

{Z1: Sa standardnog ulaza učitati niz cjelih brojeva većih od nule(ne većih od 100) u niz. Kraj unosa, unesena vrijednost 0! U datoteku izlaz.dat dati niz sortiran od većeg ka manjem, bez ponavljanja istih brojeva. Koristiti nizove, procedure i funkcije.}

USES CRT;

Const Izlazna\_datoteka='C:\Izlaz.Dat';  
Max\_Elemenata=500;

Type Cjeli\_Broj=Integer;  
Elemenat=Cjeli\_Broj;  
Niz\_Elemenata=Array[1..Max\_Elemenata] of Integer;

Var Niz:Niz\_Elemenata;  
Vrijednost:Cjeli\_Broj;  
Broj\_Elemenata,I,J:Integer;  
Dat:Text;

PROCEDURE Unos\_Vrijednosti\_U\_Niz;

FUNCTION Provjera\_Vrijednosti:Boolean;  
Begin  
If (Vrijednost>0) and (Vrijednost<100) then  
Provjera\_Vrijednosti:=True else  
Provjera\_Vrijednosti:=False;  
End;

FUNCTION Unesen\_Broj:Boolean;  
Begin  
Readln(Vrijednost);  
If Vrijednost=0 then Unesen\_Broj:=False  
Else Unesen\_Broj:=True;  
End;

PROCEDURE Inicijalizacija\_Niza;  
Begin  
For I:=1 to Max\_Elemenata do Niz[I]:=0;  
Broj\_Elemenata:=0;  
I:=1;  
J:=1;  
End;

Begin  
Inicijalizacija\_Niza;  
While Unesen\_Broj do  
Begin  
If Provjera\_Vrijednosti=True then  
Begin  
Niz[I]:=Vrijednost;  
Inc(I);  
Inc(Broj\_Elemenata);  
End;  
End;  
End;

PROCEDURE Sortiranje\_Niza(Var Niz\_1:Niz\_Elemenata);  
Var Pom:Cjeli\_Broj;  
Begin  
For I:=1 to Broj\_Elemenata-1 do  
Begin  
For J:=I+1 to Broj\_Elemenata do  
Begin  
If Niz\_1[I]<Niz\_1[J] then

```

Begin
  Pom:=Niz_1[J] ;
  Niz_1[J]:=Niz_1[I];
  Niz_1[I]:=Pom;
End;
End;
End;
End;

```

```

PROCEDURE Stampanje_Rezultata_U_Datoteku;

```

```

FUNCTION Ispisujem:Boolean;
Begin
  If Niz[I]=Niz[I+1] then
    Ispisujem:=False else
    Ispisujem:=True;
End;
Begin
  Assign(Dat,Izlazna_Datoteka);
  Rewrite(Dat);
  For I:=1 to Broj_Elemenata do
    Begin
      If Ispisujem then Write(Dat,Niz[I]);
    End;
    Close(Dat);
  End;

```

```

Begin
  CLRSCR;
  Unos_Vrijednosti_U_Niz;
  Sortiranje_Niza(Niz);
  Stampanje_Rezultata_U_Datoteku;
End.

```

{Zad 2 sa ispitnog roka 27.06.2001 Data je datoteka ulaz.dat sa strukturom: slog1=record p1,p2:integer; end;  
U datoteku izlaz.dat sa strukturom: slog2=record p1,p2,p3,p4:integer; p5:real; end; presuti sadrzaj datoteke ulaz.dat  
gdje su:  $p3=p1+p2$ ;  $p4=p1*p2$ ;  $p5=(p1-p2)/N$  za  $p1>p2$ ;  $p5=(p2-p1)/N$  za  $p1<p2$ ;  $p5=p1/N$  za  $p1=p2$ , gdje je N broj  
slogova u datoteci ulaz.dat.}

Uses CRT;

Const Ulazna\_Datoteka='ulaz.dat';

Izlazna\_Datoteka='izlaz.dat';

Max\_Elemenata=550;

Type Slog1=Record

P1,P2:Integer;

End;

Slog2=Record

P1,P2,P3,P4:Integer;

P5:Real;

End;

Niz\_Sloga1=Array[1..Max\_Elemenata] of Slog1;

Niz\_Sloga2=Array[1..Max\_Elemenata] of Slog2;

Var Niz1:Niz\_Sloga1;

Niz2:Niz\_Sloga2;

Dat1:File Of Slog1;

Dat2:File Of Slog2;

Elemenata:Integer;

I,J:Integer;

PROCEDURE Citanje\_Iz\_Dat1;

PROCEDURE Inicijalizacija;

Begin

Elemenata:=0;

I:=1;

End;

Begin

Assign(Dat1,Ulazna\_Datoteka);

Reset(Dat1);

While NOT Eof(Dat1) do

Begin

Read(Dat1,Niz1[I]);

Niz2[I].P1:=Niz1[I].P1;

Niz2[I].P2:=Niz1[I].P2;

Inc(I);

Inc(Elemenata);

End;

Close(Dat1);

End;

PROCEDURE Upisivanje\_U\_Dat2;

Var Brojac:Integer;

Begin

J:=1;

Brojac:=1;

Assign(Dat2,Izlazna\_Datoteka);

Rewrite(Dat2);

While Brojac<=Elemenata do

Begin

With Niz2[J] do

Begin

P3:=P1+P2;

P4:=P1\*P2;

```

    If P1>P2 Then P5:=(P1-P2)/Elemenata;
    If P1<P2 Then P5:=(P2-P1)/Elemenata;
    If P1=P2 Then P5:=P1/Elemenata;
End;
Write(Dat2,Niz2[J]);
Inc(J);
Inc(Broj2);
End;
Close(Dat2);
End;

```

```

PROCEDURE Citanje_Iz_Dat;
Var K:Integer;
Begin
    Assign(Dat2,Izlazna_Datoteka);
    Reset(Dat2);K:=1;
    While Not EOF(Dat2) do
        Begin
            Read(Dat2,Niz2[K]);
            Inc(K);
        End;
    Close(Dat2);
    For I:=1 to K do
        Writeln('P1:',Niz2[I].P1,' ',P2:',Niz2[I].P2);
        Writeln('P3:',Niz2[I].P3,' ',Niz2[I].P4);
    End;
Begin
    Clrscr;
    Citanje_Iz_Dat1;
    Upisivanje_U_Dat2;
    Citanje_Iz_Dat;
End.

```

```

Program Zad10;
Uses CRT;
Const Ulazna_Datoteka='ulaz.dat';
      Izlazna_Datoteka='izlaz.dat';
      Max_Elemenata=550;
Type Slog1=Record
    P1,P2:Integer;
End;

    Slog2=Record
    P1,P2,P3,P4:Integer;
    P5:Real;
End;

    Niz_Sloga1=Array[1..Max_Elemenata] of Slog1;
    Niz_Sloga2=Array[1..Max_Elemenata] of Slog2;

```

```

Var Niz1:Niz_Sloga1;
    Niz2:Niz_Sloga2;
    Dat1:File Of Slog1;
    Dat2:File Of Slog2;
    Elemenata:Integer;

```

```

PROCEDURE Upisivanje_U_Datoteku1;
Var Broj1,Broj2,K:Integer;
    C:char;

```

```

FUNCTION Unos:Boolean;

```

```

Begin
  Writeln('Jos');
  Readln(C);
  If Upcase(C)='N' then Unos:=False else
    Unos:=True;
End;

```

```

Begin
  K:=0;
  Assign(Dat1,Ulazna_Datoteka);
  Rewrite(Dat1);
  Repeat
    INC(K);
    With Niz1[K] do
      Begin
        Writeln('P1: ');
        Readln(P1);
        Writeln('P2: ');
        Readln(P2);
      End;
    Write(Dat1,Niz1[K]);
  Until NOT Unos;
  Close(Dat1);
End;

```

```

Begin
  Upisivanje_U_Datoteku1;
End.

