МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

### Лабораторная работа № 5 по дисциплине

«Программирование»

«Исследование алгоритмов сортировки»

Выполнил студент группы ИВТб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А.

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

Киров 2021

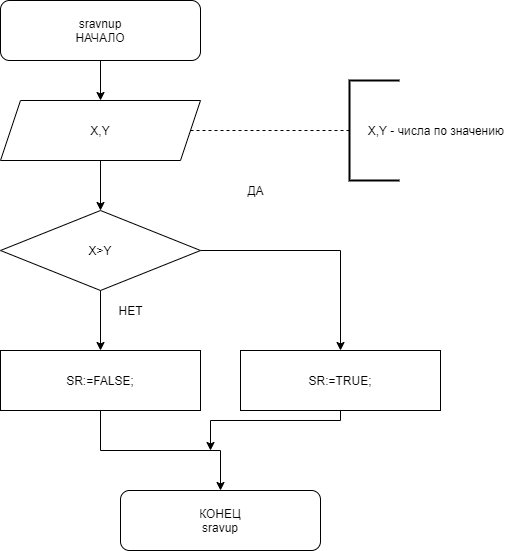
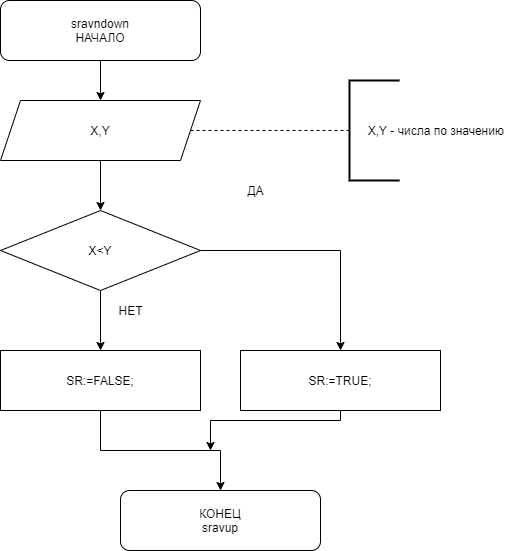
***Цель работы:*** получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

