```
{ Sa presupunem ca avem de sortat un fisier de dimensiuni foarte mari
  (sute de mii sau milioane de inregistrari). Intr-o asemenea situatie
  nu putem incarca in memorie continutul fisierului, ci trebuie sa facem
  sortarea direct pe disc, incarcand in memorie doar o parte din date.

  O varianta de rezolvare este sortarea prin interschimbare direct pe
  fisier. Se parcurge fisierul in mod repetat si se citesc in memorie
  tot cate doua inregistrari consecutive. Daca se constata ca doua
  inregistrari consecutive sunt in ordinea gresita, atunci se inverseaza
  cele doua inregistrari intre ele, dar direct in fisier, nu doar in
  memorie. }
program Sortare_fisier_cu_tip_de_dimensiune_mare;

uses Crt;

type
    { Definim tipul de date care se va pastra in fisier. Putem
      considera un tip de date simplu, care contine doar un
      numar Longint. }
    TDate = record
          info : Longint;
    end;

var
   { Pentru a putea lucra cu fisierul pe disc, definim o variabila de
     tip fisier de tipul definit mai sus. }
   fis : file of TDate;


procedure Genereaza;
var i, n : Longint;
    d : TDate;
begin
     { Pentru a crea fisierul pe disc, generam numere aleatoare pe
       care le scriem rand pe rand in fisier. Este important sa
       scriem datele in fisier rand pe rand, si sa nu le pastram
       in memorie pentru a le scrie pe toate la sfarsit, deoarece
       avem un numar foarte mare de inregistrari. }
     n := 1000;
     Rewrite(fis);
     Randomize;
     WriteLn('Generez fisierul cu ', n, ' inregistrari.');
     for i := 1 to n do
         begin
              { In mod normal functia Random returneaza tipul Integer.
                Pentru a obtine valori Longint, inmultim valorile
                returnate de doua apeluri ale functiei Random. }
              d.info := Random(MaxInt);
              d.info := d.info * Random(MaxInt);
              Write(fis, d);
         end;
     Close(fis);
     WriteLn('Fisierul a fost generat.');
end;


procedure VerificaFisierSortat;
var d1, d2 : TDate;
begin
     { Facem o verificare pentru a ne asigura ca fisierul nu este
       sortat. Pentru asta citim din fisier inregistrari succesive
```

```
       pana cand fie se termina fisierul, fie gasim doua inregistrari
       succesive care sunt in ordine gresita. Daca am gasit o asemenea
       pereche de inregistrari, atunci fisierul nu este sortat. Daca
       am ajuns la sfarsitul fisierului fara sa gasim o asemenea
       pereche, atunci fisierul este sortat. }
     WriteLn('Verific daca fisierul este sortat...');
     Reset(fis);

     { Daca fisierul are dimensiunea 0 sau 1, atunci el este sortat
       implicit.}
     if FileSize(fis) <= 1 then
        begin
             WriteLn('Fisierul este sortat.');
        end
     else
         begin
              { Daca avem cel putin doua inregistrari in fisier, atunci
                incepem sa il parcurgem.
                Citim primele doua inregistrari din fisier. }
              Read(fis, d1);
              Read(fis, d2);
              { Cat timp nu am atins sfarsit de fisier si cat timp
                cele doua inregistrari curente sunt in ordinrea
                corecta, citim mai departe cate o inregistrare. }
              while not(eof(fis)) and (d1.info < d2.info) do
                    begin
                         { Inregistrarea a doua trece acum in prima. }
                         d1 := d2;
                         { Si in inregistrarea a doua citim urmatoarea
                           intrare in fisier. }
                         Read(fis, d2);
                    end;

              { Verificam care a fost conditia de oprire. Daca a fost
                ordinea gresita intre inregistrari, atunci fisierul
                nu este sortat. }
              if (d1.info >= d2.info) then
                 WriteLn('Fisierul nu este sortat.')
              else
                 WriteLn('Fisierul este sortat.');
         end;
end;


procedure SorteazaFisierPeDisc;
var sortat : Boolean;
    d1, d2, aux : TDate;
begin
     WriteLn('Sortez fisierul direct pe disc...');
     Reset(fis);
     { Daca fisierul contine 0 sau 1 inregistrari, atunci el e deja sortat. }
     if FileSize(fis) <= 1 then
        begin
             WriteLn('Fisierul este deja sortat.');
        end
     else
         begin
              { Facem sortare prin insertie. }
              repeat
                    { Afisam pe ecran cate un caracter la fiecare parcurgere
                      a fisierului. }
```

```
                Write('#');

                sortat := True;

                { Repozitionam fisierul pe prima inregistrare. }
                Seek(fis, 0);

                { Citim prima inregistrare din fisier, dar in variabila
                  d2. d2 reprezinta inregistrarea curenta, iar d1
                  reprezinta inregistrarea precedenta. }
                Read(fis, d2);

                { Cat timp nu am ajuns la sfarsit de fisier, citim
                  cate o inregistrare si o comparam cu inregistrarea
                  precendenta. }
                while not(eof(fis)) do
                      begin
                           { Ce era inainte in variabila d2 trece acum
                             in d1. Cu alte cuvinte inregistrarea care
                             inainte este "curenta", acum devine
                             "precedenta". }
                           d1 := d2;

                           { Citim o noua inregistrare "curenta". }
                           Read(fis, d2);

                           { Daca perechea de inregistrari e in ordinea
                             gresita, le schimbam intre ele, dar direct
                             in fisier. }
                           if d1.info > d2.info then
                              begin
                                   { Pentru inteschimbarea
                                     inregistrarilor direct in fisier
                                     nu avem nevoie de variabile
                                     auxiliare. Tot ce trebuie sa facem
                                     este sa ne pozitionam pe
                                     inregistrarea "d2" din fisier si sa
                                     scriem in locul ei inregistrarea
                                     "d1" din memorie, iar apoi sa ne
                                     pozitionam pe inregistrarea "d1" din
                                     fisier si sa scriem in locul ei
                                     inregistrarea "d2" din memorie. }

                                   { In acest moment indicatorul de
                                     fisier este pozitionat imediat
                                     dupa inregistrarea "d2". Ca urmare
                                     venim inapoi cu o pozitie pentru
                                     a putea scrie inregistrarea "d1"
                                     din memorie. }
                                   Seek(fis, FilePos(fis)-1);
                                   Write(fis, d1);

                                   { Acum indicatorul de fisier a ajuns
                                     din nou dupa inregistrarea "d2".
                                     Venim inapoi cu doua pozitii pentru
                                     a ajunge la "d1". }
                                   Seek(fis, FilePos(fis) - 2);

                                   { Scriem aici inregistrarea "d2"
                                     din memorie. }
                                   Write(fis, d2);
```

```
                                        { Acum indicatorul de fisier este
                                          pozitionat dupa "d1", adica la "d2".
                                          Trebuie sa revenim la pozitia
                                          initiala, adica dupa "d2". Pentru
                                          aceasta mergem inainte cu o
                                          pozitie. }
                                        Seek(fis, FilePos(fis) + 1);

                                        { Dupa ce am inversat inregistrarile
                                          pe disc, le inversam si in memorie.}
                                        aux := d1;
                                        d1 := d2;
                                        d2 := aux;

                                        sortat := False;
                                    end;
                            end;
                until sortat;
                WriteLn;
                WriteLn('Am sortat fisierul.');
        end;
end;


begin
     ClrScr;

     Assign(fis, 'sort.dat');

     Genereaza;
     VerificaFisierSortat;
     SorteazaFisierPeDisc;
     VerificaFisierSortat;
end.
```