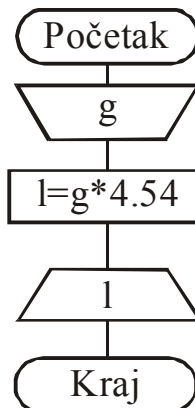

UVOD – RAD SA OSNOVNIM TIPOVIMA PODATAKA

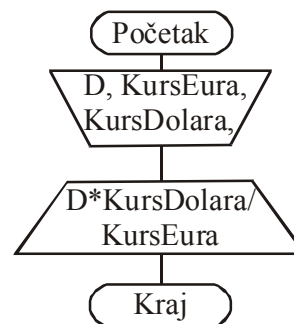
1. Napisati program na programskom jeziku PASKAL kojim se vrši prevododjenje kolicine tecnosti iz galona u litre, ako je 1 galon = 4.54 litra.

```
program IzGalonaULitre;
uses WinCrt;
var
  g, l : real;
begin
  write('Unesite kolicinu tecnosti u galonima: ');
  readln(g);
  l := g*4.54;
  writeln('Odgovarajuca kolicina u litrima je: ', l:8:2 );
end.
```



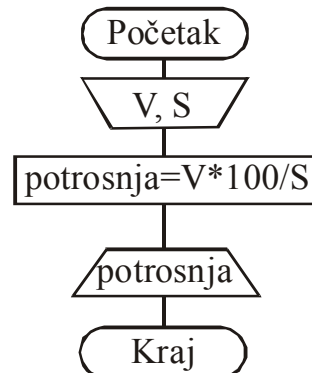
2. Napisati program na programskom jeziku PASKAL kojim se vrši preračunavanje americkih dolara u Euro, ako su poznate dinarske vrednosti ovih valuta.

```
program Konverzija;
uses WinCrt;
var
  D, KursDolara, KursEura : real;
begin
  write('Koliko dolara treba konvertovati u Eure: ');
  readln(D);
  write('Uneti kurs dolara izrazen u dinarima: ');
  readln(KursDolara);
  write('Uneti kurs Eura izrazen u dinarima: ');
  readln(KursEura);
  writeln(D:10:2, ' $ =', D*KursDolara/KursEura:8:2, ' Eura');
end.
```



3. Sa punim automobilskim rezervoarom kapaciteta V litara predjen je put od S kilometara. Napisati program kojim se racuna potrosnja automobila u litrima na 100 km.

```
program PotrosnjaGoriva;
uses WinCrt;
var
  V, S, potrosnja : real;
begin
  write('Unesite kapacitet rezervoara: ');
  readln(V);
  write('Unesite predjeni put: ');
  readln(S);
  potrosnja:=V*100/S;
  writeln('Potrosnja na 100 km je: ', potrosnja:6:2, ' l');
end.
```

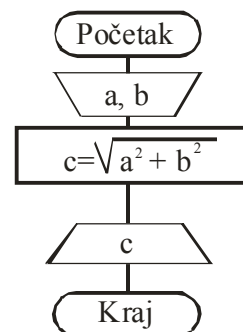


4. Napisati program na programskom jeziku PASKAL kojim se izracunava hipotenuza pravouglog trougla na osnovu zadatih kateta.

```

program Hipotenuza;
uses WinCrt;
var
  a, b, c : real;
begin
  write('Unesite duzinu kateta: ');
  readln(a, b);
  c:=sqrt(sqr(a)+sqr(b));
  writeln('Duzina hipotenuze je: ', c:6:2);
end.

```

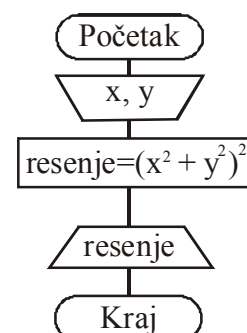


5. Napisati program koji resava sledeci matematički izraz: $(x^2 + y^2)^2$

```

program razlomak1;
uses WinCrt;
var
  x, y, resenje : real;
begin
  write('Unesite vrednosti promenljivih x i y: ');
  readln(x, y);
  resenje := sqr((sqr(x)+sqr(y)));
  writeln('Resenje razlomka je: ', resenje:6:2);
end.

```

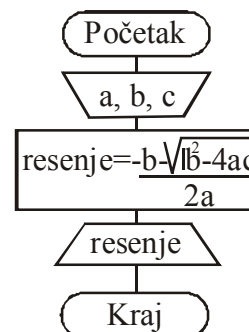


6. Napisati program koji izracunava izraz: $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$

```

program razlomak1;
uses WinCrt;
var
  a, b, c, resenje : real;
begin
  write('Unesite vrednosti parametara a, b i c: ');
  readln(a, b, c);
  resenje := (-b-sqrt(abs(sqr(b)-4*a*c)))/(2*a);
  writeln('Resenje zadataog razlomka je: ', resenje:8:2 );
end.

```



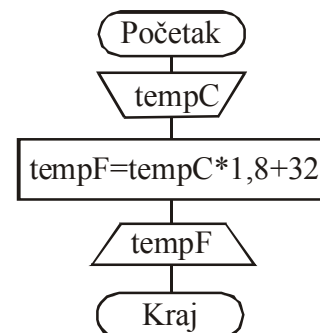
7. Napisati program kojim se prevodi temperatura iz skale Celzijusa u skalu Farenhajta, ako je odgovarajuca formula oblika:

Temperatura po Farenhajtu = (temperatura po Celzijusu)*1.80 + 32

```

program CelzijusFar;
uses WinCrt;
var

```



```
tempC,tempF : real;
```

```
begin  
  write('Unesite temperaturu u stepenima Celzijusa: ');  
  readln(tempC);  
  tempF := tempC*1.8+32;  
  writeln('Temperatura po Farenhajtu je: ', tempF:8:2 );  
end.
```

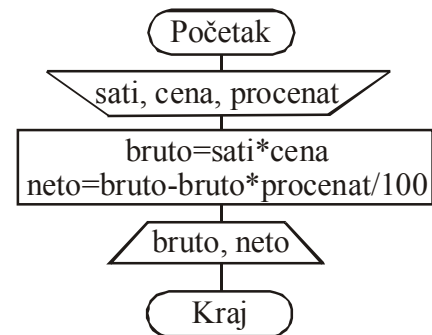
8. Za ucenike koji su zaposleni preko omladinske zadruga izracunati bruto i neto dohodak, ako je poznat broj radnih sati, cena po satu i procenat odbijanja na osnovu odredjenog doprinosa.

Izmedju poznatih i nepoznatih velicina postoje sledeci odnosi:

$$\text{bruto} = \text{sati} * \text{cena}$$

$$\text{neto} = \text{bruto} - \text{bruto} * \text{procenat} / 100$$

```
program dohodak;  
uses WinCrt;  
var  
  sati, cena, procenat, bruto : integer;  
  neto : real;  
begin  
  write('Unesite broj radnih sati: ');  
  readln(sati);  
  write('Unesite cenu po satu: ');  
  readln(cena);  
  write('Unesite procenat za odbijanje: ');  
  readln(procenat);  
  bruto := sati*cena;  
  neto := bruto - bruto*(procenat/100);  
  writeln('Bruto dohodak je: ', bruto );  
  writeln('Neto dohodak je: ', neto:8:2 );  
end.
```



USLOVNI IZRAZI

Jednostavni uslovni izraz: IF-THEN

IF-THEN izraz ima opšti oblik:

**IF uslov THEN
izraz;**

```
IF Age < 18 THEN  
WRITELN('Ovo je maloletnik.');
```

IF-THEN-ELSE izraz

Opsti oblik ovog izraza je:

IF uslov THEN

izraz1

ELSE

izraz2;

ELSE-IF

Ukoliko postoji višestruko granjanje pored koristi se sledeći izraz:

IF uslov1 THEN

izraz1

ELSE IF uslov2 THEN

izraz2

ELSE IF uslov3 THEN

izraz3

...