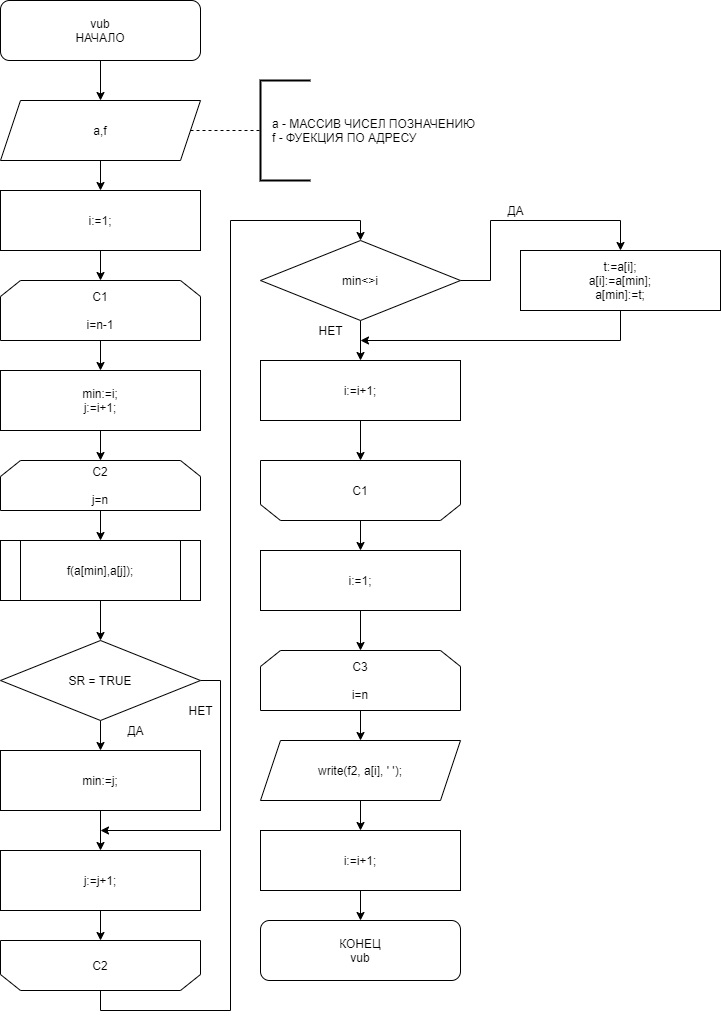
***Задания:***

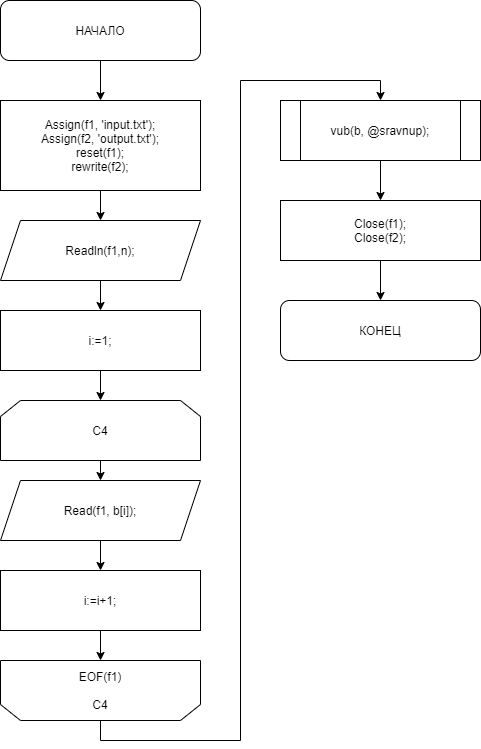
1. Реализовать сортировку данных с помощью алгоритма выбором.
2. Реализовать сортировку данных с помощью алгоритма слияния.
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
4. Считывание и вывод данных необходимо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

***Схема алгоритмов:***

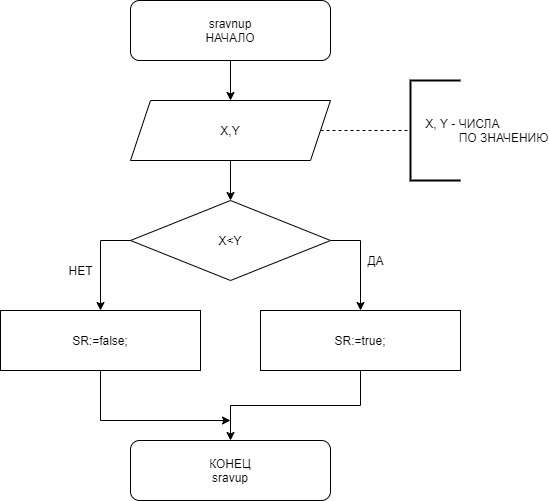
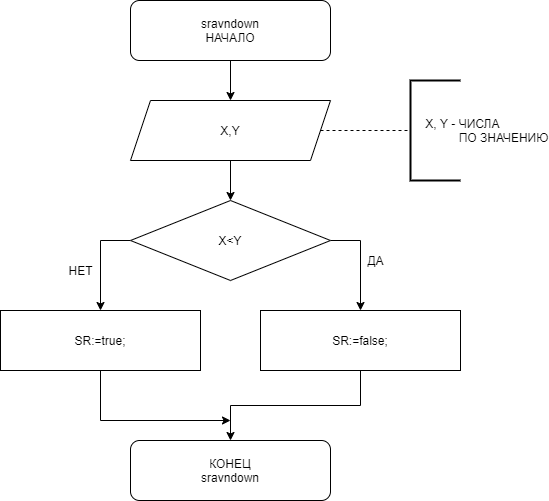
Сортировка выбором