```

{Zadatak sa auditornih vjezbi: Iz ulazne datoteke prebaciti niz brojeva u izlaznu datoteku sortirajuci ih!}

```

{Za unos u ulaznu datoteku sluzi program 1 lupis.pas}
USES CRT;
Const Ulazna_Datoteka='ulaz.dat';
      Izlazna_Datoteka='izlaz.dat';

```

```

Type Datoteka=File of Integer;
      Niz_Elemenata=Array[1..1000] of Integer;
Var Dat1,Dat2:Datoteka;
      Niz:Niz_Elemenata;
      I,J,Elemenata:Integer;
PROCEDURE Upis_Elemenata_Iz_Datoteke_U_Niz;

```

```

PROCEDURE Inicijalizacija_Datoteke;
Begin
  Assign(Dat1,Ulazna_Datoteka);
  Reset(Dat1);
  I:=1;
  Elemenata:=0;
End;

```

```

Begin
  Inicijalizacija_Datoteke;
  While NOT Eof(Dat1) do
    Begin
      Read(Dat1,Niz[I]);
      Inc(I);
      Inc(Elemenata);
    End;
End;

```

```

PROCEDURE Sortiranje;
Var POM:Integer;
Begin
  For I:=1 to Elemenata-1 do
    Begin
      For J:=I+1 to Elemenata do
        Begin
          If Niz[I]<Niz[J] then
            Begin
              Pom:=Niz[I];
              Niz[I]:=Niz[J];
              Niz[J]:=Pom;
            End;
          End;
        End;
      End;
    End;
  End;

```

```

PROCEDURE Upis_Sortiranog_Niza_U_Izlaznu_Datoteku;
Var Brojac:Integer;
Begin
  Sortiranje;
  Assign(Dat2,Izlazna_Datoteka);
  Rewrite(Dat2);
  I:=1;
  While Brojac<=Elemenata do
    Begin
      Write(Dat2,Niz[I]);
      Inc(Brojac);
      Inc(I);
    End;
  Close(Dat2);
End;

```

```

PROCEDURE Test_Ispis; {Za provjeru rezultata}

```

```

PROCEDURE Inicijalizacija_Datoteke;
Begin
  Assign(Dat2,Izlazna_Datoteka);
  Reset(Dat2);
  I:=1;
End;

```

```

Begin
  Inicijalizacija_Datoteke;
  While NOT Eof(Dat2) do
    Begin
      Read(Dat2,Niz[I]);
      Inc(I);
    End;
  For I:=1 to Elemenata do Writeln(Niz[I]);
End;

```

```

Begin
  Clrscr;
  Upis_Elemenata_Iz_Datoteke_U_Niz;
  Upis_Sortiranog_Niza_U_Izlaznu_Datoteku;
End.

```

```

Const Ulazna_Datoteka='ulaz.dat';
      Izlazna_Datoteka='izlaz.dat';

Type Datoteka=File of Integer;

Var Dat1,Dat2:Datoteka;
    Broj,Brojac:Integer;
    C:Char;

PROCEDURE Upis_U_Datoteku_Ulaz;

PROCEDURE Inicijalizacija_Datoteke;
Begin
  Assign(Dat1,Ulazna_Datoteka);
  Rewrite(Dat1);
  Brojac:=0;
End;

FUNCTION Unos:Boolean;
Begin
  Writeln('Jos(D/N)');
  Readln(C);
  If Uppcase(C)='N' then Unos:=False else
    Unos:=True;
End;

Begin
  Inicijalizacija_Datoteke;
  Repeat
    Writeln('INT: ');
    Readln(Broj);
    Write(Dat1,Broj);
    Inc(Brojac);
  Until NOT Unos;
  Close(Dat1);
End;

PROCEDURE Citanje_Iz_Datoteke;

PROCEDURE Inicijalizacija_Datoteke;
Begin
  Assign(Dat1,Ulazna_Datoteka);
  Reset(Dat1);
End;

Var BR:Integer;
Begin
  Inicijalizacija_Datoteke;
  While NOT Eof(Dat1) do
    Begin
      Read(Dat1,Br);
      Writeln(Br);
    End;
  Close(Dat1);
End;

Begin
  Upis_U_Datoteku_Ulaz;
  Citanje_Iz_Datoteke;

End.

```

{Zadatak sa auditornih vježbi Naci arifmeticku sredinu n-realnih brojeva(max 200). Na standardni izlaz odstampati arifmeticku sredinu i listu unjetih brojeva.}

```
Const Max=200;  
Type Niz_Realnih_Brojeva=Array[1..Max] of Real;
```

```
Var Niz:Niz_Realnih_Brojeva;  
    Broj:Real;  
    I:Integer;
```

```
PROCEDURE Inicijalizacija_Niza;  
Begin  
  For I:=1 to Max do Niz[I]:=0;  
  I:=1;  
End;
```

```
PROCEDURE Unos_Brojeva;  
Var Ponavljanja,Brojac:Integer;  
Begin  
  Writeln('Koliko brojeva zelite?');  
  Readln(Ponavljanja);  
  Brojac:=0;  
  Repeat  
    Readln(Broj);  
    Inc(Brojac);  
  Until Brojac=Ponavljanja;  
End;
```

{Sa standardnog ulaza učitati skup realnih brojeva. Kraj unosa unesena vrijednost 0. U izlaznu datoteku izlaz.dat, upisati. - prvi poziciju i prvu minimalnu vrijednost učitanih brojeva, te broj pojavljivanja minimalne vrijednosti - prvu poziciju i prvu maksimalnu vrijednost učitanih brojeva, te broj pojavljivanja maksimalne vrijednosti Koristiti liste, procedure i/ili funkcije}

```
Type Realni_Broj=Real;  
    Lista=^Element;  
    Element=Record  
      Broj:Realni_Broj;  
      Sljedeci:Lista;  
    End;
```

```
Var Novi,Prvi,Zadnji:Lista;  
    Broj:Realni_Broj;  
    Min,Max:Realni_Broj;  
    Prvi_Min,Prvi_Max,Frek_Min,Frek_Max:Integer;
```

```
PROCEDURE Unos_Elementa_U_Listu;
```

```
FUNCTION Unesen_Broj:Boolean;  
Begin  
  Readln(Broj);  
  If Broj=0 then Unesen_Broj:=False else  
    Unesen_Broj:=True;  
End;
```

```
PROCEDURE Inicijalizacija_Liste;  
Begin  
  Novi:=NIL;  
  Prvi:=NIL;  
  Zadnji:=NIL;  
End;
```



```

PROCEDURE Unos_U_Listu(Novi:Lista);
Begin
  If Zadnji=NIL then
    Begin
      Novi^.Sljedeci:=NIL;
      Prvi:=Novi;
      Zadnji:=Novi;
    End
  Else
    Begin
      Novi^.Sljedeci:=NIL;
      Zadnji^.Sljedeci:=Novi;
      Zadnji:=Novi;
    End;
End;

Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
    Begin
      New(Novi);
      Novi^.Broj:=Broj;
      Unos_U_Listu(Novi);
    End;
End;

PROCEDURE MinX;
Begin
  Novi:=Prvi;
  Min:=Prvi^.Broj;
  While Novi<>NIL do
    Begin

      If Novi^.Broj<Min then Min:=Novi^.Broj;
      Novi:=Novi^.Sljedeci;
    End;

  End;

PROCEDURE MaxX;
Begin
  Novi:=Prvi;
  Max:=Prvi^.Broj;
  While Novi<>NIL do
    Begin
      If Novi^.Broj>Max then
        Max:=Novi^.Broj;
        Novi:=Novi^.Sljedeci;
      End;
    End;
End;

FUNCTION Prva_Pos(Br:Realni_Broj):Integer;
Var Nadjen:Boolean;
    Pos:Integer;
Begin
  Nadjen:=False;
  Novi:=Prvi;
  Pos:=0;
  While (Novi<>NIL) and (NOT Nadjen) do
    Begin
      If Br=Novi^.Broj then Nadjen:=True;
      Novi:=Novi^.Sljedeci;
      Inc(Pos);
    End;
End;

