Вариант:200

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

Задание:

1. Реализовать сортировку данных с помощью алгоритма подсчетом.
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 реализации самостоятельно подготовить варианты входных данных (при этом объем текстовых файлов должен позволять оценить скорость работы программы).

Схема алгоритмов:

Схема алгоритма ко первому заданию

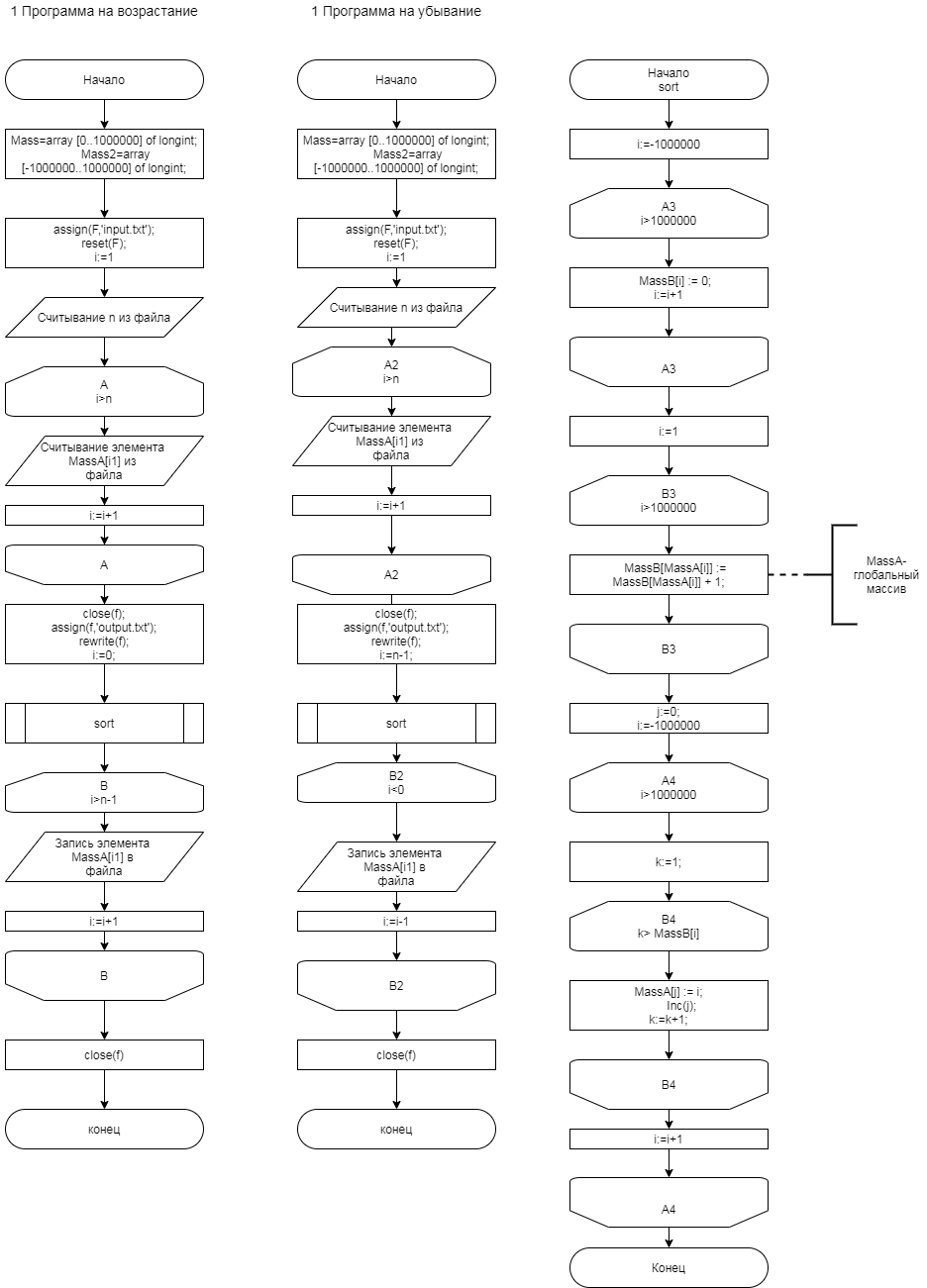
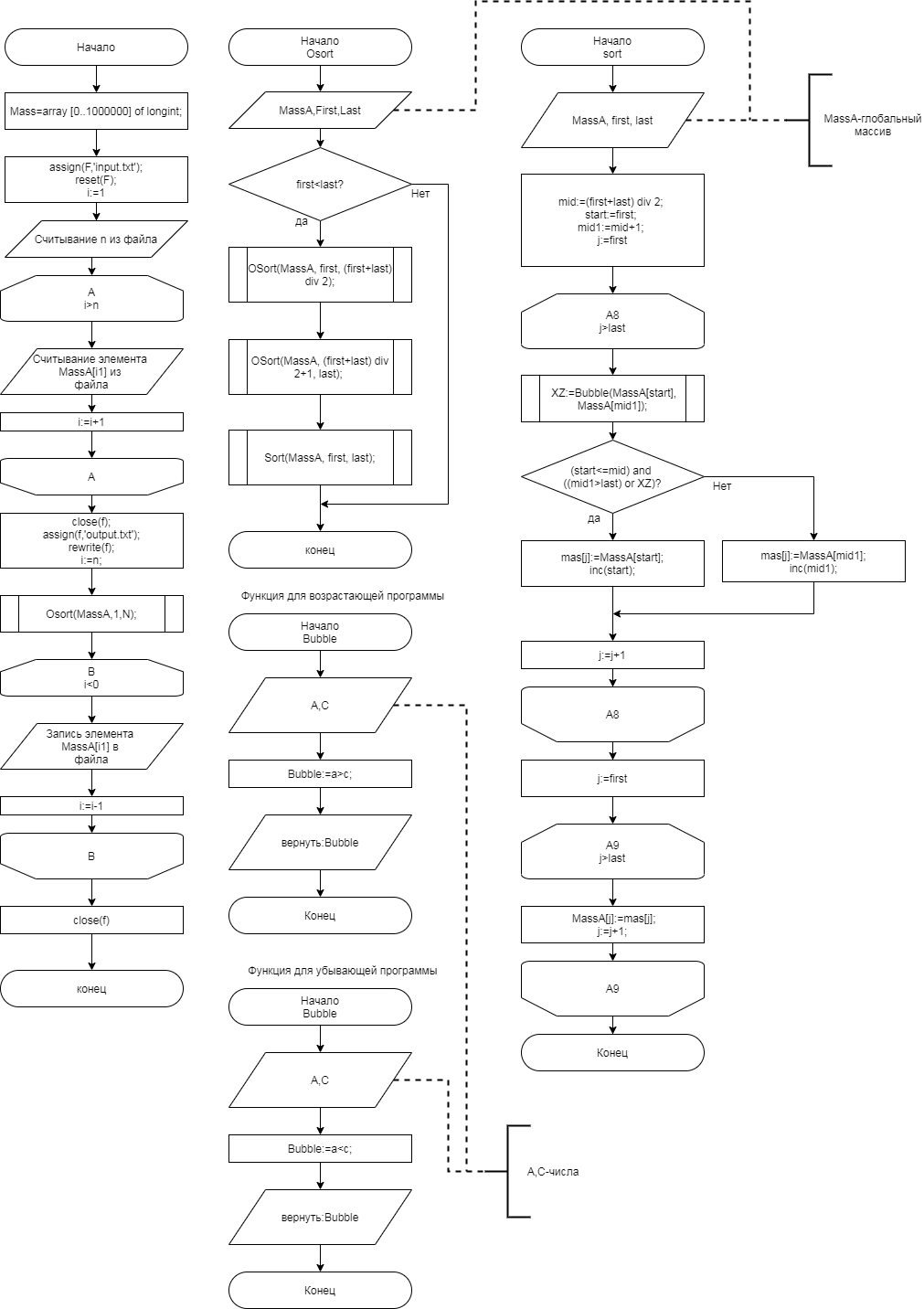