ELSE

izraz *n*;

10. Napisati program na programskom jeziku PASCAL kojim se promenljivoj max dodeljuje vrednost veceg izmedju brojeva a i b.

```
program Max2;
```

```
uses WinCrt;
```

```
var
```

```
  a, b, max : integer;
```

```
begin
```

```
  write('Unesite dva cela broja: ');
```

```
  readln(a, b);
```

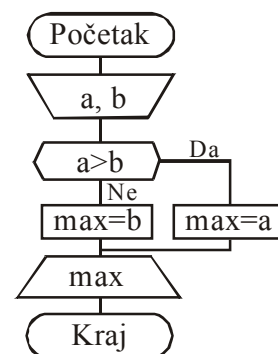
```
  if a>b
```

```
    then max:=a
```

```
    else max:=b;
```

```
  writeln('Veci broj je: ', max);
```

```
end.
```



11. Napisati program na programskom jeziku PASCAL kojim se promenljivoj max dodeljuje najveća vrednost od tri zadata cela broja.

```
program Max3;
```

```
uses WinCrt;
```

```
var
```

```
  a, b, c, max : integer;
```

```
begin
```

```
  write('Unesite tri cela broja: ');
```

```
  readln(a, b, c);
```

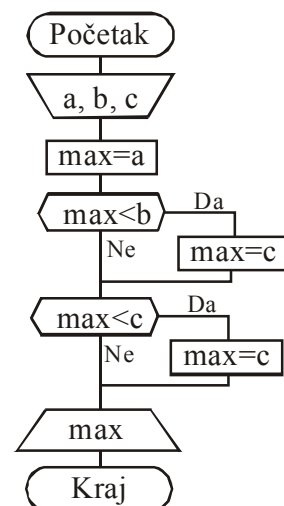
```
  max:=a;
```

```
  if max<b then max:=b;
```

```
  if max<c then max:=c;
```

```
  writeln('Najveci broj je: ', max);
```

```
end.
```



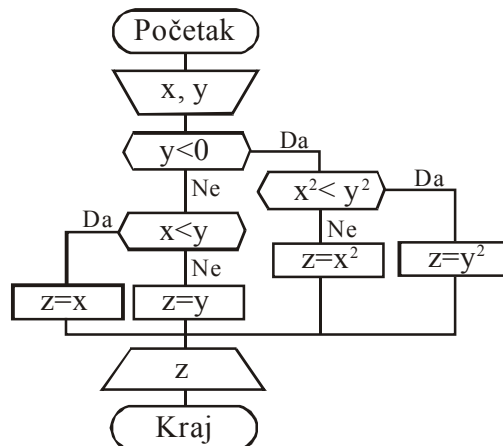
12. Napisati naredbu uslovnog prelaska kojom se za zadate realne brojeve x i y izracunava z po formuli

$$z = \begin{cases} \min(x, y), & y \geq 0 \\ \max(x^2, y^2), & y < 0 \end{cases}$$

```

program ZpoFormuli;
uses WinCrt;
var
  x, y, z : real;
begin
  write('Unesite vrednosti za x i y: ');
  readln(x, y);
  if y < 0
  then
    if sqrt(x) < sqrt(y)
    then z := sqrt(y)
    else z := sqrt(x)
  else
    if x < y then z := x
    else z := y;
  writeln('Vrednost z je: ', z:8:2);
end.

```



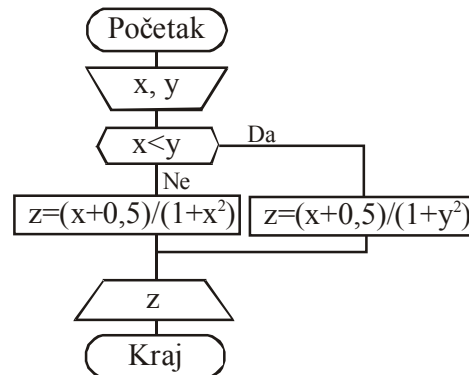
13. Napisati program kojom se za zadate realne brojeve x i y izracunava z po formuli:

$$z = \frac{\min(x, y) + 0.5}{1 + \max^2(x, y)}$$

```

program ZpoFormuli;
uses WinCrt;
var
  x, y, z : real;
begin
  write('Unesite vrednosti za x i y: ');
  readln(x, y);
  if x < y
  then z := (x+0.5)/(1+sqrt(y))
  else z := (y+0.5)/(1+sqrt(x));
  writeln('Vrednost z je: ', z:8:2);
end.

```

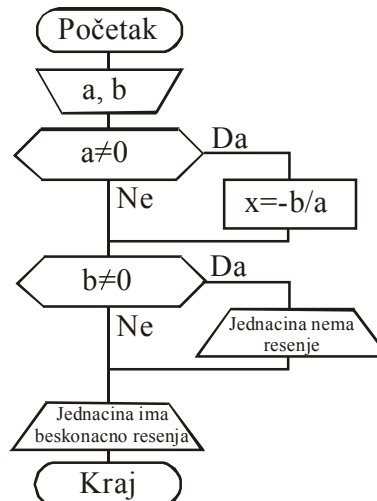


14. Napisati program kojom se resava jednačina: ax+b=0

```

program LinearnaJednacina;
uses WinCrt;
var
  a, b : real;
begin

```



```

write('Unesite koeficijente a i b: ');
readln(a, b);
if a <> 0
  then writeln('x= ', -b/a:10:3)
  else
    if b <> 0
      then writeln('Jednacina nema resenja')
      else writeln('Jednacina ima beskonacno resenja');
    end.
end.

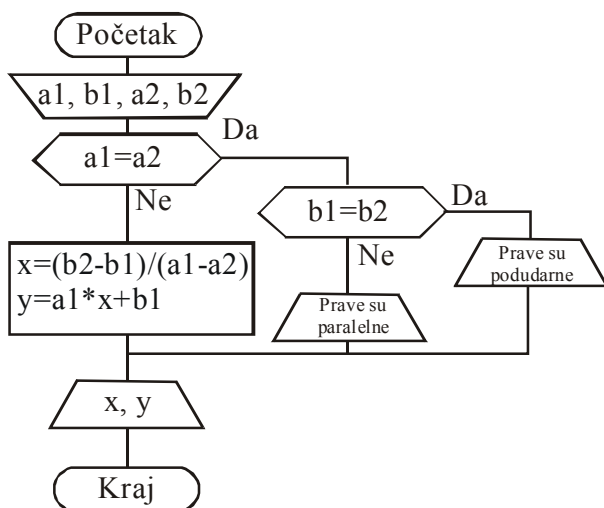
```

15. Napisati program kojim se ispituje da li se seku prave $y=a_1*x+b_1$ i $y=a_2*x+b_2$. Ako se seku odrediti koordinate preseka.

```

program PresekPravih;
uses WinCrt;
var a1, b1, a2, b2, x, y : real;
begin
  write('Unesite koeficijente a1 i b1: ');
  readln(a1, b1);
  write('Unesite koeficijente a2 i b2: ');
  readln(a2, b2);
  if a1=a2 then
    if b1=b2
      then writeln('Prave su podudarne')
      else writeln('Prave su paralelne')
    else
      begin
        x:=(b2-b1)/(a1-a2);
        y:=a1*x+b1;
        writeln('Koordinate preseka pravih su: x=', x:5:2, ', y=', y:5:2)
      end
    end.
end.

```



CASE-ELSE izraz

Opšti oblik CASE izraza je:

```

CASE izraz OF
label-1 : izraz-1;
label-2 : izraz-2;
...
label-n : izraz-n;
ELSE
izraz
END

```

16. Napisati program kojim se na osnovu unetih brojnih ocena 5, 4, 3, 2, 1 ispisuju opisne ocene 'odlican', 'vrlo dobar', 'dobar', dovoljan' i 'nedovoljan'.

```

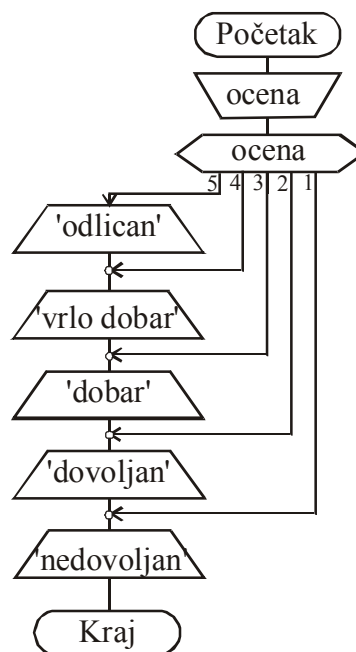
program Ocene;

```

```

uses WinCrt;
var
  ocena : integer;
begin
  write('Unesite ocenu: ');
  readln(ocena);
  case ocena of
    5 : writeln('odlican');
    4 : writeln('vrlo dobar');
    3 : writeln('dobar');
    2 : writeln('dovoljan');
    1 : writeln('nedovoljan');
  end
end.

