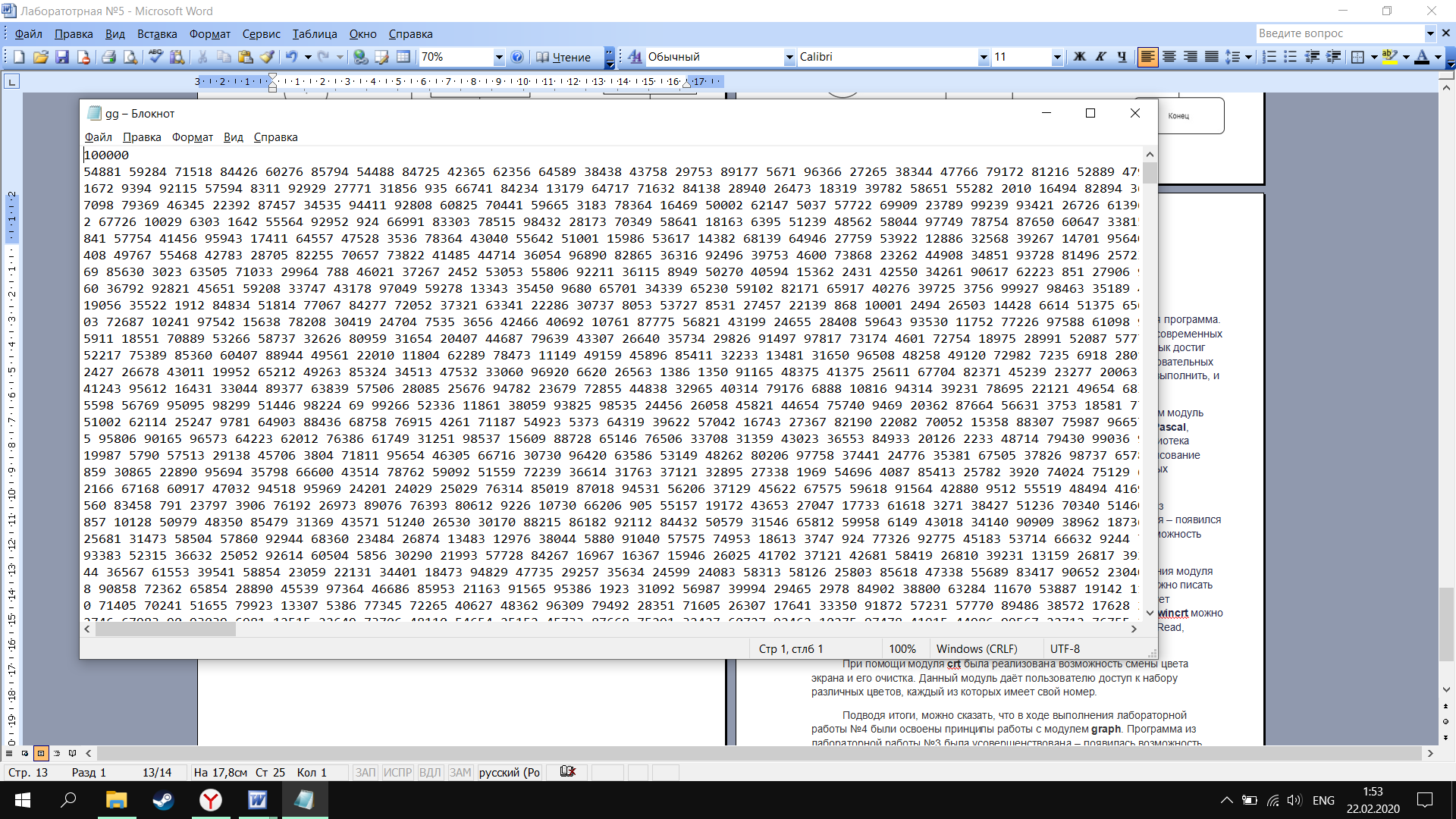
close(g);

end.

**Схема**



**Скриншоты программы**



**Вывод**

Язык программирования **Pascal** – это базовая высокоуровневая программа. Она используется как основа для программирования в большинстве современных школ. Такой востребованностью среди учебных заведений данный язык достиг благодаря своей простоте. Любая программа состоит из двух последовательных частей: описание последовательных действий, которое необходимо выполнить, и описание данных, которыми оперируют эти действия.

В ходе выполнения данной лабораторной работы были изучены методы сортировки целых чисел. Сортировку чисел возможно реализовать двумя путями: по возрастанию и по убыванию, что и требовалось сделать в данной лабораторной работе. Все методы сортировок чисел делятся на простые и сложные. Были реализованы такие сортировки чисел как **«быстрая сортировка»** и **«сортировка выбором»**. Первая сортировка является сложной, а вторая простой, что было сказано в задании.

Данная лабораторная работа способствовала обретению навыков работы с файлами в **Pascal.** Для работы с файлами были использованы специальные функции, предназначенные для этого. Одной из особенностей данной лабораторной работы являлось то, что исходные данные были записаны в файле. Конечные же данные нужно было записать уже в другой файл, предназначенный специально для этого

Для реализации сортировок были использованы подпрограммы. Благодаря **procedure** удалось реализовать сортировки, а **function** в написании компараторов для сортировок. Компаратор должен был напрямую передавать данные из функции в подпрограмму

Экспериментальным путём было продемонстрировано, что данные сортировки способны за короткое время отсортировать массив чисел от 10000 до 1000000. Данные числа доказывают, что быстрая сортировка не зря «носит» такое название.