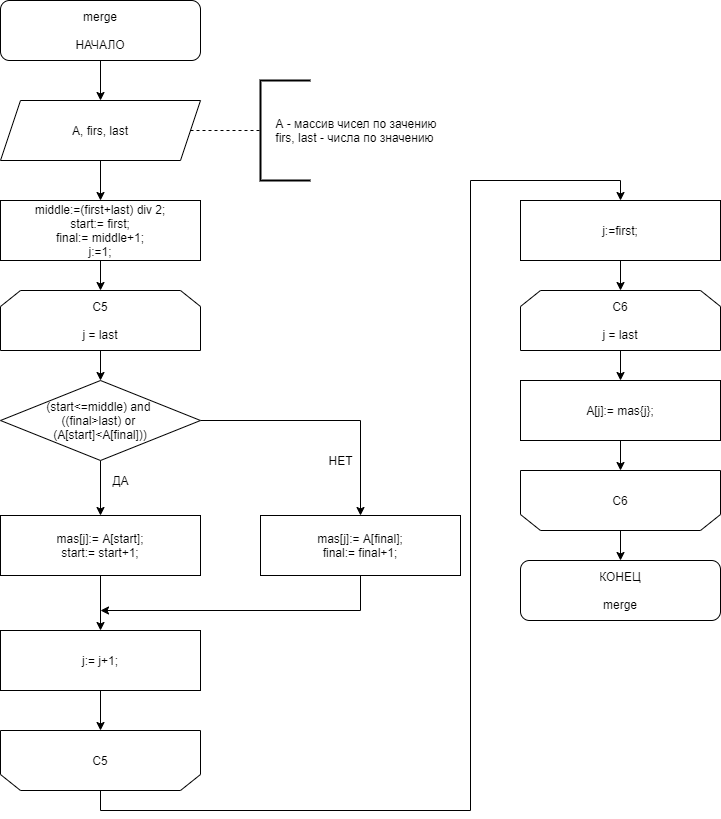
 

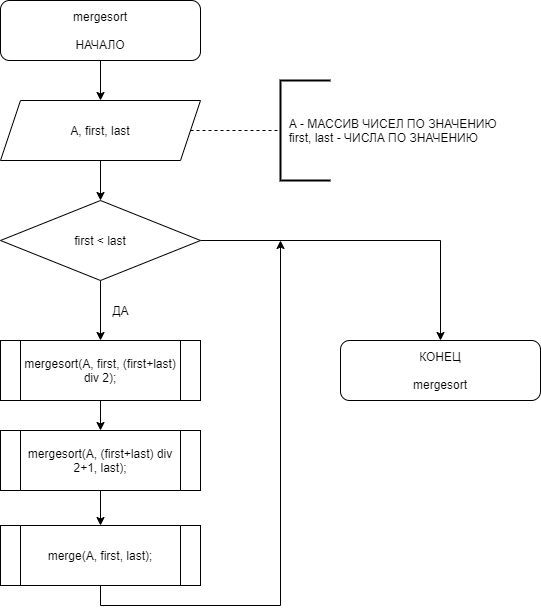


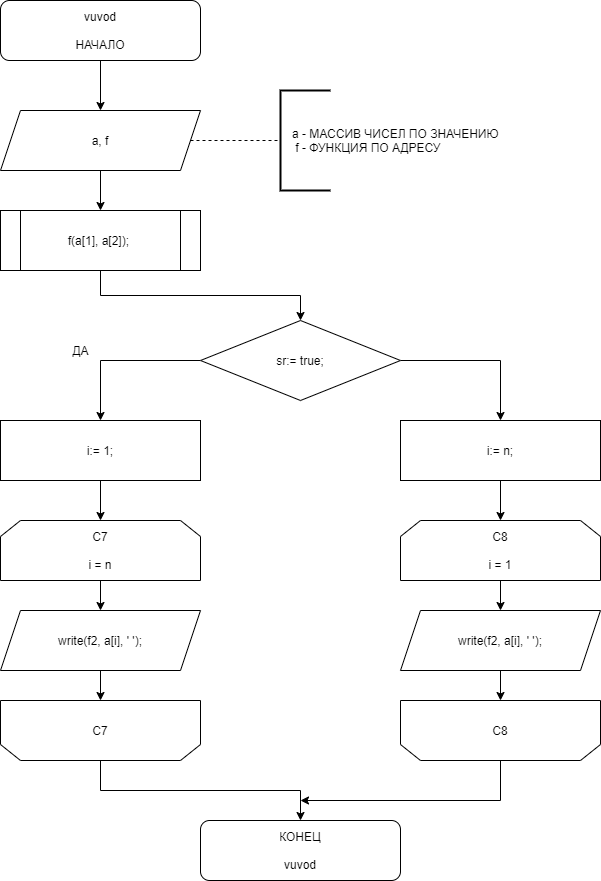


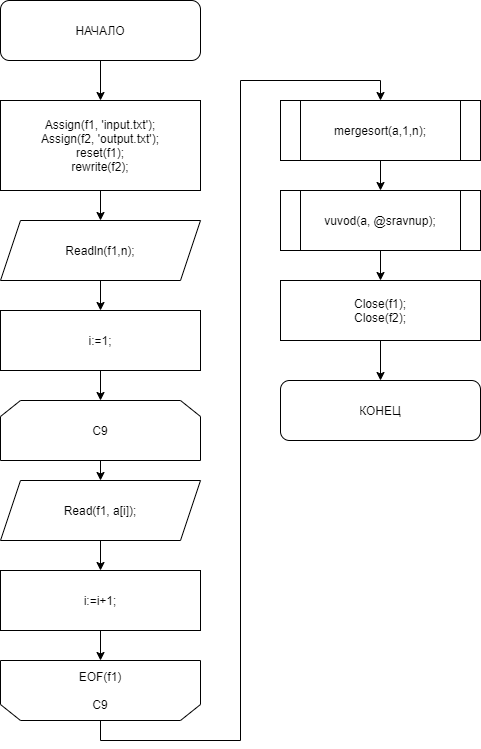
Сортировка слиянием:









***Листинг:***

Сортировка выбором:

program lab5;

type

fun = function(x, y: longint): boolean; [\\ тип](file:///\\тип) компаратора

aa = array [1..1001] of longint; \\ тип массива элементов

var

sr: boolean;

b: aa;

n, i: longint;

f1, f2: text;

{$F+}

function sravnup(x, y: longint): boolean; far; \\ компаратор для сортировки по-

begin \\ возрастанию

if x > y then

sr := true

else

sr := false;

end;

function sravndown(x, y: longint): boolean; far; \\ компаратор для сортировки по-

begin \\ убыванию

if x < y then

sr := true

else

sr := false;

end;

{$F-}

procedure vub(a: aa; f: fun); \\ процедура сортировки выбором

var

i, j, min, t: longint;

begin

for i := 1 to n-1 do

begin

min := i;

for j := i + 1 to n do

begin

f(a[min], a[j]);

if sr then

min := j;

end;

if min <> i then begin

t := a[i];

a[i] := a[min];

a[min] := t;

end;

end;

for i := 1 to n do

begin

write(f2, a[i],' ');

end;

end;

begin

assign(f1, 'input.txt'); \\ объявление файлов ввода и

assign(f2, 'output.txt');\\ вывода

reset(f1); \\ открытие 1 файла

rewrite(f2); \\ открытие 2 файла

readln(f1,n); \\ считывание 1 строки, количество числа элементов

i:=1;

repeat \\ цикл чтения всего оставшегося файла до конца и занесения чисел в массив