```

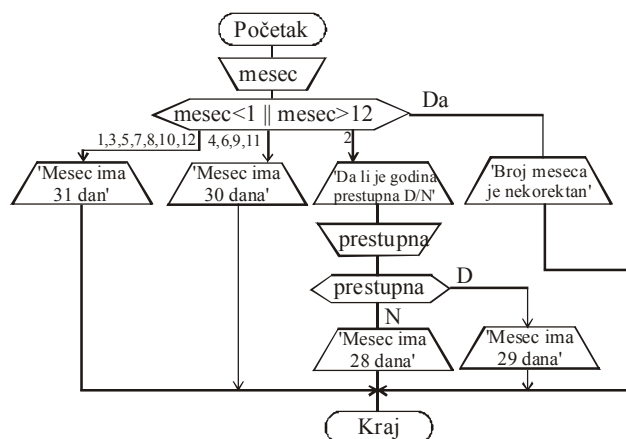


17. Napisati program koji za uneti broj meseca (na primer, 1=januar, 2=februar, itd) ispisuje njegov broj dana. U slučaju unete vrednosti 2 pitati korisnika da li je godina prestupna.

```

program DaniUMesecu;
uses WinCrt;
var
  mesec : integer; prestupna : char;
begin
  write('Unesite broj meseca od 1 do 12: ');
  readln(mesec);
  if (mesec<1) or (mesec>12)
  then writeln('Broj meseca je nekorektan')
  else
  case mesec of
    1,3,5,7,8,10,12 : writeln('Mesec ima 31 dan');
    4,6,9,11       : writeln('Mesec ima 30
dana');
    2               : begin
                      write('Da li je godina
prestupna (D/N)?');
                      readln(prestupna);
                      if (prestupna='D') or (prestupna='d')
                      then writeln('Mesec ima 29 dana')
                      else writeln('Mesec ima 28 dana');
                      end
                    end
  end
end.

```



18. Napisati program kojim se učitava znak za operaciju (+,-,*,/) i dva realna operanda, a zatim stampa rezultat.

```
program MiniKalkulator;
uses WinCrt;
```

```
var
```

```
x, y, rezultat : real;
```

```
operacija : char;
```

```
def : boolean;
```

```
begin
```

```
write('Unesite vrednost za x: ');
```

```
readln(x);
```

```
write('Unesite operaciju: ');
```

```
readln(operacija);
```

```
write('Unesite vrednost za y: ');
```

```
readln(y);
```

```
def:=true;
```

```
case operacija of
```

```
'+' : rezultat:=x+y;
```

```
'-' : rezultat:=x-y;
```

```
'*' : rezultat:=x*y;
```

```
'/' : if y=0 then
```

```
begin
```

```
def:=false;
```

```
write('Rezultat je nedefinisan')
```

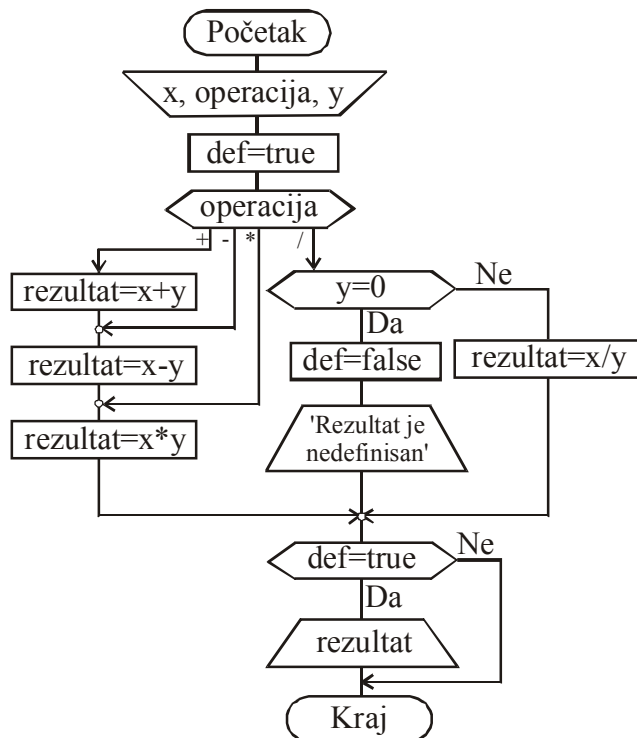
```
end
```

```
else rezultat:=x/y
```

```
end;
```

```
if def then writeln('Rezultat je ', rezultat:10:3)
```

```
end.
```



PETLJE (CIKLUSI)

FOR petlja

FOR petlja se koristi kada izraz ili brok izraza treba ponoviti određeni broj puta. U okviru petlje se nalazi početni uslov, krajnji uslov i brojač. Opšti oblik ove petlje je:

FOR kontrolna promenljiva := izraz-1 **TO** izraz-2 **DO** izraz;

gde je:

kontrolna promenljiva brojač u petlji,

izraz-1 početna vrednost,

izraz-2 krajnja vrednost.

Sve tri promenljive moraju biti istog tipa (nikad realne).

Ukoliko je početna vrednost brojača veća od krajnje, ona se umesto inkrementiranja koristi dekrementiranje, jer se petlja nikad neće izvršiti. Za dekrementiranje se umesto To koristi DOWNTO:

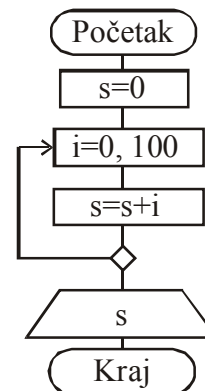
FOR kontrolna promenljiva := izraz-1 **DOWNTO** izraz-2 **DO**

izraz;

U ovoj petlji brojač počinje od veće vrednosti i umanjuje se sve dok ne dostigne kranju vrednost.

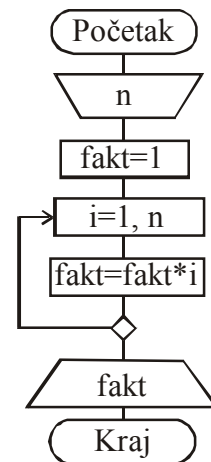
19. Napisati program za učitavanje, sumiranje i stampanje brojeva od 1 do 100.

```
program Suma100;
uses WinCrt;
var
  s : integer;
  i : integer;
begin
  s:=0;
  for i:=1 to 100 do
    s:=s+i;
  writeln('Suma je: ', s);
end.
```



20. Napisati program kojim se za zadato n izracunava faktorijal broja n (faktorijal=n!=1*2*3*...n).

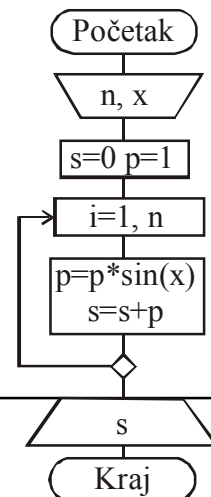
```
program Suma100;
uses WinCrt;
var
  fakt : integer;
  i, n : integer;
begin
  write('Upisati broj n: ');
  readln(n);
  fakt:=1;
  for i:=1 to n do
    fakt:=fakt*i;
  writeln('Faktorijal zadatog broja je: ', fakt);
end.
```



21. Napisati program kojim se za dati ceo broj n i realan broj x,

izracunava izraz: $S = \sin(x) + \sin^2(x) + \dots + \sin^n(x)$

```
program SumaSinus;
uses WinCrt;
var
  i, n : integer;
  s, p, x : real;
begin
  write('Upisati broj n: ');
  readln(n);
  write('Upisati broj x: ');
  readln(x);
  s:=0;
```



```

p:=1;
for i:=1 to n do
begin
  p:=p*sin(x);
  s:=s+p
end;
writeln('Suma izraza je ', s:8:2);
end.