```

```

    Prva_Pos:=Pos;
End;

FUNCTION Frekfencija(Br:Realni_Broj):Integer;
Var Frek:Integer;
Begin
    Frek:=0;
    Novi:=Prvi;
    While Novi<>NIL do
        Begin
            If Br=Novi^.Broj then Inc(Frek);
            Novi:=Novi^.Sljedeci;
        End;
        Frekfencija:=Frek;
    End;

Begin
    Unos_Elemenata_U_Listu;
MinX;
    Writeln('min: ',min);
    writeln('prva pos: ',prva_pos(min));
    writeln('frek:',frekfencija(min));

End.

```

---

```

Uses CRT;
Const Ulazna_Datoteka1='ulaz1.dat';
      Ulazna_Datoteka2='ulaz2.dat';
      Max=1000;

Type Elemenat=Integer;
      Niz_Elemenata=Array[1..Max] of Elemenat;
      Datoteka=File of Elemenat;

      Var Dat1,Dat2:Datoteka;
          Niz1,Niz2:Niz_Elemenata;
          Br1,Br2,I,J:Integer;

PROCEDURE Upisivanje_Iz_Datoteke_U_Niz;

PROCEDURE Inicijalizacija_Niza;
Begin
    For I:= 1 to Max do Niz1[I]:=0;
    For I:= 1 to Max do Niz2[I]:=0;
    I:=1;J:=1;
    Br1:=0;Br2:=0;
End;

PROCEDURE Citanje_Iz_Dat;
Begin
    Assign(Dat1,Ulazna_Datoteka1);
    Reset(Dat1);
    While NOT Eof(Dat1) do
        Begin
            Read(Dat1,Niz1[I]);
            Inc(I);Inc(Br1);
        End;
    Close(Dat1);
    Assign(Dat2,Ulazna_Datoteka2);
    Reset(Dat2);
    While Not Eof(Dat2) do

```

```

Begin
  Read(Dat2,Niz2[J]);
  Inc(J);Inc(Br2);
End;
Close(Dat2);
End;

```

```

Begin
  Inicijalizacija_Niza;
  Citanje_Iz_Dat;
End;

```

```

PROCEDURE Sortiranje_Nizova(N:Niz_Elemenata;El:Integer);
Var POM:Elemenat;
Begin
  For I:= 1 to El-1 do
    Begin
      For J:=I+1 to El do
        Begin
          If N[I]<N[J] then
            Begin
              Pom:=N[I];
              N[I]:=N[J];
              N[J]:=Pom;
            End;
          End;
        End;
      End;
    End;
  End;
End;

```

```

PROCEDURE Presjek_Nizova;
Begin
  Upisivanje_Iz_Datoteke_U_Niz;
  Sortiranje_Nizova(Niz1,Br1);
  Sortiranje_Nizova(Niz2,Br2);
  For I:= 1 to Br1 do
    Begin
      For J:= 1 to Br2 do
        Begin
          If Niz1[I]=Niz2[J] then Writeln(Niz1[I]);
        End;
      End;
    End;
  End;
Begin
  Clrscr;
  Presjek_Nizova;
  Readln;
End.

```

```

Const Ulazna_Datoteka1='ulaz1.dat';
      Ulazna_Datoteka2='ulaz2.dat';
      Max=1000;

```

```

Type Elemenat=Integer;
      Niz_Elemenata=Array[1..Max] of Elemenat;
      Datoteka=File of Elemenat;

```

```

Var Dat1,Dat2:Datoteka;
      Niz1,Niz2:Niz_Elemenata;
      Br1,Br2,I,J:Integer;
      c:char;
      Broj:integer;
PROCEDURE Upis_U_Datoteku_Ulaz1;

```

```
PROCEDURE Inicijalizacija_Datoteke1;  
Begin  
  Assign(Dat1,Ulazna_Datoteka1);  
  Rewrite(Dat1);  
End;
```

```
PROCEDURE Inicijalizacija_Datoteke2;  
Begin  
  Assign(Dat2,Ulazna_Datoteka2);  
  Rewrite(Dat2);  
End;
```

```
FUNCTION Unos:Boolean;  
Begin  
  Writeln('Jos(D/N)');  
  Readln(C);  
  If Upcase(C)='N' then Unos:=False else  
    Unos:=True;  
End;
```

```
Begin  
  Inicijalizacija_Datoteke1;  
  Inicijalizacija_Datoteke2;  
  Repeat  
    Writeln('Datoteka 1');  
    Writeln('INT: ');  
    Readln(Broj);  
    Write(Dat1,Broj);  
  Until NOT Unos;  
  Close(Dat1);  
  Repeat  
    Writeln('Datoteka 2');  
    Writeln('INT: ');  
    Readln(Broj);  
    Write(Dat2,Broj);  
  Until NOT Unos;  
  Close(Dat2);
```

```
End;
```

```
PROCEDURE Citanje_Iz_Datoteke;
```

```
PROCEDURE Inicijalizacija_Datoteke1;  
Begin  
  Assign(Dat1,Ulazna_Datoteka1);  
  Reset(Dat1);  
End;
```

```
PROCEDURE Inicijalizacija_Datoteke2;  
Begin  
  Assign(Dat2,Ulazna_Datoteka2);  
  Reset(Dat2);  
End;
```

```
Var BR:Integer;  
Begin  
  Inicijalizacija_Datoteke1;  
  Inicijalizacija_Datoteke2;  
  While NOT Eof(Dat1) do  
    Begin  
      Writeln('Dat1: ');  
      Read(Dat1,Br);  
      Writeln(Br);
```

```

End;
Close(Dat1);
While NOT Eof(Dat2) do
Begin
  Writeln('Dat2: ');
  Read(Dat2,Br);
  Writeln(Br);
End;
Close(Dat2)
End;

```

```

Begin
  Upis_U_Datoteku_Ulaz1;
  Citanje_Iz_Datoteke;

```

```

End.

```

{Sa standardnog ulaza učitati skup realnih brojeva, kraj unosa učitana vrijednost 0. Iz skupa formirati sortiranu listu brojeva, koji su djeljivi sa 2 i djeljivi sa 3. Datu listu upisati u datoteku izlaz.dat. Koristiti liste, procedure i/ili funkcije}

```

Uses Crt;
Type Realni_Broj=Integer;
  Lista=^Elemenat;
  Elemenat=Record
    Broj:Realni_Broj;
    Sljedeci:Lista;
  End;

```

```

Var Novi,Prvi,Zadnji:Lista;
    Broj:Realni_Broj;

```

```

PROCEDURE Unos_Elemenata_U_Listu;

```

```

FUNCTION Unesen_Broj:Boolean;
Begin
  Readln(Broj);
  If Broj=0 then Unesen_Broj:=False else
    Unesen_Broj:=True;
End;

```

```

PROCEDURE Inicijalizacija_Liste;
Begin
  Novi:=NIL;
  Prvi:=NIL;
  Zadnji:=NIL;
End;