read(f1,b[i]);

inc(i);

until eof(f1) ;

//vub(b, @sravnup); \\ вызов процедуры сортировки по возрастанию

vub(b, @sravndown); \\ вызов процедуры по убыванию

close(f1); \\ закрытие 1 файла

close(f2); \\ закрытие 2 файла

end.

Сортировка слиянием:

program lab5;

type

fun = function(x, y: longint): boolean; \\ тип компаратора

massiv = array[1..1000001] of longint; \\ тип массива элементов

var

sr: boolean;

n, i: longint;

A, mas: massiv;

f1, f2: text;

{$F+}

function sravndown(x, y: longint): boolean; \\ компаратор по убывания

begin

if x < y then

sr := false

else

sr := true;

end;

function sravnup(x, y: longint): boolean; \\ компаратор по возрастанию

begin

if x < y then

sr := true

else

sr := false;

end;

{$F-}

procedure Merge(var A: massiv; first, last: longint); \\ процедура слияния

var

middle, start, final, j: longint;

begin

middle := (first + last) div 2;

start := first;

final := middle + 1;

for j := first to last do

begin

if (start <= middle) and ((final > last) or (A[start] < A[final])) then

begin

mas[j] := A[start];

inc(start);

end

else

begin

mas[j] := A[final];

inc(final);

end;

end;

for j := first to last do A[j] := mas[j];

end;

procedure MergeSort(var A: massiv; first, last: longint); \\ рекурсивная процедура

begin \\ сортировки

if first < last then

begin

MergeSort(A, first, (first + last) div 2);