```

22. Napisati program za izracunavanje i stampanje vrednosti realne funkcije:

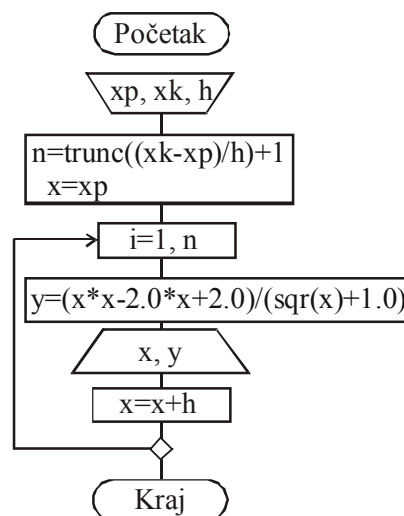
$$y = \frac{x^2 - 2x + 2}{x^2 + 1}$$

ako se x menja od xp do xk sa korakom h.

```

program razlomak;
uses WinCrt;
var
  i, n : integer;
  xp, xk, h, x, y : real;
begin
  write('Upisati vrednosti za xp, xk i korak h: ');
  readln(xp, xk, h);
  n:=trunc((xk-xp)/h)+1;
  x:=xp;
  for i:=1 to n do
  begin
    y:=(x*x-2.0*x+2.0)/(sqr(x)+1.0);
    writeln(x:10:2, y:10:2);
    x:=x+h
  end;
end.

```



WHILE petlja

WHILE petlja se sastoji od uslova koji treba da bude zadovoljen da bi se [završila petlja, pri čemu ne postoji nikakav brojač](#). Opsti oblik ove petlje je:

WHILE uslov **DO**

izraz;

Izraz se izvršava sve dok je uslov ispunjen.

23. Napisati program kojim se izracunava suma neparnih brojeva od 1 do 99.

```

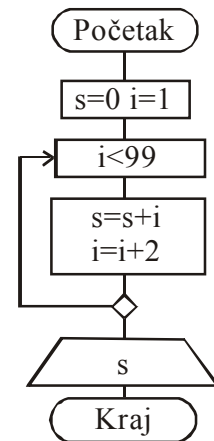
program sumaNeparnih;
uses WinCrt;
var
  i, s : integer;
begin

```

```

s:=0;
i:=1;
while i<99 do
begin
s:=s+i;
i:=i+2;
end;
writeln('Suma neparnih brojeva je: ', s);
end.

```

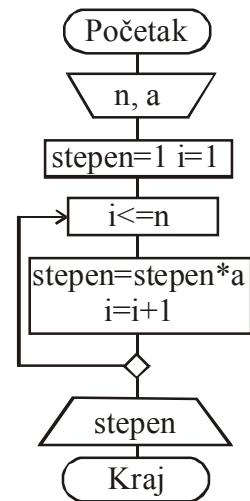


24. Napisati program za izracunavanje n-tog stepena celog broja a

```

program NtiStepen;
uses WinCrt;
var
a, n, i, stepen : integer;
begin
write('Unesite brojeve a i n: ');
readln(a, n);
stepen:=1;
i:=1;
while i<=n do
begin
stepen:=stepen*a;
i:=i+1
end;
writeln('Stepen broja ', a, ' je ', stepen);
end.

```



25. Napisati program kojim se u tekstu cije je obelezje kraja '*' odredjuje broj samoglasnika, suglasnika i praznina.

```

program BrojacSimbola;
uses WinCrt;
var
simbol : char;
brSamoglasnika, brSuglasnika, brPraznina : integer;
begin
brSamoglasnika := 0;
brSuglasnika := 0;
brPraznina := 0;
writeln('Unesite tekst koji se završava simbolom *: ');
read(simbol);
while simbol <> '*' do
begin
case simbol of
'A','E','I','O','U' : brSamoglasnika := brSamoglasnika + 1;
'B','C','D','F','G','H','J','K','L','M','N','P','Q',

```

```

'R','S','T','V','W','X','Y','Z' : brSuglasnika := brSuglasnika + 1;
' ' : brPraznina := brPraznina + 1;
end;
read(simbol)
end;
writeln('Broj samoglasnika je: ', brSamoglasnika);
writeln('Broj suglasnika je: ', brSuglasnika);
writeln('Broj praznina je: ', brPraznina);
end.

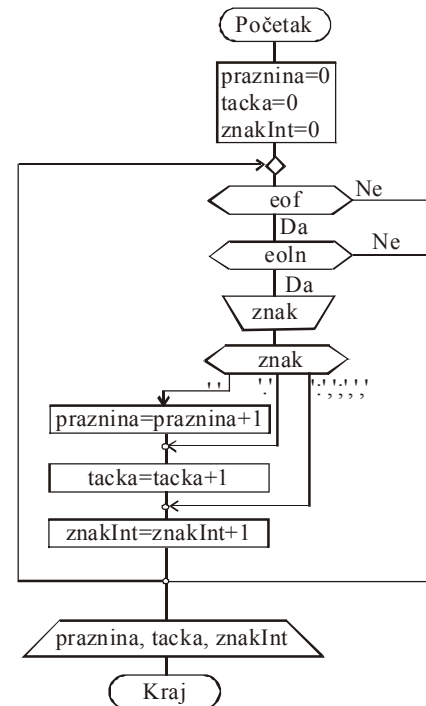
```

26. Napisati program kojim se u standardnoj ulaznoj (input) tekstualnoj datoteci odredjuje broj praznina, tacka i interpukcijskih znakova tipa: zapeta, dvotacka, tacka-zapeta, ako se u datoteci nalaze samo nabrojani znaci.

```

program Znakovi;
uses WinCrt;
var
  znak : char;
  praznina, tacka, znakInt : integer;
begin
  praznina := 0;
  tacka := 0;
  znakInt := 0;
  while not eof do
    begin
      while not eoln do
        begin
          read(znak);
          case znak of
            ' ' : praznina := praznina + 1;
            '.' : tacka := tacka + 1;
            ',', ':', ';' : znakInt := znakInt + 1;
          end {case}
        end; {eoln}
      readln {prelazak na sledecu liniju}
    end; {eof}
  writeln('Praznine: ', praznina);
  writeln('Tacke ', tacka);
  writeln('Broj znakova interpukcije (,,:) je: ', znakInt);
end.

```



27. Napisati program kojim se odredjuje broj jedinica u binarnom zapisu prirodnog broja n.

```

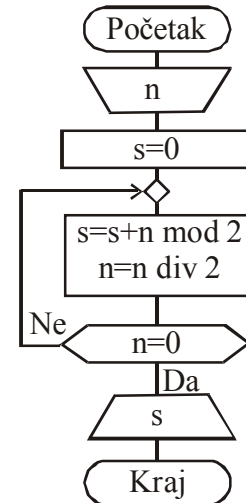
program BrojJedinica;
uses WinCrt;

```

```

var n,s : integer;
begin
  write('Unesite broj n: ');
  readln(n);
  s:=0;
  repeat
    s:=s+n mod 2; {n mod 2 izdvaja cifru 0 ili 1}
    n:=n div 2
  until (n=0);
  writeln('U binarnom zapisu broja ima: ',s,' jedinica');
end.

```



FUNKCIJE

Funkcije predstavljaju podprograme koji vraćaju jednu vrednost. [Funkcije su deo nekog izraza](#) koji se koristi u glavnom programu. Označavaju se sa [rezervisanom reči FUNCTION](#) iza koje sledi ime funkcije i formalni parametri. Razlikuje se od procedure jer se mora odrediti tip funkcije, odnosno tip rezultata. Ukoliko funkcija ne vraca vrednost kao tip rezultata navodi se *void*. Sintaksa funkcije ima sledeci oblik:

FUNCTION Ime_funkcije(VAR formalni_parametri : tip; parametri : tip):tip_funkcije;

Funkcija se može [pozivati tako što se dodeli nekoj promenljivoj funkcija sa parametrima, npr:](#)
 D := Avg(A, B, C);(*Avg je naziv funkcije, a u zagradi su parametri*)

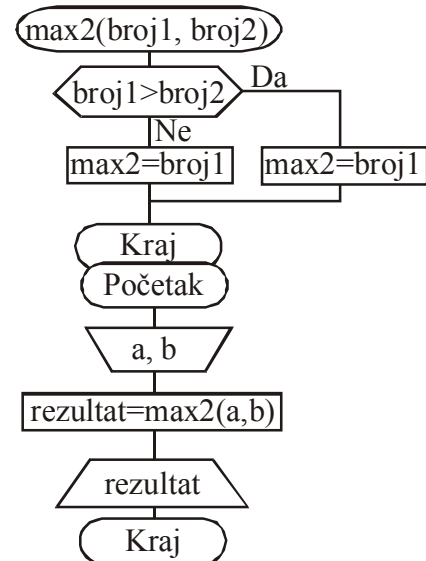
Parametri funkcije mogu biti slova, izrazi ili promenljive.