```

```

PROCEDURE Unos_U_Listu(Novi:Lista);
Begin
  If Zadnji=NIL then
  Begin
    Novi^.Sljedeci:=NIL;
    Prvi:=Novi;
    Zadnji:=Novi;
  End
  Else
  Begin
    Novi^.Sljedeci:=NIL;
    Zadnji^.Sljedeci:=Novi;
    Zadnji:=Novi;
  End;
End;

```

```

Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
    Begin
      New(Novi);
      Novi^.Broj:=Broj;
      Unos_U_Listu(Novi);
    End;

  End;

PROCEDURE Sort;
Var TMP:Lista;
    Pom:Integer;
Begin
  TMP:=Prvi^.Sljedeci;
  While TMP<>NIL do
    Begin
      Novi:=Prvi;
      While Novi<>NIL do
        Begin
          If Novi^.Broj<Tmp^.Broj then
            Begin
              Pom:=novi^.broj;
              Novi^.broj:=tmp^.broj;
              Tmp^.broj:=pom;
            End;
          Novi:=Novi^.Sljedeci;
        End;
      TMP:=TMP^.Sljedeci;
    End;
  End;

PROCEDURE Ispis_Liste_U_Dat;

FUNCTION Prirodan_Broj(Br:Integer):Boolean;
Var K:Integer;
Begin
  Prirodan_Broj:=True;
  For K:= 2 to Br-1 do
    Begin
      If Br mod K=0 then Prirodan_Broj:=False;
    End;
  End;

FUNCTION Ponavlja:Boolean;
Begin
  If Novi^.Broj=Novi^.SLjedeci^.Broj then Ponavlja:=True else
    Ponavlja:=False;
  End;

Begin
  Novi:=Prvi;
  While Novi<>NIL do
    BBegin
      if (Prirodan_Broj(Novi^.Broj) And (Not Ponavlja)) then
        Writeln(Novi^.Broj);
        Novi:=novi^.sljedeci;
      End;
    End;

Begin
  Clrscr;

```

```
Unos_Elemenata_U_Listu;  
Sort;  
Ispis_liste_U_Dat;  
End.
```

{Sa standardnog ulaza učitati skup realnih brojeva, kraj unosa učitana vrijednost 0. Iz skupa formirati sortiranu listu brojeva, koji su djeljivi sa 2 i djeljivi sa 3. Datu listu upisati u datoteku izlaz.dat. Koristiti liste, procedure i/ili funkcije}

```
Uses Crt;  
Type Realni_Broj=Integer;  
  Lista=^Elemenat;  
  Elemenat=Record  
    Broj:Realni_Broj;  
    Sljedeci:Lista;  
  End;  
End;
```

```
Var Novi,Prvi,Zadnji:Lista;  
    Broj:Realni_Broj;
```

```
PROCEDURE Unos_Elemenata_U_Listu;
```

```
FUNCTION Unesen_Broj:Boolean;  
Begin  
  Repeat  
    Broj:=Broj+1;  
    If Broj=0 then Unesen_Broj:=False else  
      Unesen_Broj:=True;  
  Until Broj=36000;  
End;
```

```
PROCEDURE Inicijalizacija_Liste;
```

```
Begin  
  Novi:=NIL;  
  Prvi:=NIL;  
  Zadnji:=NIL;  
End;
```

```
PROCEDURE Unos_U_Listu(Novi:Lista);
```

```
Begin  
  If Zadnji=NIL then  
    Begin  
      Novi^.Sljedeci:=NIL;  
      Prvi:=Novi;  
      Zadnji:=Novi;  
    End  
  Else  
    Begin  
      Novi^.Sljedeci:=NIL;  
      Zadnji^.Sljedeci:=Novi;  
      Zadnji:=Novi;  
    End;  
End;
```

```
Begin  
  Inicijalizacija_Liste;  
  While Unesen_Broj do  
    Begin  
      New(Novi);  
      Novi^.Broj:=Broj;  
      Unos_U_Listu(Novi);  
    End;
```

```
End;
```

```

PROCEDURE Sort;
Var TMP:Lista;
    Pom:Integer;
Begin
TMP:=Prvi^.Sljedeci;
While TMP<>NIL do
Begin
Novi:=Prvi;
While Novi<>NIL do
Begin
If Novi^.Broj<Tmp^.Broj then
Begin
Pom:=novi^.broj;
Novi^.broj:=tmp^.broj;
Tmp^.broj:=pom;
End;
Novi:=Novi^.Sljedeci;
End;
TMP:=TMP^.Sljedeci;
End;
End;

PROCEDURE Ispis_Liste_U_Dat;

FUNCTION Prirodan_Broj(Br:Integer):Boolean;
Var K:Integer;
Begin
Prirodan_Broj:=True;
For K:= 2 to Br-1 do
Begin
If Br mod K=0 then Prirodan_Broj:=False;
End;
End;

FUNCTION Ponavlja:Boolean;
Begin
If Novi^.Broj=Novi^.SLjedeci^.Broj then Ponavlja:=True else
Ponavlja:=False;
End;

Begin
Novi:=Prvi;
While Novi<>NIL do
Begin
if (Prirodan_Broj(Novi^.Broj) And (Not Ponavlja)) then
Writeln(Novi^.Broj);
Novi:=novi^.sljedeci;
End;
End;

Begin
Clrscr;
Unos_Elemenata_U_Listu;
Sort;
Ispis_liste_U_Dat;
End.

```



{Sa standardnog ulaza ucitati niz cijelih brojeva vecih od nula u listu. Kraj unosa - unesena vrijednost 0. U datoteku izlaz.dat odstampati listu sortiranu od manjeg ka vecem, sa svim brojevima koji se visestruko ponavljaju. Koristiti liste, procedure i/ili funkcije.}

```
Const Izlazna_Datoteka='C:\Izlaz.dat';
```

```
Type Cjeli_Broj=Integer;
```

```
    Lista=^Elemenat;  
    Elemenat=Record  
    Broj:Cjeli_Broj;  
    Sljedeci:Lista;  
End;
```

```
Var Novi,Prvi,Zadnji:Lista;  
    Broj:Cjeli_Broj;  
    Dat:Text;
```

```
PROCEDURE Unos_Elemenata_U_Listu;
```

```
FUNCTION Unesen_Broj:Boolean;  
Begin  
    Readln(Broj);  
    If Broj=0 then Unesen_Broj:=False else  
        Unesen_Broj:=True;  
End;
```

```
PROCEDURE Inicijalizacija_Liste;
```

```
Begin  
    Prvi:=Nil;  
    Zadnji:=Nil;  
    Novi:=Nil;  
End;
```

```
PROCEDURE Ubacivanje_Elementa_U_Listu(Novi:Lista);
```

```
Begin  
    If Zadnji=NIL then  
        Begin  
            Novi^.Sljedeci:=Nil;  
            Prvi:=Novi;  
            Zadnji:=Novi;  
        End  
    Else  
        Begin  
            Zadnji^.Sljedeci:=Novi;  
            Novi^.Sljedeci:=Nil;  
            Zadnji:=Novi;  
        End;  
End;
```

```
Begin  
    Inicijalizacija_Liste;  
    While Unesen_Broj do  
        Begin  
            New(Novi);  
            Novi^.Broj:=Broj;  
            Ubacivanje_Elementa_U_Listu(Novi);  
        End;  
End;
```

```
PROCEDURE Sortiranje_Liste;
```

```
Var Pom:Cjeli_Broj;  
    TMP:Lista;
```

```

Begin
TMP:=Prvi^.Sljedeci;
While TMP<>NIL do
Begin
Novi:=Prvi;
While Novi<>NIL do
Begin
If Novi^.Broj<TMP^.Broj then
Begin
POM:=Tmp^.Broj;
TMP^.Broj:=Novi^.Broj;
Novi^.Broj:=POM;
End;
Novi:=Novi^.Sljedeci;
End;
TMP:=TMP^.Sljedeci;
End;
End;

PROCEDURE Ispis_Liste_U_Datoteku;
Begin
Assign(Dat,Izlazna_Datoteka);
Rewrite(Dat);
Novi:=Prvi;
While Novi<>NIL do
Begin
Writeln(Dat,Novi^.Broj);
Novi:=Novi^.Sljedeci;
End;
Close(Dat);
End;
Begin
Unos_Elemenata_U_Listu;
Sortiranje_Liste;
Ispis_Liste_U_Datoteku;
End.