MergeSort(A, (first + last) div 2 + 1, last);

Merge(A, first, last);

end;

end;

procedure vuvod(a: massiv; f: fun); \\ процедура записи чисел во 2 файл

var

I : longint;

begin

f(A[1],A[2]);

if sr = true then

begin

for i := 1 to n do \\ запись по возрастанию

begin

write(f2, a[i], ' ');

end;

end

else

begin

for i := n downto 1 do \\ запись по убыванию

begin

write(f2, a[i], ' ');

end;

end;

end;

begin

assign(f1, 'input.txt' ); \\ объявление файлов ввода и вывода

assign(f2, 'output.txt' );

reset(f1);

rewrite(f2);

readln(f1,n);

i:=1;

while not eof(f1) do

begin

read(f1,A[i]);

inc(i);

end;

mergesort(a, 1, n); \\ вызов процедуры сортировки

//vuvod(a, @sravnup); \\ запись массива по возрастанию

vuvod(a, @sravndown); \\ запись массива по убыванию

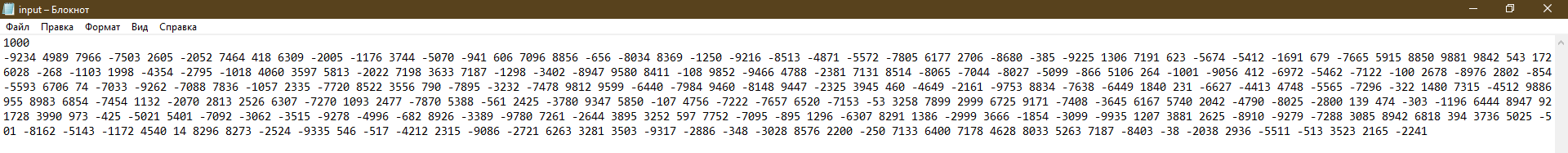
close(f1);

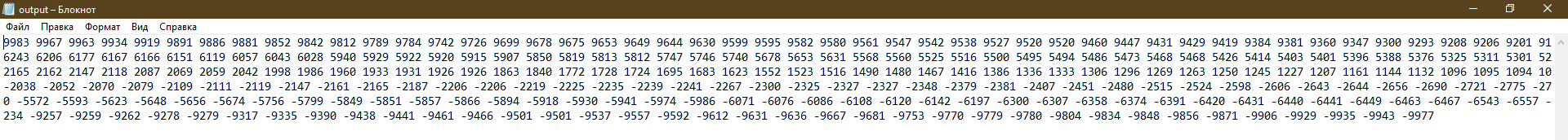
close(f2);

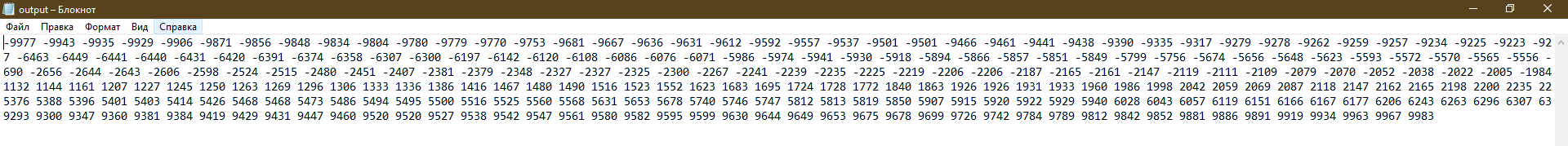
end.