28. Napisati funkciju pod imenom max koja vraca veci od dva zadata cela broja.

```

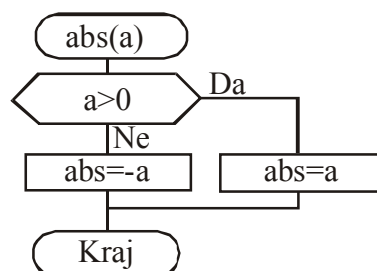
program FunkcijaMax2;
uses WinCrt;
function max2(broj1, broj2 : integer) : integer;
begin
  if broj1 > broj2
  then max2 := broj1
  else max2 := broj2
end;
var
  a, b, rezultat : integer;
begin
  writeln('Unesite dva cela broja: ');
  readln(a,b);
  rezultat := max2(a,b);
  writeln('Veci broj je: ', rezultat);
end.

```



29. **Pretpostavljajući da funkcije abs i odd nisu standardne, sastaviti njihov zapis.**

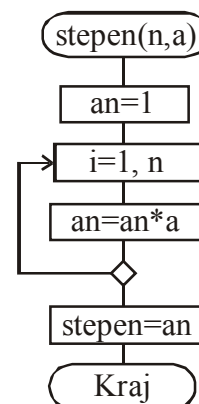
```
function abs(a:real):real;
begin
  if a>=0
  then abs:=a
  else abs:=-a
end;
```



```
function odd(a:integer):boolean;
begin
  odd:=a mod 2=1
end;
```

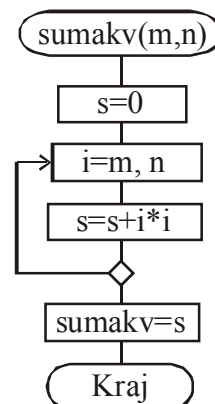
30. **Napisati funkciju za izracunavanje n-tog stepena celog broja a.**

```
function stepen(a, n : integer) : integer;
var an, i : integer;
begin
  an :=1;
  for i:=1 to n do
    an:=an*a;
  stepen :=an
end;
```



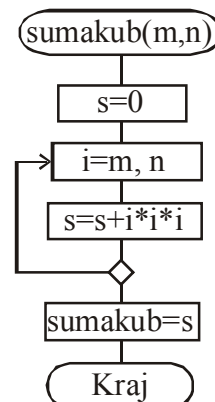
31. **Napisati funkciju kojom se izracunava suma kvadrata celih brojeva od m do n.**

```
function sumakv(m, n : integer): integer;
var
  s, i : integer;
begin
  s:=0;
  for i:=m to n do
    s:=s+i*i;
  sumakv:=s
end;
```



32. **Napisati funkciju kojom se izracunava suma kubova celih brojeva od m do n.**

```
function sumakub(m, n : integer): integer;
var
  s, i : integer;
begin
  s:=0;
  for i:=m to n do
    s:=s+i*i*i;
  sumakub:=s
end;
```



PROCEDURE

Procedura se definiše slično kao program, i sastoji je od zaglavlja, dela za deklaraciju i izraza. Procedura počinje rezervisanom reči PROCEDURE. Parametri koji su rezervisani za izlaz moraju biti deklarirani kao promenljivi parametri. Oni moraju imati rezervisanu reč VAR u svojoj deklaraciji. S druge strane neki parametri služe samo kao ulazni; oni predaju vrednosti proceduri i obično su deklarirani kao parametri vrednosti. Ako parametar služi za oboje, ulaz i izlaz, onda mora biti deklarisan kao promenljivi parametar.

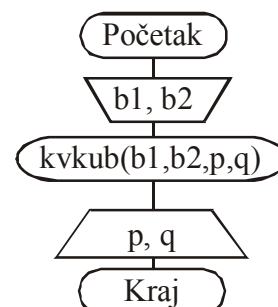
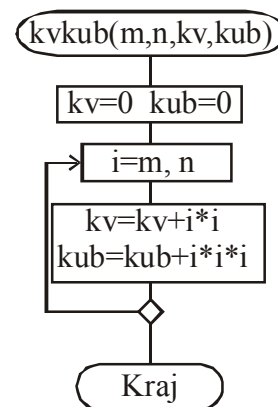
```
procedure ime_procedure(var formalni_parametri : tip);  
begin  
... naredbe  
end;
```

Formalni parametri se nazivaju još i vrednosni parametri. Procedura se poziva na sledeći način:

```
procedure ime;  
ili  
procedure ime(lista parametara);
```

33. Napisati program koji koristi proceduru kvkub cije su ulazne vrednosti dva cela broja m i n , a izlazne vrednosti su cela brojevi p i q . Rezultat p treba da bude zbir kvadrata brojeva od m do n , a rezultat q zbir kubova brojeva od m do n . Umesto pisanja dve funkcije za izracunavanje zbira kvadrata i zbira kubova pisemo jednu proceduru koja izracunava ove dve vrednosti.

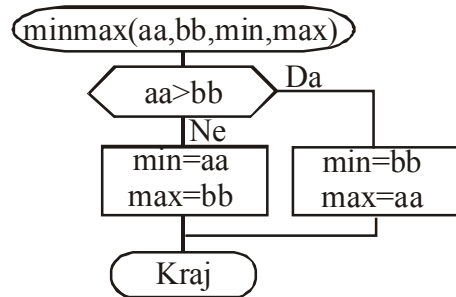
```
program PrimerProcedura;  
{Ilustruje koriscenje procedure}  
uses WinCrt;  
var  
  broj1, broj2, m, n, k, p, q : integer;  
  procedure kvkub(m, n:integer; var kv, kub : integer);  
    {Izracunava sumu kvadrata i sumu kubova brojeva od m do n}  
  var  
    i : integer;  
  begin  
    kv:=0;  
    kub:=0;  
    for i:=m to n do  
      begin  
        kv:=kv+i*i;  
        kub:=kub+i*i*i;  
      end  
    end;  
  end;  
begin  
  write('Upisati vrednosti za broj1 i broj2 (broj1<broj2): ');  
  readln(broj1, broj2);  
  kvkub(broj1, broj2, p, q);
```



```
writeln(p:6, q:6);
end.
```

34. Napisati proceduru kojom se izracunava minimum i maksimum na osnovu dva zadata cela broja.

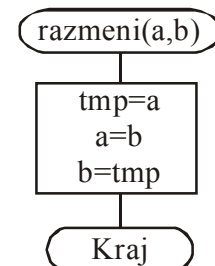
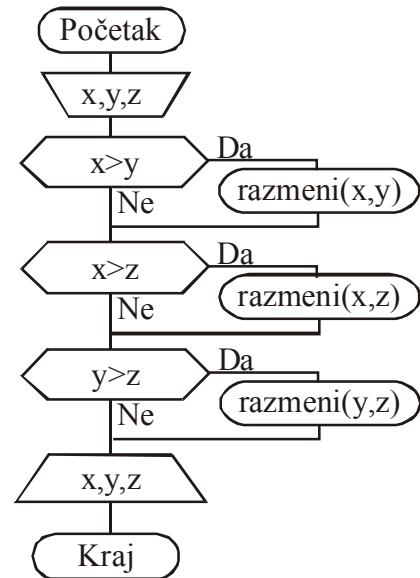
```
procedure minmax(aa, bb:integer; var min, max : integer);
begin
  if aa>bb
  then
    begin
      min:=bb;
      max:=aa;
    end
  else
    begin
      min:=aa;
      max:=bb;
    end
  end;
end;
```



35. Napisati program koji tri broja x, y i z uredjuje u neopadajuci poredak (rastuci). U programu koristiti proceduru pod imenom razmeni, kojom se razmenjuju vrednosti dve promenljive.