```

{Zadatak sa ispitnog roka: 27.06.2001;Grupa B Sa standardnog ulaza učitati skup realnih brojeva, kraj unosa vrijednost 0! Na standardni izlaz odstampati sortiran niz, izbacujući višestruko unjete vrijednosti, srednju vrijednost i sumu datog niza. Koristiti liste, procedure i/ili funkcije}

```

Type Realni_Broj=Real;
Lista=^Elemenat;
Elemenat=Record
Broj:Realni_Broj;
Sljedeci:Lista;
End;

Var Novi,Zadnji,Prvi:Lista;
Broj:Realni_Broj;
Suma:Real;
Elemenata:Integer;

PROCEDURE Unos_Elemenata_U_Listu;

FUNCTION Unesen_Broj:Boolean;
Begin
Readln(Broj);
If Broj=0 then Unesen_Broj:=False else
Unesen_Broj:=True;
End;

PROCEDURE Inicijalizacija_Liste;

```

```

Begin
  Prvi:=Nil;
  Zadnji:=Nil;
  Novi:=Nil;
  Elemenata:=0;
  Suma:=0;
End;

PROCEDURE Ubacivanje_Elemenata_U_Listu(Novi:Lista);
Begin
  If Zadnji=Nil then
    Begin
      Novi^.Sljedeci:=Nil;
      Zadnji:=Novi;
      Prvi:=Novi;
    End
  Else
    Begin
      Novi^.Sljedeci:=Nil;
      Zadnji^.Sljedeci:=Novi;
      Zadnji:=Novi;
    End;
  End;
Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
    Begin
      New(Novi);
      Inc(Elemenata);
      Novi^.Broj:=Broj;
      Suma:=Suma+Novi^.Broj;
      Ubacivanje_Elemenata_U_Listu(Novi);
    End;
  End;

PROCEDURE Sortiranje_Liste;
Var TMP:Lista;
    Pom:Realni_Broj;
Begin
  TMP:=Prvi^.Sljedeci;
  While TMP<>NIL do
    Begin
      Novi:=Prvi;
      While Novi<>NIL do
        Begin
          If Novi^.Broj>TMP^.Broj then
            Begin
              POM:=TMP^.Broj;
              TMP^.Broj:=Novi^.Broj;
              Novi^.Broj:=POM;
            End;
          Novi:=Novi^.Sljedeci;
        End;
      TMP:=TMP^.Sljedeci;
    End;
  End;

PROCEDURE Ispis_Liste_Na_Ekran;
FUNCTION Ponavljanje:Boolean;
Begin
  If Novi^.Sljedeci^.Broj=Novi^.Broj then
    Ponavljanje:=True else Ponavljanje:=False;
End;

```

```

Begin
  Novi:=Prvi;
  While Novi<>NIL do
    Begin
      If NOT ponavljanje then Writeln(Novi^.Broj:2:2);
      Novi:=Novi^.Sljedeci;
    End;
    Writeln('SUMA ELEMENAT LISTE JE:',Suma:2:2);
    Writeln('SREDNJA VRIJEDNOST:',Suma/Elemenata);

```

```
End;
```

```

Begin
  Unos_Elemenata_U_Listu;
  Sortiranje_Liste;
  Ispis_Liste_Na_Ekran;

```

```
End.
```

{Sa standardnog ulaza učitati skup realnih brojeva. Kraj unosa unesena vrijednost 0. U izlaznu datoteku ispisati unesene brojeve sortirane po frekfenciji ponavljanja. Koristiti nizove! max=1000 razlicitih brojeva. Procedure ili/i Funkcije}

```

Const Max_Elemenata=1000;
      Izlazna_Datoteka='C:\Izlaz.dat';

```

```

Type Realni_Broj=Integer;
      Elemenat=Record
        Broj:Realni_Broj;
        Frekfencija:Integer;
      End;
      Niz_Elemenata=Array[1..Max_Elemenata] of Elemenat;

```

```

Var Niz:Niz_Elemenata;
      Broj,I,J,Elemenata:Integer;

```

```
PROCEDURE Unos_Elemenata_U_Niz;
```

```
PROCEDURE Inicijalizacija_Niza;
```

```

Begin
  For I:=1 to Max_Elemenata do
    Begin
      Niz[I].Broj:=0;
      Niz[I].Frekfencija:=0;
    End;
    Elemenata:=0;
    I:=1;
  End;

```

```
PROCEDURE Unos_Elemenata;
```

```

Begin
  Niz[I].Broj:=Broj;
  Inc(Elemenata);
  Inc(I);
End;

```

```
FUNCTION Unesen_Broj:Boolean;
```

```

Begin
  Readln(Broj);
  Unesen_Broj:=True;
  If Broj=0 then Unesen_Broj:=False;
End;

```

PROCEDURE Ubacivanje\_Elemenata;

Begin

Inicijalizacija\_Niza;

While Unesen\_Broj do

Begin

Unos\_Elemenata;

End;

End;

Begin {Glavna procedura}

Ubacivanje\_Elemenata;

End;

PROCEDURE Frekfencija\_Elemenata\_Niza;

Var Nadjen:Boolean;

Begin

Nadjen:=False;

For I:=1 to Elemenata do

Begin

For J:=1 to Elemenata do

Begin

If Niz[I].Broj=Niz[J].Broj then

Begin

Niz[I].Frekfencija:=Niz[I].Frekfencija+1;

Nadjen:=True;

End;

End;

If Not Nadjen then Niz[I].Frekfencija:=1;

End;

End;

PROCEDURE Sortiranje\_Po\_Vrijednosti;

Var Pom:Elemenat;

Begin

For I:=1 To Elemenata do

Begin

For J:=1 to Elemenata do

Begin

If Niz[I].Broj>Niz[J].Broj then

Begin

Pom:=Niz[I];

Niz[I]:=Niz[J];

Niz[J]:=Pom;

End;

End;

End;

End;

PROCEDURE Sort\_Po\_Frekfenciji;

Var Pom:Elemenat;

Begin

For I:=1 to Elemenata-1 do

Begin

For J:=I+1 to Elemenata do

Begin

If Niz[I].Frekfencija>Niz[J].Frekfencija then

Begin

Pom:=Niz[I];

Niz[I]:=Niz[J];

Niz[J]:=Pom;

End;

End;

End;

```

End;

PROCEDURE Ispis;

FUNCTION Ponavlja_Se:Boolean;
Begin
  If Niz[I].Broj=Niz[I+1].Broj then Ponavlja_Se:=True
  Else Ponavlja_Se:=False;
End;

Begin
  For I:=1 to Elemenata do
    Begin
      If NOT Ponavlja_Se then Writeln(Niz[I].Broj,'f',Niz[I].Frekfencija);
    End;
  End;

  Begin
    Unos_Elemenata_U_Niz;
    Frekfencija_Elemenata_Niza;
    Sortiranje_Po_Vrijednosti;
    Sort_Po_Frekfenciji;
    Ispis;
  End.

Program Frekfencija_Liste;
Type Realni_Broj=Real;
  Lista=^Elemenat;
  Elemenat=Record
    Broj:Realni_Broj;
    Frekfencija:Integer;
    Sljedeci:Lista;
  End;

Var Novi,Zadnji,Prvi:Lista;
  Broj:Realni_Broj;
  Elemenata:Integer;

PROCEDURE Unos_Elemenata_U_Listu;

FUNCTION Unesen_Broj:Boolean;
Begin
  Readln(Broj);
  If Broj=0 then Unesen_Broj:=False else
  Unesen_Broj:=True;
End;

PROCEDURE Inicijalizacija_Liste;
Begin
  Prvi:=Nil;
  Zadnji:=Nil;
  Novi:=Nil;
  Elemenata:=0;
End;

PROCEDURE Ubacivanje_Elemenata_U_Listu(Novi:Lista);
Begin
  If Zadnji=Nil then
    Begin
      Novi^.Sljedeci:=Nil;
      Zadnji:=Novi;
      Prvi:=Novi;
    End