***Экранные формы:***

Сортировка выбором

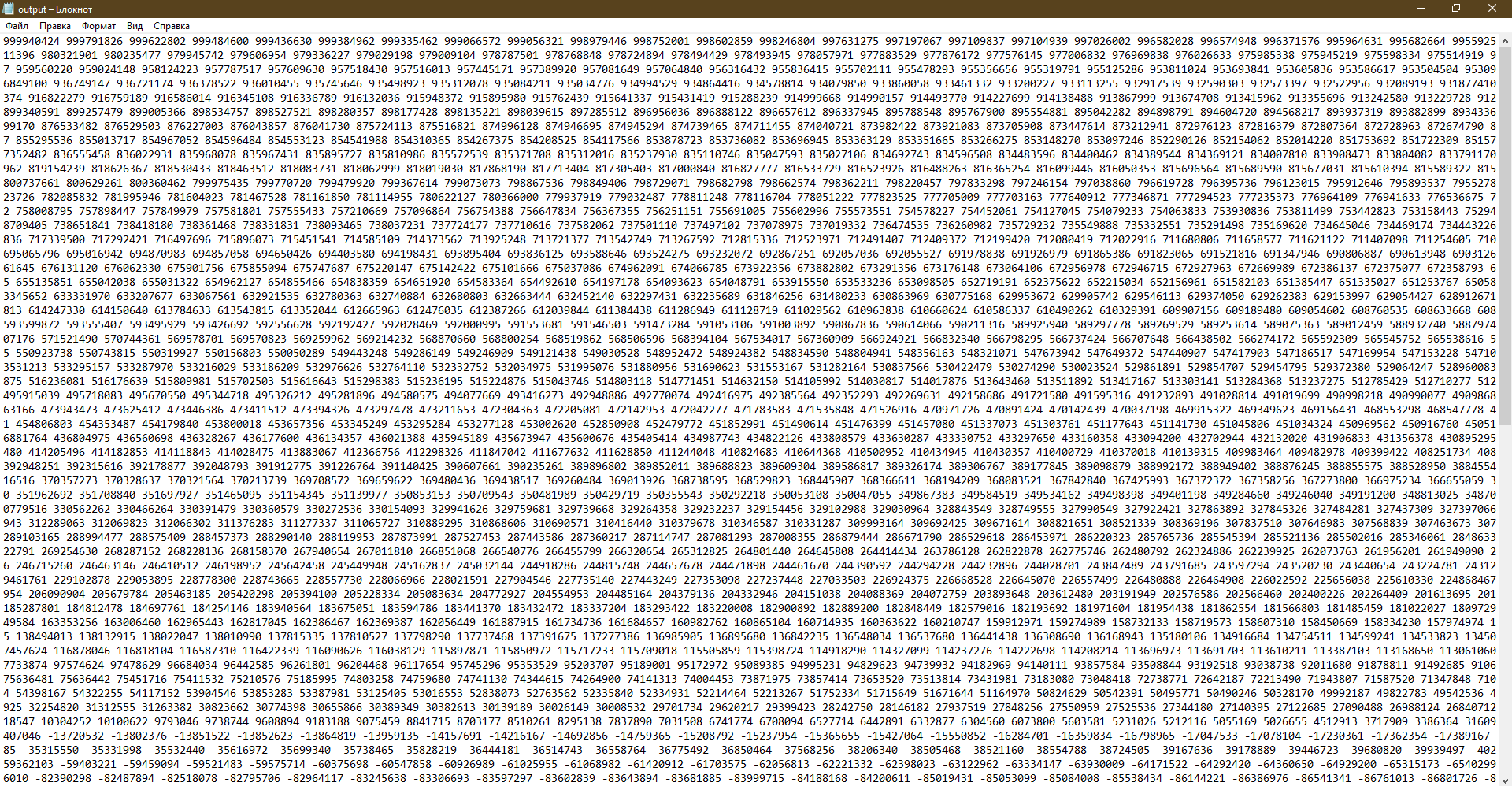


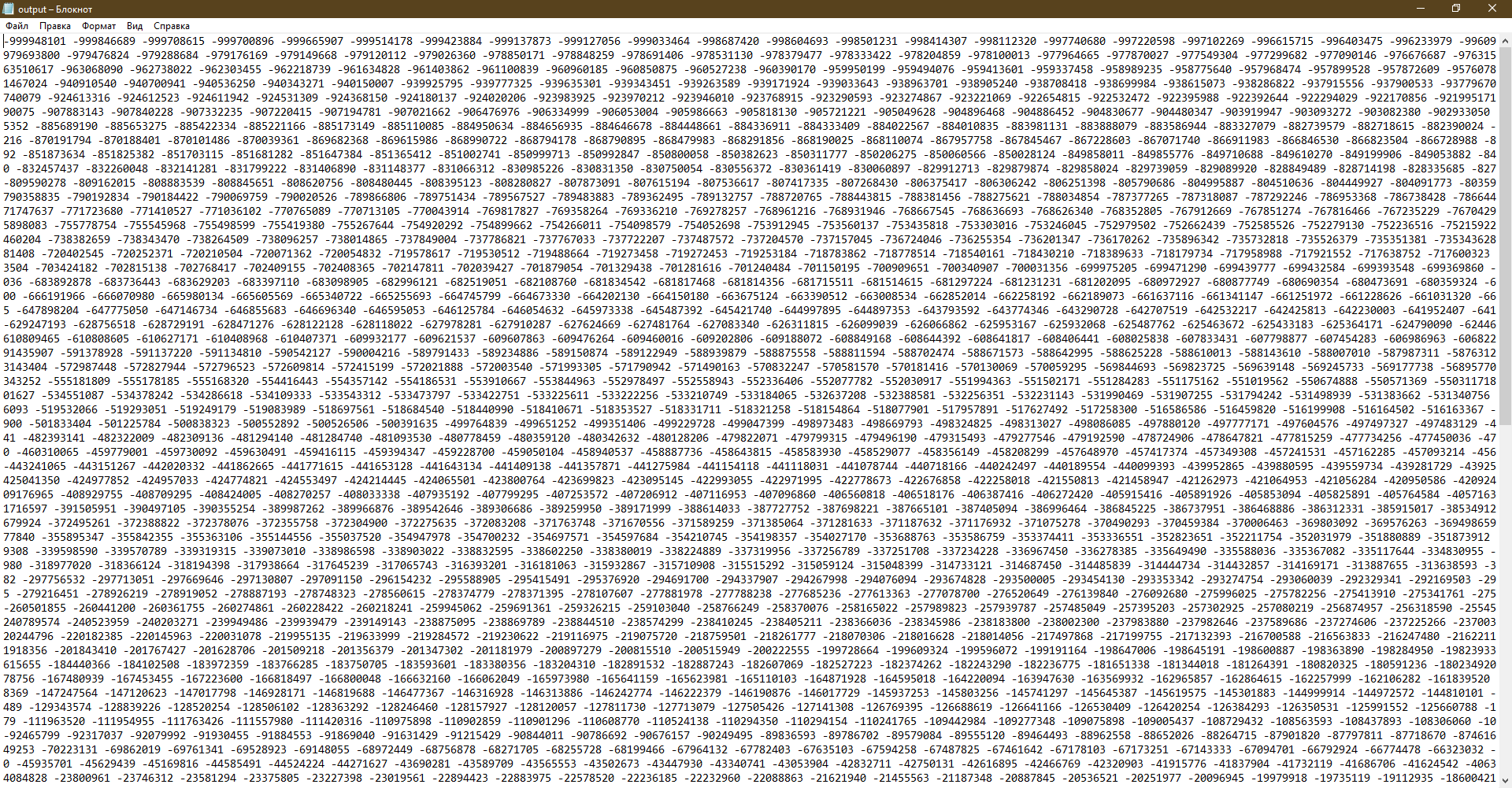




Сортировка слиянием:







***Вывод:***

В ходе выполнения лабораторной работы были получены базовые сведения о наиболее известных алгоритмах сортировки, изучены принципы работы с текстовыми файлами,

были реализованы программы сортировки данных с помощью алгоритмов выбором и слияния, также был создан компаратор, который определяет вид сортировки: по возрастанию или по убыванию. Данный компаратор передавался в процедуры в качестве параметра по адресу. Его изменение позволяло менять порядок сортировки.

Данные для сортировки считывались из файла «input», а отсортированный массив записывался в файл «output». Для работы с ними, файлы привязывались к переменным с помощью «ASSIGN» и открывались внутри программы для дополнительной записи с помощью «RESET» и перезаписи с помощью «REWRITE». После завершения всех необходимых для сортировки действий внутри программы файлы закрывались с помощью «CLOSE».