Resenje programa se sastoji u tome da se prvo porede x i y, zatim x i z, i na kraju y i z. Ako su dva broja rasporedjena na zeljeni nacin ide se na sledece poredjenje, a ukoliko nisu procedurom razmeni menjaju mesta.

```
program Poredak;
{Ilustruje koriscenje procedure}
uses WinCrt;
var
  x, y, z : integer;
  procedure razmeni(var a, b : integer);
  {Izracunava sumu kvadrata i sumu kubova brojeva od m
  do n}
  var
    tmp : integer;
  begin
    tmp:=a;
    a:=b;
    b:=tmp
  end;
begin
  write('Upisati vrednosti za x, y i z: ');
  readln(x, y, z);
```



```

if x>y then razmeni(x, y);
if x>z then razmeni(x, z);
if y>z then razmeni(y, z);
writeln('Raspored brojeva u neopadajucem (rastucem) poredku je: ', x:6, y:6, z:6);
end.

```

REKURZIJA

Funkcije ili procedure se mogu pozivati unutar funkcije ili procedure i to svojstvo se naziva rekurzija.

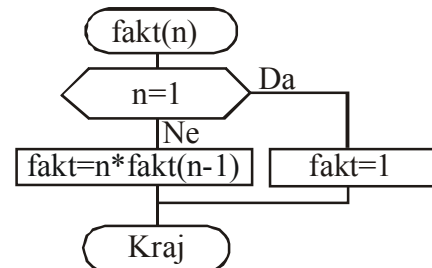
36. Napisati rekurzivnu funkciju za izracunavanje faktoriijala broja n.

fakt(n) = 1 za n=1,
fakt (n) = n*fakt(n-1)

```

function fakt(n : integer): integer;
begin
  if n=1
  then fakt:=1
  else fakt:=n*fakt(n-1)
end;

```



37. Banka na pocetku svake godine ulagacima obracunava kamatu od p procenata na novcanu sumu koja je orocena u toku prethodne godine. Napisati program kojim se izracunava novcani iznos kojim ceraspolagati ulagac posle n godina, ako je pocetno ulaganje S.

Problem se moze resiti rekurzivno:

Ako je an novcani iznos akumuliran nakon n godina, tada je:

a0=s - pocetno ulaganje

a1=s+s*p/100 - iznos posle prve godine

a2=a1+a1*p/100 - iznos posle druge godine

Vidi se da je akumulirani iznos nakon i-te godine:

ai=a(i-1)+a(i-1)*p/100

ili ai=(1+p/100)*a(i-1)

pa se konacno dobija rekurzivna formula:

a0=S

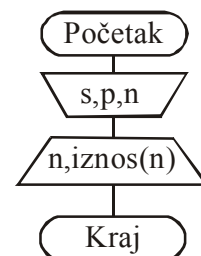
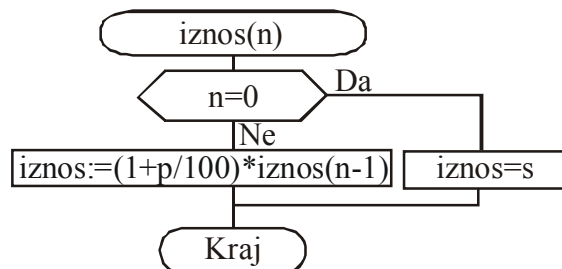
ai=(1+p/100)*a(i-1)

Ova formula se lako transformise u funkciju:

```
function iznos(n:integer):real;
begin
  if n=0
  then iznos:=s
  else iznos:=(1+p/100)*iznos(n-1)
end;
```

```
program Banka;
uses WinCrt;
var
```

```
  n, p, s : integer;
function iznos(n:integer):real;
begin
  if n=0
  then iznos:=s
  else iznos:=(1+p/100)*iznos(n-1)
end;
begin
  write('Unesite pocetni iznos novca, kamatu i broj godina: ');
  readln(s, p, n);
  writeln('Iznos novca posle ', n, ' godina je ', iznos(n):10:2)
end.
```



38. Ako se sa ulaza ucitava niz znakova cija duzina nije poznata, a obelezje kraja niza je *. Napisati program koji obezbedjuje ispis niza u inverznom poretku.

Na primer, ako su ulazni podaci:

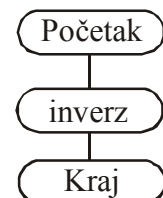
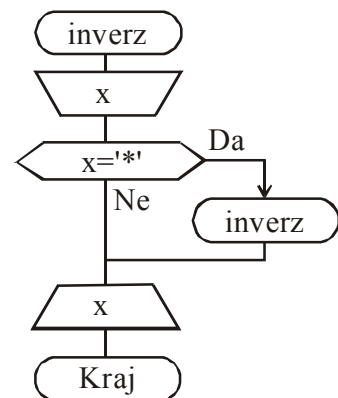
abcde*

racunar na izlazu treba da ispise:

*edcba

```
program inverzno(input,output);
uses WinCrt;
procedure inverz;
var
  x:char;
begin
  read(x);
  if x<>'*' then inverz;
  write(x)
end;
```

```
begin
  write('Upisati niz karaktera koji se zavravaju znakom *: ');
  inverz
end.
```



NIZOVI

Nizovi su skupovi podataka istog tipa. Nizovi su indeksne promenljive. Svaki član niza zauzima određenu memorijsku lokaciju. Ovakvi nizovi se nazivaju jednodimenzionalni nizovi podataka. Korisni su za predstavljanje npr. imena ljudi, rezultata ispita ili nekih sličnih podataka. Opšti oblik predstavljanja nekog niza:

N[1]

N[2]

N[3]

...

N[m]

N-ime niza; brojevi u zagradama predstavljaju indekse niza;

Pored jednodimenzionalnih nizova postoje i dvodimenzionalni nizovi (matrice) koji se koriste za složenije strukture. Dvodimenzionalni nizovi se sastoje od redova (vrsta) i kolona.

Primer dvodimenzionalnog niza:

A[i][j]

A-ime matrice

i-indeks vrste matrice

j-indeks kolone matrice

Definicija nizova je sledeća:

TYPE

Ime niza= **ARRAY**[veličina indeksa] **OF** tip podatka;

ili na pr.:

TYPE

niz = **ARRAY**[1..6] **OF** **INTEGER**;

Deklaracija niza:

VAR

mojniz :niz;

39. Zadat je realan niz x. Napisati program kojim se izracunava srednja vrednost clanova zadatog niza.

program ProsecnaVrednostClanovaNiza;

uses WinCrt;

{Maksimalni dozvoljeni broj elemenata-clanova niza je 100}

const k=100;

{Deklarisemo novi tip podataka - niz od 100 realnih brojeva}

type

niz=array[1..k] of real;

var

n, i : integer;

s : real;

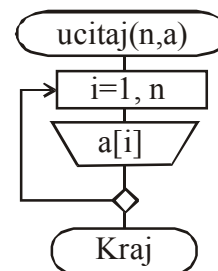
x : niz;

procedure ucitaj(n:integer; var a:niz);

{Procedura ucitava clanove niza}

var

i : integer;



```

begin
  for i:=1 to n do
    read(a[i])
  end;

```

```

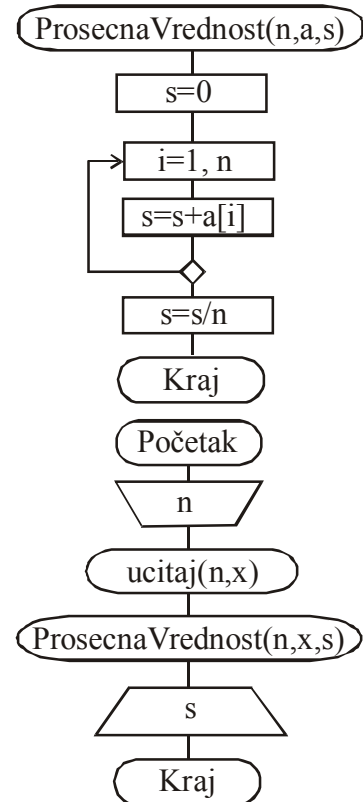
procedure ProsečnaVrednost(n:integer; a:niz; var s:real);
{Procedura koja izracunava prosečnu vrednost članova niza}
var
  i : integer;
begin
  s:=0;
  for i:=1 to n do
    s:=s+a[i];
  s:=s/n
end;

```

```

begin
  write('Unesite broj članova niza: ');
  readln(n);
  writeln('Unesite članove niza: ');
  ucitaj(n, x);
  ProsečnaVrednost(n,x,s);
  writeln('Prosečan vrednost članova niza je: ', s:8:2);
end.

