



```

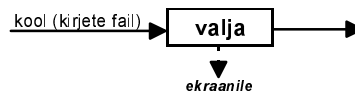
procedure sisesta(var kool:opfail; var tekst:text);
var i,j: integer;
    kirje: opilane;
    ch: char;
begin
    reset(tekst);
    rewrite(kool);
    while not eof(tekst) do
    with kirje do
        begin
            readln (tekst, eesnimi);
            readln (tekst, perenimi);
            readln (tekst, klass);
            for j:=1 to 5 do read (tekst, edukus[j]);
            write(kool,kirje);
        end;
    end;
end;

```

NB! var tekst:text

Kui protseduuri parameetrik on F,
siis on ta alati muutujaparameteer (var)

9.5 Kirjete F väljastamise protseduur (näide)



```

procedure valja (var f:opfail);
var i,j: integer;
    abi: opilane;
begin
    reset(f);
    writeln(' Nr Eesnimi Perenimi Klass Hinded');
    writeln('_____');
    i:=0;
    while not eof(f) do
    begin
        i:=i+1;
        read(f,abi); { loen kirjete F-st kirje abi-sse }
        with abi do
            begin
                write(i:3, ' ', eesnimi, perenimi, klass, ' ');
                for j:=1 to 5 do begin write(edukus[j]:2);
                writeln
            end
        end
    end;
end;

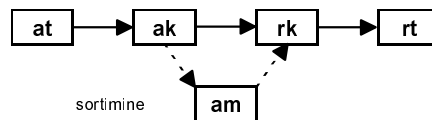
```

10. Kirjete F ülesandest

moodusta kirjete F, väljasta kirjete F, moodusta uus F, väljasta see F
algandmed kettal olevas tf-s, tulemused kettale tf-i
andmete liikumine

tähistame:

at - algandmed (tf)
rt - resultaadid (tf)
ak - algandmed (kirjete F)
rk - resultaadid (kirjete F)
am - algandmed (massiiv)



catpascal-is 2 näiteprogrammi kirjete F-de peale
kaupf.p, parimf.p

11. Hulgad (H)**11.1 Kirjeldamine**

senivaadeldud tüüpide korral sai mingit tüüpi elemendile omistada 1 väärtust;

praktilas tekib ülesandeid, kus tüüpi elemendid on väärtuste hulgad

Pascali H on lähedane matemaatikas kasutatavale

mõned ülesanded on H-dega lihtsamad lahendada

H-tüüp kirjeldatakse mingi ordinaaltüübi abil

Ordinaaltüüp on selline tüüp, mille elementide jaoks on määratud eelmine ja järgmine väärtus

Pascalis kõik skalaartüübid, v.a. *real*

integer korral täisarvude alamhulk, kuna max H elementide arv on piiratud (sõltub realisatsioonist)

ordinaaltüübi elementidele saab kasutada funktsioone SUCC(i) ja PRED(i)

type hulk = *set of* ordinaaltüüp;

var h: hulk;

või

var h: *set of* ordinaaltüüp;

näiteks:

type varvid = (punane, kollane, sinine); {loendustüübi kirjeldus}

segu = *set of* varvid; {hulgatüübi kirjeldus}

var varv: segu; {muutujakirjeldus}

var varv: *set of* (punane, kollane, sinine); {ühendatud kirjeldus}

type tahed = ('A'..'Z', 'a'..'z'); {loendustüübi kirjeldus}

tahehulk = *set of* tahed; {hulgatüübi kirjeldus}

var vokaalid, kaashaalikud, nimi: tahehulk;

c: char;

LDkevad2004 5.

H-tüübi elementide väärtusteks on hulgad
 [punane, kollane, sinine] [punane, sinine] [kollane]
 [punane..sinine]
 ['a','b','c','d'] ['i'..'n'] ['a','e','i','o','u']
 ['J','a','a','n'],
 [] - tühi hulk
 omistamine
 vokaalid := ['a','e','i','o','u']
 nimi := vokaalid
 varv := []
 varv := [kollane]
 kuna tüübid on rekursiivsed, siis näiteks võib
 var varvimassiiv: array [1..10] of sega;

11.2 Tehted hulkadega

5 võrdlus- ja 3 (hulga-) aritmeetilist tehet

võrdlustehted:

= <> <= >= in
 võrdlustehete tulemuseks on tõeväärtus, s.t. kas true või false
 kollane = [punane, sinine] false
 [punane] <> [sinine] true
 [] <= [sinine] true
 = ja <> kontrollivad H-de võrdsust
 <= ja >= kontrollivad H-de sisalduvust
 kuidas kontrollida > ja < ?
 in kontrollib, kas element kuulub hulka
 punane in [punane, sinine] true
 if sinine in varv then ...
 while 'a' in nimi do ...
 if c in vokaalid then ...

LDkevad2004 6.

hulgaaritmeetilised tehted:

+ ühend (disjunktsioon)
 * ühisosa (konjunktsioon)
 - vahe
 [punane, sinine] + [punane, kollane] = [punane, kollane, sinine]
 [punane, sinine] * [punane, kollane] = [punane]
 [punane, sinine] - [punane, kollane] = [sinine]

Tehete prioriteetid

- 0) ()
- 1) not
- 2) *, /, div, mod, and
- 3) +, -, or
- 4) =, <>, <=, >=, <, >, in

11.3 Näide H-de kasutamise kohta

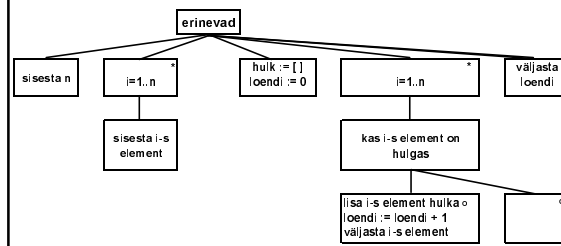
Ülesanne: sisesta täisarvude jada; on teada:

- 1) selles jadas on palju korduvaid arve
- 2) arvude väärtused on vahemikus on 0..255
- 3) selles jadas on maksimaalselt 500 arvu

leia, mitu erinevat arvu on selles jadas ja väljasta nad (s.t. erinevad arvud)

tähistused:

n - elementide arv jadas
 loendi - erinevate elementide arv
 hulk - hulk, kuhu paneme erinevad elemendid
 jada - massiivi identifikaator
 i - massiivi indeks



```

program erinevad;
var   hulk: set of 0..255;
      jada: array [1..500] of 0..255;
      loendi, i, n: 0..500;
begin
  writeln('sisesta jada liikmete arv');
  read(n);
  for i:=1 to n do read (jada[i]);
  hulk := [ ]; loendi := 0;
  for i:=1 to n do
    if not (jada[i] in hulk) then
      begin
        hulk := hulk + [jada[i]];
        loendi := loendi + 1;
        write (jada[i])
      end;
  writeln;
  writeln('loendi=', loendi:3)
end.

```