```

```

Else
Begin
  Novi^.Sljedeci:=Nil;
  Zadnji^.Sljedeci:=Novi;
  Zadnji:=Novi;
End;
End;
Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
  Begin
    New(Novi);
    Inc(Elemenata);
    Novi^.Broj:=Broj;
    Novi^.Frekfencija:=0;
    Ubacivanje_Elemenata_U_Listu(Novi);
  End;
End;
PROCEDURE Frekfencija_Elemenata_Liste;
Var TMP:Lista;
Begin
  TMP:=Prvi;
  While TMP<>NIL do
  Begin
    Novi:=Prvi;
    While Novi<>NIL do
    Begin
      IF Novi^.Broj=TMP^.Broj then
        Novi^.Frekfencija:=Novi^.Frekfencija+1;
      Novi:=Novi^.Sljedeci;
    End;
    TMP:=TMP^.Sljedeci;
  End;
End;

PROCEDURE Sortiranje_Liste;
Var TMP:Lista;
    Pom:Elementat;
Begin
  TMP:=Prvi^.Sljedeci;
  While TMP<>NIL do
  Begin
    Novi:=Prvi;
    While Novi<>NIL do
    Begin
      If Novi^.Broj>TMP^.Broj then
      Begin
        POM:=TMP^;
        TMP^:=Novi^;
        Novi^:=POM;
      End;
      Novi:=Novi^.Sljedeci;
    End;
    TMP:=TMP^.Sljedeci;
  End;
End;

PROCEDURE Ispis_Liste_Na_Ekran;

FUNCTION Ponavljanje:Boolean;
Begin
  If Novi^.Sljedeci^.Broj=Novi^.Broj then
    Ponavljanje:=True else Ponavljanje:=False;

```

```

End;

Begin
  Novi:=Prvi;
  While Novi<>NIL do
    Begin
      Writeln(Novi^.Broj:2:2,' F',Novi^.Frekfencija);
      Novi:=Novi^.Sljedeci;
    End;
  End;

End;
Begin
  Unos_Elemenata_U_Listu;
  Frekfencija_Elemenata_Liste;
  Sortiranje_Liste;
  Ispis_Liste_Na_Ekran;
End.

```

---

```

Program Frekfencija_Liste;
Type Realni_Broj=Real;
  Lista=^Elemenat;
  Elemenat=Record
    Broj:Realni_Broj;
    Frekfencija:Integer;
    Sljedeci:Lista;
  End;

Var Novi,Zadnji,Prvi:Lista;
    Broj:Realni_Broj;
    Elemenata:Integer;

PROCEDURE Unos_Elemenata_U_Listu;

FUNCTION Unesen_Broj:Boolean;
Begin
  Readln(Broj);
  If Broj=0 then Unesen_Broj:=False else
    Unesen_Broj:=True;
End;

PROCEDURE Inicijalizacija_Liste;
Begin
  Prvi:=Nil;
  Zadnji:=Nil;
  Novi:=Nil;
  Elemenata:=0;
End;

PROCEDURE Ubacivanje_Elemenata_U_Listu(Novi:Lista);
Begin
  If Zadnji=Nil then
    Begin
      Novi^.Sljedeci:=Nil;
      Zadnji:=Novi;
      Prvi:=Novi;
    End
  Else
    Begin
      Novi^.Sljedeci:=Nil;
      Zadnji^.Sljedeci:=Novi;
      Zadnji:=Novi;
    End;
End;

```



```

End;
Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
    Begin
      New(Novi);
      Inc(Elemenata);
      Novi^.Broj:=Broj;
      Novi^.Frekfencija:=0;
      Ubacivanje_Elemenata_U_Listu(Novi);
    End;
  End;
PROCEDURE Frekfencija_Elemenata_Liste;
Var TMP:Lista;
Begin
  TMP:=Prvi;
  While TMP<>NIL do
    Begin
      Novi:=Prvi;
      While Novi<>NIL do
        Begin
          IF Novi^.Broj=Tmp^.Broj then
            Novi^.Frekfencija:=Novi^.Frekfencija+1;
            Novi:=Novi^.Sljedeci;
          End;
          Tmp:=Tmp^.Sljedeci;
        End;
      End;
    End;
  End;

PROCEDURE Sortiranje_Liste;
Var TMP:Lista;
    Pom:Elemenat;
Begin
  TMP:=Prvi^.Sljedeci;
  While TMP<>NIL do
    Begin
      Novi:=Prvi;
      While Novi<>NIL do
        Begin
          If Novi^.Broj>TMP^.Broj then
            Begin
              POM:=TMP^;
              TMP^:=Novi^;
              Novi^:=POM;
            End;
            Novi:=Novi^.Sljedeci;
          End;
          TMP:=TMP^.Sljedeci;
        End;
      End;
    End;
  End;

PROCEDURE Ispis_Liste_Na_Ekran;

FUNCTION Ponavljanje:Boolean;
Begin
  If Novi^.Sljedeci^.Broj=Novi^.Broj then
    Ponavljanje:=True else Ponavljanje:=False;
End;

Begin
  Novi:=Prvi;
  While Novi<>NIL do
    Begin

```

```

    Writeln(Novi^.Broj:2:2,' F',Novi^.Frekfencija);
    Novi:=Novi^.Sljedeci;
End;

```

```

End;
Begin
    Unos_Elemenata_U_Listu;
    Frekfencija_Elemenata_Liste;
    Sortiranje_Liste;
    Ispis_Liste_Na_Ekran;
End.

```

{ZAD 5: Sa standardnog ulaza ucitati skup realnih brojeva. Kraj unosa - unesena vrijednost 0. U izlaznu datoteku izlaz.dat upisati: - prvu poziciju i prvu minimalnu vrijednost ucitanih brojeva, te broj pojavljivanja minimalne vrijednosti. - prvu poziciju i prvu maksimalnu vrijednost ucitanih brojeva, te broj pojavljivanja minimalne vrijednosti. }

```

Const Max_Elemenata=200;
    Izlazna_Datoteka='C:\Izlaz.dat' ;
Type Realni_Broj=Real;
    Niz_Elemenata=Array[1..Max_Elemenata] of Realni_Broj;

```

```

Var Niz:Niz_Elemenata;
    Broj:Realni_Broj;
    Elemenata_Niza,I:Integer;
    Minimalni_Elemenat,Maksimalni_Elemenat:Realni_Broj;
    Pozicija_Minimalnog,Pozicija_Maksimalnog:Integer;
    Dat:Text;
PROCEDURE Unos_Elemenata_U_Niz;

```

```

FUNCTION Unesen_Broj:Boolean;
Begin
    Readln(Broj);
    If Broj=0 then Unesen_Broj:=False else
        Unesen_Broj:=True;
End;

```

```

PROCEDURE Inicijalizacija_Niza;
Begin
    Elemenata_Niza:=0;
    For I:=1 to Max_Elemenata do Niz[I]:=0;
    I:=1;
End;