```



40. Zadat je realni niz x. Napisati program kojim se izracunava maksimalna vrednost zadatog niza, korišćenjem procedura.

```

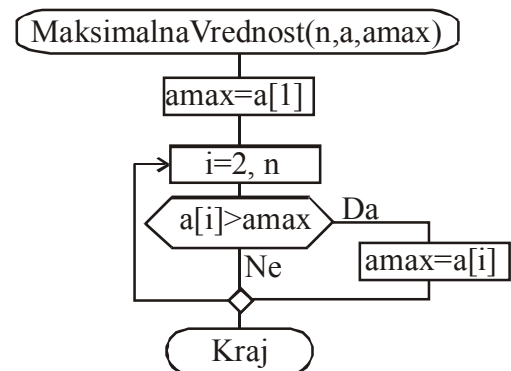
program MaksimalnaVrednostClanovaNiza;
uses WinCrt;
{Maksimalni dozvoljeni broj elemenata-clanova niza je 100}
const k=100;
{Deklarisemo novi tip podataka - niz od 100 realnih brojeva}
type
  niz=array[1..k] of real;
var
  n, i : integer;
  amax : real;
  x : niz;

```

```

procedure ucitaj(n:integer; var a:niz);
{Procedura ucitava članove niza}
var
  i : integer;

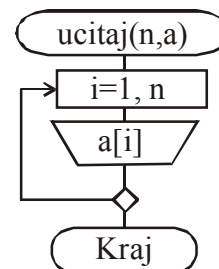
```



```

begin
  for i:=1 to n do
    read(a[i])
  end;

```



```

procedure MaksimalnaVrednost(n:integer; a:niz; var amax:real);
  {Procedura koja izracunava prosecnu vrednost clanova niza}

```

```

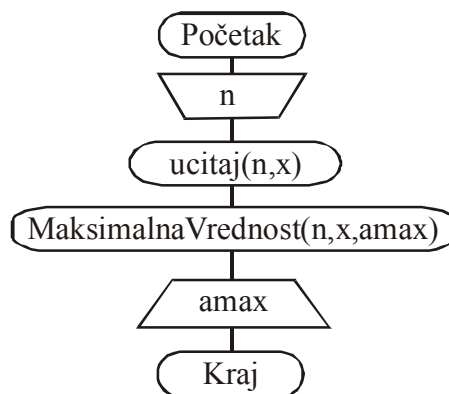
begin
  amax:=a[1];
  for i:=2 to n do
    if a[i]>amax then amax:=a[i];
  end;

```

```

begin
  write('Unesite broj clanova niza: ');
  readln(n);
  writeln('Unesite clanove niza: ');
  {Unosi se jedan rezultat pa se sa Enter prelazi u sledeci
  red, pa se unosi sledeci pa Enter itd}
  ucitaj(n, x);
  MaksimalnaVrednost(n,x,amax);
  writeln('Maksimalna vrednost u nizu je: ', amax:8:2);
end.

```

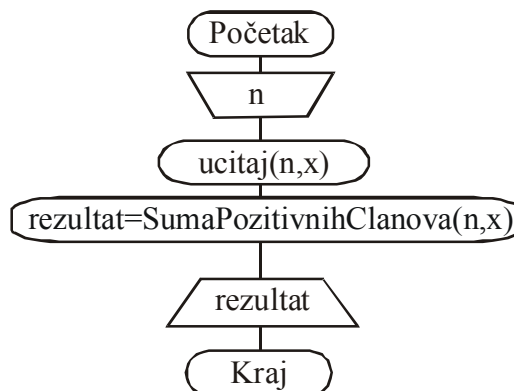


41. Zadat je celobrojni niz x. Napisati program kojim se izracunava suma pozitivnih elemenata zadatog niza.

```

program SumaPozitivnih;
uses WinCrt;
{Maksimalni dozvoljeni broj elemenata-clanova
niza je 30}
const k=30;
{Deklarisemo novi tip podataka - niz od 100
realnih brojeva}
type
  niz=array[1..k] of integer;
var
  n, i : integer;
  rezultat : integer;
  x : niz;

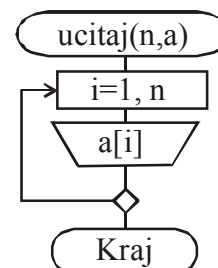
```



```

procedure ucitaj(n:integer; var a:niz);
  {Procedura ucitava clanove niza - u jednoj liniji}
  var
    i : integer;
  begin
    for i:=1 to n do
      read(a[i])
    end;

```



```

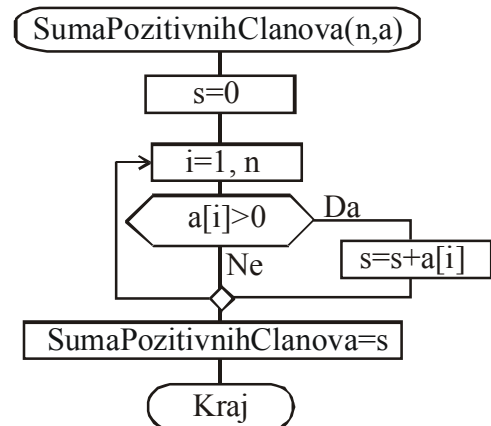
function SumaPozitivnihClanova(n:integer; a:niz):integer;
{Funkcija koja izracunava zbir pozitivnih clanova niza}
var
  i:integer;
  s:real;
begin
  s:=0;
  for i:=1 to n do
    if a[i]>0 then s:=s+a[i];
  SumaPozitivnihClanova:=s
end;

```

```

begin
  write('Unesite broj elemenata niza: ');
  readln(n);
  writeln('Unesite clanove niza: ');
  ucitaj(n, x);
  rezultat := SumaPozitivnihClanova(n,x);
  writeln('Suma pozitivnih elemenata niza je: ', rezultat:8);
end.

```



42. Napisati program kojim se određuje koliko je ucesnika takmicenja iz programiranja imalo natprosecne rezultate.

Problem se moze razloziti na sledece module:

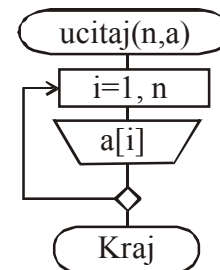
1. *Ucitavanje vrednosti elemenata niza.*
Ovaj modul se realizuje procedurom ucitaj.
2. *Izracunavanje prosecnog rezultata:*
Realizuje se procedurom prosek.
3. *Prebrojavanje natprosecnog rezultata.*
Realizuje se u samom programu.
4. *Ispisivanje izvestaja (izlazni podaci)*
Realizuje se ispisom rezultata u samom programu.

```

program Natprosecni;
uses WinCrt;

const k=100;
type
  niz=array[1..k] of real;
var
  n, broj, i : integer;
  s : real;
  x : niz;

```



```

procedure ucitaj(n:integer; var a:niz);
{Procedura ucitava rezultate takmicara - elemente niza}
var
  i : integer;
begin
  for i:=1 to n do
    readln(a[i])
  end;
end;

```

```

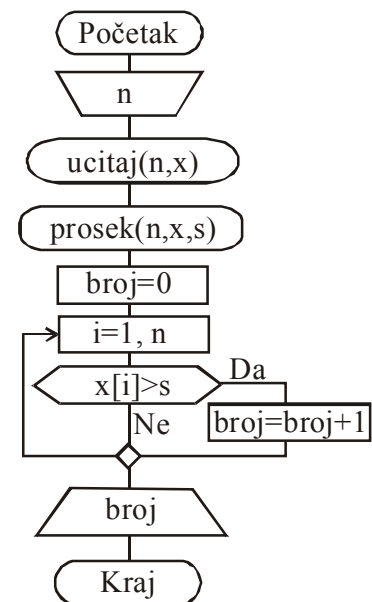
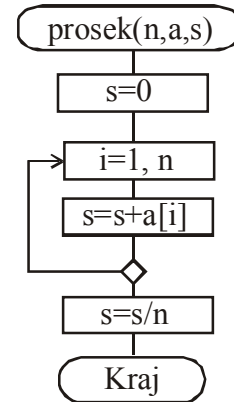
procedure prosek(n:integer; a:niz; var s:real);
{Procedura koja izracunava prosecan rezultat}
var
  i : integer;
begin
  s:=0;
  for i:=1 to n do
    s:=s+a[i];
  s:=s/n
end;

```

```

begin
  write('Unesite broj takmicara: ');
  readln(n);
  writeln('Unesite rezultate: ');
  {Unosi se jedan rezultat pa se sa Enter prelazi u sledeci red, pa se
  unosi sledeci rezultat, pa Enter itd}
  ucitaj(n, x);
  prosek(n,x,s);
  broj:=0; {prebrojavanje prosečnih rezultata}
  for i:=1 to n do
    if x[i]>s then broj:=broj+1;
  writeln('Prosecan rezultat je: ', s);
  writeln('Natprosečne rezultate je imalo: ', broj, ' ucesnika')
end.

