

```

Program NaturalMergeSort;

Uses Crt;

Type item=integer;
    tape=file of item;

Var c:tape;
    n,trak:integer;
    buf:item;
    smer,nacin:boolean;
    temp:array[1..2] of integer;
    x:char;

Function R (var x:tape):item;
(* R(x) nadomesca x^ - zaradi datotecnega izravnalnika *)
    var buf:item;
    begin
        read(x,buf);
        seek(x,filepos(x)-1);
        R:=buf;
    end;

Function Cmp(a,b:item):boolean;
    begin
        if smer then Cmp:=a<b
            else Cmp:=b<a;
    end;

Procedure List(var f:tape; n:integer);
    var buf,buf1:item;
    begin
        if smer then buf1:=0
            else buf1:=MAXINT;
        reset(f);
        writeln('TRAK ',n:2);
        while not eof(f) do begin
            read(f,buf);
            if Cmp(buf,buf1) then write('|',buf:2)
                else write(buf:3);

            buf1:=buf;
        end;
        writeln;
        reset(f);
        x:=ReadKey;
    end;

Procedure Naturalmerge;

```

```

var l:integer;
    eor:boolean;
    a,b:tape;

procedure copy(var x,y:tape);
    var buf:item;
    begin
        read(x,buf); write(y,buf);
        temp[trak]:=buf;
        if eof(x) then eor:=true
            else eor:=Cmp(R(x),buf);
    end;

procedure copyrun(var x,y:tape);
    begin
        repeat
            copy(x,y);
        until eor;
    end;

procedure distribute;
    begin
        repeat
            trak:=1;
            copyrun(c,a);
            trak:=2;
            if not eof(c) then copyrun(c,b);
        until eof(c);
        List(a,1);
        List(b,2);
    end;

procedure distribute1;
    begin
        trak:=1;
        copyrun(c,a);
        trak:=2;
        if not eof(c) then copyrun(c,b);
        repeat
            trak:=1;
            if (not eof(c))and(Cmp(temp[1],R(c))) then copyrun(c,a);
            if not eof(c) then copyrun(c,a);
            trak:=2;
            if (not eof(c))and(Cmp(temp[2],R(c)))then copyrun(c,b);
            if not eof(c) then copyrun(c,b);
        until eof(c);
        List(a,1);
        List(b,2);
    end;

procedure mergerun;

```

```

begin
  repeat
    if Cmp(R(a),R(b)) then begin
      copy(a,c);
      if eor then copyrun(b,c);
    end
    else begin
      copy(b,c);
      if eor then copyrun(a,c);
    end;
  until eor;
end;

procedure merge;
begin
  while (not eof(a))and(not eof(b)) do begin
    mergerun;
    inc(l);
  end;
  while not eof(a) do begin
    copyrun(a,c);
    inc(l);
  end;
  while not eof(b) do begin
    copyrun(b,c);
    inc(l);
  end;
  List(c,0);
end;

procedure merge1;
begin
  while not eof(b) do
    begin
      mergerun;
      inc(l);
    end;
  if not eof(a) then
    begin
      copyrun(a,c);
      inc(l);
    end;
  List(c,0);
end;

```

```

begin

```

```

assign(a, 'zac1.dat');
assign(b, 'zac2.dat');
repeat
  rewrite(a);
  rewrite(b);
  reset(c);
  if nacin then distribute
    else distribute1;
  reset(a);
  reset(b);
  rewrite(c);
  l:=0;
  if nacin then merge
    else merge1;
until l=1;
close(b); close(a);
erase(b); erase(a);
end;

```

Begin

```

nacin:=true; (* enostaven distribute in zapleten merge *)
(* nacin:=false; popravljen distribute in enostaven merge *)
smer:=true; (* narascajocje *)
(* smer:=false; padajocje *)
assign(c, 'zac.dat');
rewrite(c);
for n:=1 to 50 do
  begin
    buf:=random(99)+1;
    write(c, buf)
  end;
reset(c);
List(c, 3);
Naturalmerge;
close(c);
erase(c);
x:=Readkey;

```

End.