```

```

PROCEDURE Unos_Elemenata(Broj:Realni_Broj);
Begin
    Niz[I]:=Broj;
    Inc(I) ;
    Inc(Elemenata_Niza);
End;

```

```

Begin
    Inicijalizacija_Niza;
    While Unesen_Broj do
        Begin
            Unos_Elemenata(Broj);
        End;
End;

```

```

PROCEDURE Min(Niz_1:Niz_Elemenata);
Var Brojac:Integer;
Begin
    Minimalni_Elemenat:=Niz_1[1];
    Pozicija_Minimalnog:=1;

```

```

For Brojac:=2 to Elemenata_Niza do
Begin
If Niz_1[Brojac]<Minimalni_Elemenat then
Begin
Minimalni_Elemenat:=Niz_1[Brojac];
Pozicija_Minimalnog:=Brojac;
End;
End;
End;

```

```

PROCEDURE Max(Niz_1:Niz_Elemenata);
Var Brojac:Integer;
Begin
Maksimalni_Elemenat:=Niz_1[1];
Pozicija_Maksimalnog:=1;
For Brojac:=2 to Elemenata_Niza do
Begin
If Niz_1[Brojac]>Maksimalni_Elemenat then
Begin
Maksimalni_Elemenat:=Niz_1[Brojac];
Pozicija_Maksimalnog:=Brojac;
End;
End;
End;

```

```

FUNCTION Frekfencija(Broj:Realni_Broj):Integer;
Var Brojac,Frek:Integer;
Begin
Frek:=0;
For Brojac:=1 to Elemenata_Niza do
Begin
If Broj=Niz[Brojac] then Inc(Frek) ;
End;
Frekfencija:=Frek;
End;

```

```

PROCEDURE Ispis_Vrijednosti_U_Datoteku;
Begin
Assign(Dat,Izlazna_Datoteka);
Rewrite(Dat);
Writeln(Dat,'Max: ',Maksimalni_Elemenat,' ', Pozicija_Maksimalnog);
Writeln(Dat,'MIN: ',Minimalni_Elemenat,' ', Pozicija_Minimalnog);
Writeln(Dat,'Frekfencija Maksimalnog: ',Frekfencija(Maksimalni_Elemenat));
Writeln(Dat,'Frekfencija Minimalnog: ',Frekfencija(Minimalni_Elemenat));
Close(Dat);
End;

```

```

Begin
unos_elemenata_u_niz;
Max(Niz);
Min(Niz);
ispis_vrijednosti_u_datoteku;

```

End.

{Sa standardnog ulaza učitati skup realnih brojeva, kraj unosa učitana vrijednost 0. Iz skupa formirati sortiranu listu brojeva, koji su djeljivi sa 2 i djeljivi sa 3. Datu listu upisati u datoteku izlaz.dat. Koristiti liste, procedure i/ili funkcije}

```

Const Izlazna_Datoteka='C:\Izlaz.dat';

```

```

Type Realni_Broj=Integer;
Lista=^Elemenat;
Elemenat=Record

```

```

    Broj:Realni_Broj;
    Sljedeci:Lista;
End;

Var Novi,Prvi,Zadnji:Lista;
    Broj:Realni_Broj;
    Dat:Text;
PROCEDURE Unos_Elemenata_U_Listu;

FUNCTION Unesen_Broj:Boolean;
Begin
    Readln(Broj);
    If Broj=0 then Unesen_Broj:=False else
        Unesen_Broj:=True;
End;

PROCEDURE Inicijalizacija_Liste;
Begin
    Novi:=NIL;
    Prvi:=NIL;
    Zadnji:=NIL;
End;

PROCEDURE Unos_U_Listu(Novi:Lista);
Begin
    If Zadnji=NIL then
        Begin
            Novi^.Sljedeci:=NIL;
            Prvi:=Novi;
            Zadnji:=Novi;
        End
    Else
        Begin
            Novi^.Sljedeci:=NIL;
            Zadnji^.Sljedeci:=Novi;
            Zadnji:=Novi;
        End;
End;

Begin
    Inicijalizacija_Liste;
    While Unesen_Broj do
        Begin
            New(Novi);
            Novi^.Broj:=Broj;
            Unos_U_Listu(Novi);
        End;

End;

FUNCTION Djeljiv_Sa_2_I_3(Br:Realni_Broj):Boolean;
Begin
    If (((Br mod 2)=0) and ((Br mod 3)=0)) then
        Djeljiv_Sa_2_I_3:=True else Djeljiv_Sa_2_I_3:=False;
End;

PROCEDURE Ispis_Liste_U_Dat;
Begin
    Assign(Dat,Izlazna_datoteka);
    Rewrite(Dat);
    Novi:=Prvi;
    While Novi<>NIL do
        BBegin

```

```

    If Djeljiv_Sa_2_I_3(Novi^.Broj) then
        Writeln(Dat,Novi^.Broj);
        Novi:=novi^.sljedeci;
    End;
    Close(Dat);
End;
Begin
    Unos_Elemenata_U_Listu;
    Ispis_liste_U_Dat;
End.

```

{Sa standardnog ulaza ucitati skup realnih brojeva. Kraj unosa unesena vrijednost 0. U izlaznu datoteku izlaz.dat, upisati. - prvi poziciju i prvu minimalnu vrijednost ucitanih brojeva, te broj pojavljivanja minimalne vrijednsti - prvu poziciju i prvu maksimalnu vrijednost ucitanih brojeva, te broj pojavljivanja maksimalne vrijednosti Koristiti liste, procedure i/ili funkcije}

```

Type Realni_Broj=Real;
    Lista=^Elemenat;
    Elemenat=Record
        Broj:Realni_Broj;
        Sljedeci:Lista;
    End;

Var Novi,Prvi,Zadnji:Lista;
    Broj:Realni_Broj;
    Minimalni,Maximalni:Realni_Broj;
    Prvi_Min,Prvi_Max:Integer;

PROCEDURE Unos_Elemenata_U_Listu;

FUNCTION Unesen_Broj:Boolean;
Begin
    Readln(Broj);
    If Broj=0 then Unesen_Broj:=False else
        Unesen_Broj:=True;
End;

PROCEDURE Inicijalizacija_Liste;
Begin
    Novi:=NIL;
    Prvi:=NIL;
    Zadnji:=NIL;
End;

PROCEDURE Unos_U_Listu(Novi:Lista);
Begin
    If Zadnji=NIL then
        Begin
            Novi^.Sljedeci:=NIL;
            Prvi:=Novi;
            Zadnji:=Novi;
        End
    Else
        Begin
            Novi^.Sljedeci:=NIL;
            Zadnji^.Sljedeci:=Novi;
            Zadnji:=Novi;
        End;
End;

Begin
    Inicijalizacija_Liste;
    While Unesen_Broj do

```

```

Begin
  New(Novi);
  Novi^.Broj:=Broj;
  Unos_U_Listu(Novi);
End;
End;

PROCEDURE Min(Novi:Lista);
Var Nadjen:Boolean;
Begin
  Minimalni:=Prvi^.Broj;
  Prvi_Min:=0;
  Novi:=Prvi;
  While Novi<>NIL do
    Begin
      If Novi^.Broj<Minimalni then
        Minimalni:=Novi^.Broj;
      Novi:=Novi^.Sljedeci;
    End;
    Nadjen:=False;
    Novi:=Prvi;
    While (Novi<>NIL) and (Nadjen=False) do
      Begin
        If Prvi^.Broj=Minimalni then
          Nadjen:=True;
          Inc(Prvi_Min);
          Prvi:=Prvi^.Sljedeci;
        End; {Isto je i za maksimalni! - samo znak i promjenljive promjenim}
      End;
    End;

FUNCTION Frekfencija(Broj:Realni_Broj):Integer;
Var Frek:Integer;
Begin
  Novi:=Prvi;
  Frek:=1;
  While Novi<>NIL do
    Begin
      If Novi^.Broj=Broj then Frek:=Frek+1;
      Novi:=Novi^.Sljedeci;
    End;
  Frekfencija:=Frek;
End;

Begin
  Unos_Elemenata_U_Listu;
  Min(Novi);
  Writeln('Minimaln: ', Minimalni,'Prvi min: ', Prvi_Min);
  Writeln('Frekfencija Minimalnog',Frekfencija(Minimalni));
End.