Схема алгоритма ко второму заданию



Исходный код:

* 1 программа на возрастание:

**program** qq;

**type** Mass=**array** [0..1000000] **of** longint;

Mass2=**array** [-1000000..1000000] **of** longint;

**var** f:text;

MassA:Mass;

n,B,i1:longint;

**procedure** sort;

**var** MassB:mass2;

i, j:longint;

k:integer;

**begin**

**for** i := -1000000 **to** 1000000 **do**

MassB[i] := 0;

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

MassB[MassA[i]] := MassB[MassA[i]]+1 ;

j := 0;

**for** i := -1000000 **to** 1000000 **do**

**for** k := 1 **to** MassB[i] **do**

**begin**

MassA[j] := i;

Inc(j);

**end**;

**end**;

**begin**

assign(f,'input.txt');

reset(f);

read(f,n);

**for** i1:=1 **to** n **do**

read(f,MassA[i1]);

close(f);

sort;

assign(f,'output.txt');

rewrite(f);

**for** i1:=0 **to** n-1 **do**

write(f,MassA[i1],' ');

close(f);

**end**.

* 1 программа на убывание:

**program** qq;

**type** Mass=**array** [0..1000000] **of** longint;

Mass2=**array** [-1000000..1000000] **of** longint;

**var** f:text;

MassA:Mass;

n,B,i1:longint;

**procedure** sort;

**var** MassB:mass2;

i, j:longint;

k:integer;

**begin**

**for** i := -1000000 **to** 1000000 **do**

MassB[i] := 0;

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

MassB[MassA[i]] := MassB[MassA[i]] + 1;

j := 0;

**for** i := -1000000 **to** 1000000 **do**

**for** k := 1 **to** MassB[i] **do**

**begin**

MassA[j] := i;

Inc(j);

**end**;

**end**;

**begin**

assign(f,'input.txt');

reset(f);

read(f,n);

**for** i1:=1 **to** n **do**

read(f,MassA[i1]);

close(f);

sort;

assign(f,'output.txt');

rewrite(f);

**for** i1:=n-1 **downto** 0 **do**

write(f,MassA[i1],' ');

close(f);

**end**.

* 2 программа на возрастание:

**program** qq;

**type** Mass=**array** [1..1000000] **of** longint;

**var** f:text;

MassA:Mass;

n,B,i1:longint;

**procedure** Sort(**var** MassA: Mass; first, last:Longint);

**var** mid, start, mid1, j: Longint;

mas: mass;

XZ:boolean;

**function** Bubble(a, c: integer): boolean;

**begin**

Bubble:=a>c;

**end**;

**begin**

mid:=(first+last) **div** 2;

start:=first;

mid1:=mid+1;

**for** j:=first **to** last **do**

**if** (start<=mid) **and** ((mid1>last) **or** Bubble(MassA[start],MassA[mid1])) **then**

**begin**

mas[j]:=MassA[start];

inc(start);

**end**

**else**

**begin**

mas[j]:=MassA[mid1];

inc(mid1);

**end**;

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

**end**;

**procedure** OSort(**var** MassA:mass; first, last:Longint);

**begin**

**if** first<last **then**

**begin**

OSort(MassA, first, (first+last) **div** 2);

OSort(MassA, (first+last) **div** 2+1, last);

Sort(MassA, first, last);

**end**;

**end**;

**begin**

assign(f,'input.txt');

reset(f);

readln(f,n);

**for** i1:=1 **to** n **do**

read(f,MassA[i1]);

close(f);

Osort(MassA,1,N);

assign(f,'output.txt');

rewrite(f);

**for** i1:=n **downto** 1 **do**

write(f,MassA[i1],' ');

close(f);

**end**.

* 2 программа на убывание:

**program** qq;

**type** Mass=**array** [1..1000000] **of** longint;

**var** f:text;

MassA:Mass;

n,B,i1:longint;

**procedure** Sort(**var** MassA: Mass; first, last:Longint);

**var** mid, start, mid1, j: Longint;

mas: mass;

XZ:boolean;

**function** Bubble(a, c: integer): boolean;

**begin**

Bubble:=a<c;

**end**;

**begin**

mid:=(first+last) **div** 2;

start:=first;

mid1:=mid+1;

**for** j:=first **to** last **do**

**if** (start<=mid) **and** ((mid1>last) **or** Bubble(MassA[start],MassA[mid1])) **then**

**begin**

mas[j]:=MassA[start];

inc(start);

**end**

**else**

**begin**

mas[j]:=MassA[mid1];

inc(mid1);

**end**;

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

**end**;

**procedure** OSort(**var** MassA:mass; first, last:Longint);

**begin**

**if** first<last **then**

**begin**

OSort(MassA, first, (first+last) **div** 2);

OSort(MassA, (first+last) **div** 2+1, last);

Sort(MassA, first, last);