```

---

```

Const Max_Elemenata=1000;

```

```

Type Elemenat=Record
  Broj:Integer;
  Frekfencija:Integer;
End;
Niz_Elemenata=Array[1..Max_Elemenata] of Elemenat;

```

```

Var Niz,Niz_2:Niz_Elemenata;
    Broj,Elemenata_Niza,I,J:Integer;

```

PROCEDURE Unos\_Elemenata\_U\_Niz;

FUNCTION Unesen\_Broj:Boolean;  
Begin  
  Readln(Broj);  
  If Broj=0 then Unesen\_Broj:=False else  
    Unesen\_Broj:=True;  
End;

PROCEDURE Inicijalizacija\_Niza;  
Begin  
  Elemenata\_Niza:=1;  
  For I:=1 to Max\_Elemenata do  
    Begin  
      Niz[I].Broj:=0;  
      Niz[I].Frekfencija:=0;  
    End;  
  For I:=1 to Max\_Elemenata do  
    Begin  
      Niz\_2[I].Broj:=0;  
      Niz\_2[I].Frekfencija:=0;  
    End;  
  I:=1;  
End;

PROCEDURE Unos\_Elemenata(Broj:Integer);  
Begin  
  Niz[I].Broj:=Broj;  
  Inc(I);  
  Inc(Elemenata\_Niza);  
End;

Begin  
  Inicijalizacija\_Niza;  
  While Unesen\_Broj do  
    Begin  
      Unos\_Elemenata(Broj);  
    End;  
End;

PROCEDURE Frekfencija;  
Var Nadjen:Boolean;  
Begin  
  For I:=1 to Elemenata\_Niza do  
    Begin  
      Nadjen:=False;  
      For J:=1 to Elemenata\_Niza do  
        Begin  
          If Niz[I].Broj=Niz[J].Broj then  
            Begin  
              Nadjen:=True;  
              Niz[I].Frekfencija:=Niz[I].Frekfencija+1;  
            End ;  
        End;  
      If Not Nadjen then Niz[I].Frekfencija:=1;  
    End;  
  End;

PROCEDURE Sortiranje\_Po\_Frekfenciji;  
Var Pom:Elemenat;  
Begin  
  For I:=1 to Elemenata\_Niza-1 do  
    Begin

```

For J:=I+1 to Elemenata_Niza do
Begin
If Niz[I].Frekfencija>Niz[J].Frekfencija then
Begin
Pom:=Niz[I];
Niz[I]:=Niz[J];
Niz[J]:=Pom;
End;
End;
End;
End;

```

```

PROCEDURE Sortiranje_Po_Vrijednosti;
Var Pom:Elemenat;
Begin
For I:=1 to Elemenata_Niza-1 do
Begin
For J:=I+1 to Elemenata_Niza do
Begin
If Niz[I].Broj>Niz[J].Broj then
Begin
Pom:=Niz[I];
Niz[I]:=Niz[J];
Niz[J]:=Pom;
End;
End;
End;
End;

```

```

PROCEDURE Ispis;

```

```

FUNCTION Ispisuj:Boolean;
Begin
If Niz[I].Broj=Niz[I+1].Broj then Ispisuj:=False
Else Ispisuj:=True;
End;

```

```

Begin
For I:=1 to Elemenata_Niza do
If Ispisuj then
Begin
Writeln('Br: ',Niz[I].Broj,'Frek:',Niz[I].Frekfencija);
End;
End;

```

```

Begin
Unos_Elemenata_U_Niz;
Frekfencija;
Sortiranje_Po_Vrijednosti;
Sortiranje_Po_Frekfenciji;
Ispis;
End.

```

---

```

program f;
uses crt;
const
max=200;
type niz_el=record
broj:integer;
frek:integer;
end;

```



```

var
brojac,broj,m,n : integer;
niz, frek_ele_u_nizu : array [1..max] of integer;
niz1:niz_el;
I,J:Integer;

procedure upis;
begin
brojac:=0;
repeat
begin
brojac:=brojac+1;
readln(broj);
niz[brojac]:=broj;
end;
until (broj=0) or (brojac=max);
end;

procedure frek;
var j: integer;

begin
for j:=1 to brojac do
begin
frek_ele_u_nizu[j]:=0;
end;
for n:=1 to brojac do
begin
for m:=1 to brojac do
begin
writeln(niz[n],' ',niz[m]);
if niz[n]=niz[m]
then
frek_ele_u_nizu[n]:=frek_ele_u_nizu[n]+1;
end;
end;
end;

begin

end.

```

---

```

Type Realni_Broj=Real;
  Lista=^Elemenat;
  Elemenat=Record
    Broj:Integer;
    Frek:Integer;
    Sljedeci:Lista;
  End;

```

```

Var Novi,Prvi,Zadnji:Lista;
    Broj:Integer;

```

```

PROCEDURE Unos_Elemenata_U_Listu;

```

```

FUNCTION Unesen_Broj:Boolean;
Begin
  Readln(Broj);
  If Broj=0 then Unesen_Broj:=False else
  Unesen_Broj:=True;

```

```

End;

PROCEDURE Inicijalizacija_Liste;
Begin
  Novi:=NIL;
  Prvi:=NIL;
  Zadnji:=NIL;
End;

PROCEDURE Unos_U_Listu(Novi:Lista);
Begin
  If Zadnji=NIL then
    Begin
      Novi^.Sljedeci:=NIL;
      Prvi:=Novi;
      Zadnji:=Novi;
    End
  Else
    Begin
      Novi^.Sljedeci:=NIL;
      Zadnji^.Sljedeci:=Novi;
      Zadnji:=Novi;
    End;
  End;

Begin
  Inicijalizacija_Liste;
  While Unesen_Broj do
    Begin
      New(Novi);
      Novi^.Broj:=Broj;
      Unos_U_Listu(Novi);
    End;
  End;
PROCEDURE Sort;
Var TMP:Lista;
    Pom:Integer;
Begin
  TMP:=Prvi^.Sljedeci;
  While TMP<>NIL do
    Begin
      Novi:=Prvi;
      While Novi<>NIL do
        Begin
          If Novi^.Broj>Tmp^.Broj then
            Begin
              Pom:=novi^.broj;
              Novi^.broj:=tmp^.broj;
              Tmp^.broj:=pom;
            End;
          Novi:=Novi^.Sljedeci;
        End;
      TMP:=TMP^.Sljedeci;
    End;
  End;

PROCEDURE Ispis;
Begin
  Novi:=Prvi;
  While Novi<>Nil do
    Begin
      Writeln(Novi^.Broj);
      Novi:=Novi^.Sljedeci;
    End;
  End;

```

```
End;  
Writeln('Svi ELeменти liste su ispisani');  
End;  
PROCEDURE Frekfencije;  
Begin  
  Novi:=Prvi;  
  While Novi<>NIL do  
    Begin  
  
      Novi:=Novi^.Sljedeci;  
    End;  
  End;  
Begin  
  Unos_Elemenata_U_Listu;  
  Sort;  
  Ispis;  
End.
```