**end**;

**end**;

**begin**

assign(f,'input.txt');

reset(f);

readln(f,n);

**for** i1:=1 **to** n **do**

read(f,MassA[i1]);

close(f);

Osort(MassA,1,N);

assign(f,'output.txt');

rewrite(f);

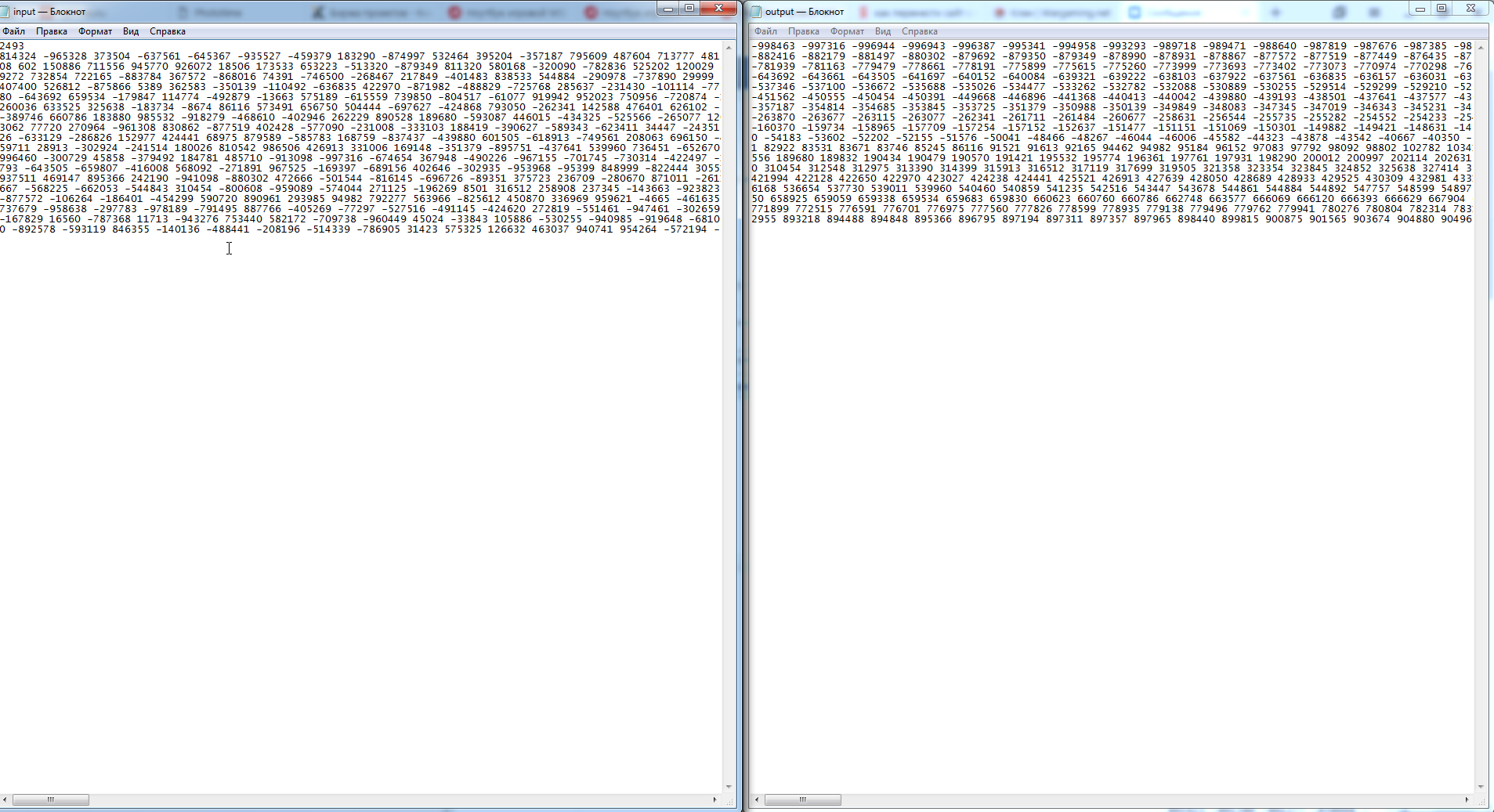
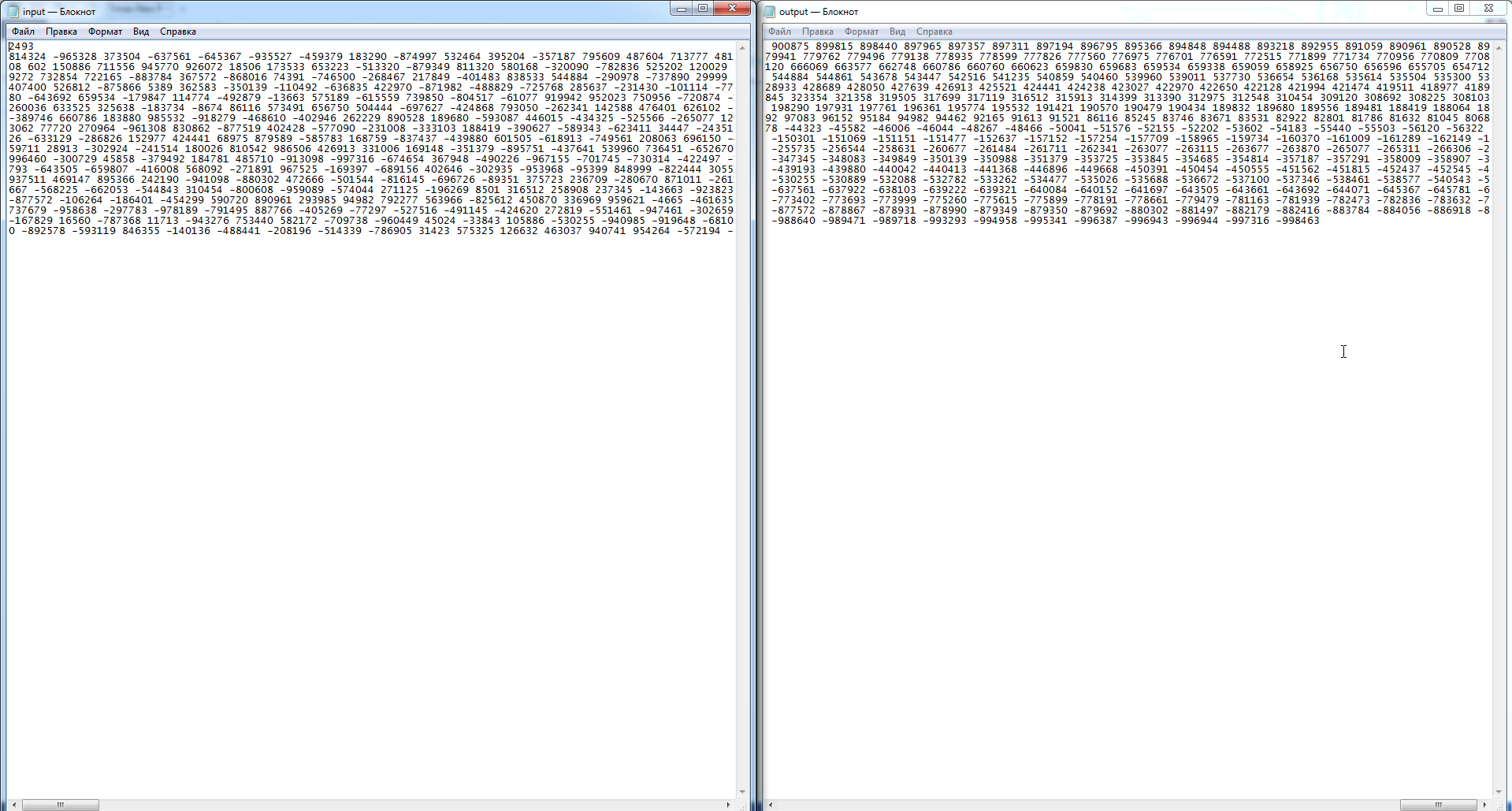
**for** i1:=n **downto** 1 **do**

write(f,MassA[i1],' ');

close(f);

**end**.

Фото отчет выполненой программы:



Вывод: В ходе выполнение лабораторной работы по программированию были получены:

* Навыки работы с текстовым файлом
* Навыки работы с типизированным файлом
* Навык работы с сортировки и добавление в них компаратора и работа с ним.

При выполнение лабораторной работы были выявлены сложности такие как:

* Изучение компаратора и добавление его в программу.
* Работа с текстовым файлом.
* Работа с сортировками, более подробное изучение их.