```



SORTIRANJE

43. Zadat je celobrojni niz a . Napisati program koji koristi proceduru `sort1` za sortiranje niza u monotono neopadajući poredak.

Prvi algoritam sortiranja radi na sledeći način:

Pri prvom prolasku kroz niz ($i=1$) razmenjuju se vrednosti člana $a[1]$ sa članovima ostatka niza $a[2], a[3], \dots, a[n]$ koji su manji od njega. Zatim (za $i=2$) istim postupkom komponenta $a[2]$ dobija najmanju vrednost od ostatka niza. Procedura `sort1` koristi proceduru `razmeni` da razmeni vrednosti promenljivih.

```

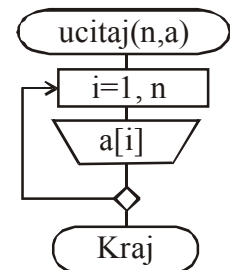
program Sortiranje;
uses WinCrt;
{Maksimalni dozvoljeni broj elemenata-clanova niza je 100}
const k=100;
{Deklarisemo novi tip podataka - niz od 100 realnih brojeva}
type
  niz=array[1..k] of real;
var
  n : integer;
  a : niz;

```

```

procedure ucitaj(n:integer; var a:niz);
{Procedura ucitava clanove niza}
var
  i : integer;
begin
  for i:=1 to n do
    read(a[i])
  end;
end;

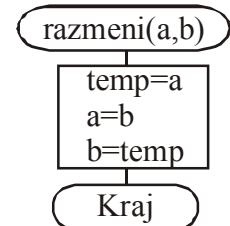
```



```

procedure razmeni(var a,b : real);
var
  tmp : real;
begin
  tmp:=a;
  a:=b;
  b:=tmp
end;

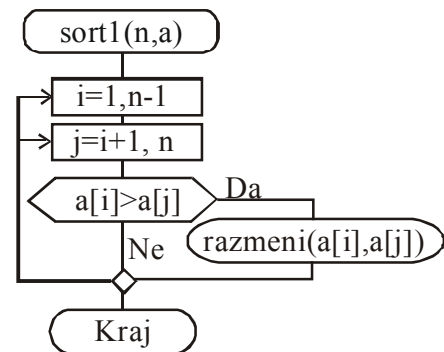
```



```

procedure sort1(n:integer; var a:niz);
{Procedura koja sortira clanove niza u neopadajucem poredku}
var
  i, j : integer;
begin
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if a[i]>a[j] then razmeni(a[i], a[j]);
    end;
  end;
end;

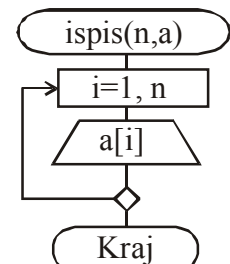
```



```

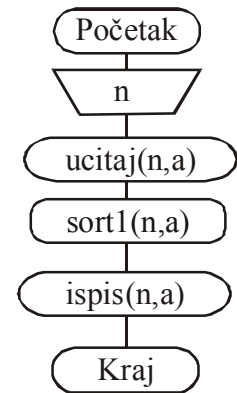
procedure pisi(n:integer; var a:niz);
{Procedura koja ispisuje clanove niza u neopadajucem poredku}
var
  i : integer;
begin
  for i:=1 to n do

```



```
begin
  write(a[i]:6:2);
  if i mod 10 = 0 then writeln
end
end;
```

```
begin
  write('Unesite broj elemenata niza: ');
  readln(n);
  writeln('Unesite elemente niza: ');
  ucitaj(n, a);
  sort1(n, a);
  writeln('Sortirani niz je: ');
  pisi(n, a);
end.
```



MATRICE

Dvodimenzionalni nizovi se sastoje od redova (vrsta) i kolona.

Primer dvodimenzionalnog niza (matrice):

$A[i][j]$

A-ime matrice

i-indeks vrste matrice

j-indeks kolone matrice

Deklaracija matrice:

type

dim=1..max;

mat=array[dim,dim] of tip;

var

A:mat;

ili

const m=10;

type niz=array[1..m] of real;

mat=array[1..m] of niz;

var

a:mat;

44. Napisati program (funkciju) kojom se za realnu matricu A (NxN) izracunava suma svih elemenata (clanova) matrice.

```
program SumaClanovaMatrice;
uses WinCrt;
```

```
const k=10;
```

```
type
```

```
niz=array[1..k] of real;
```

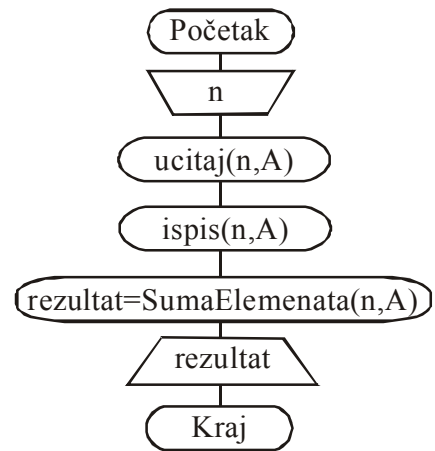
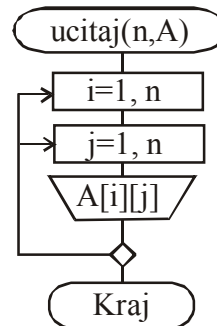
```
matrica=array[1..k] of niz;
```

```
var
```

```
A : matrica;
```

```
rezultat : real;
```

```
i, j, n : integer;
```



```
procedure ucitajMatricu(n:integer; var A:matrica);
```

```
var
```

```
i, j : integer;
```

```
begin
```

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

```
for j:=1 to n do
```

```
read(A[i][j])
```

```
end;
```

```
function SumaElemenata(n:integer; A:matrica) : real;
```

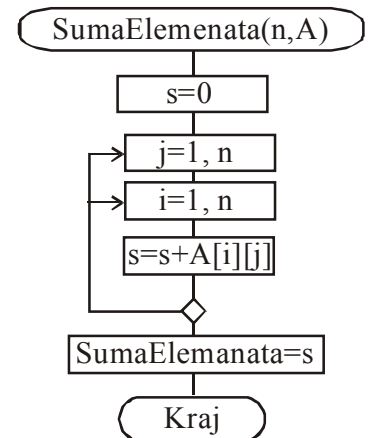
```
var
```

```
s : real;
```

```
i, j : integer;
```

```
begin
```

```
s:=0;
```



```

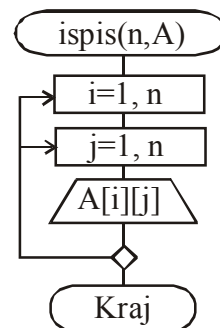
for i:=1 to n do
  for j:=1 to n do
    s:=s+A[i,j];
    SumaElemenata:=s
end;

```

```

begin
write('Upisati dimenzije matrice (n):');
readln(n);
write('Unesite elemente matrice (3x3): ');
ucitajMatricu(n, A);
writeln('Elementi matrice su: ');
for i:=1 to n do
  for j:=1 to n do
    writeln(A[i][j]:8:2);
  writeln;
rezultat:=SumaElemenata(n, A);
write('Suma elemenata matrice je: ', rezultat:8:2);
end.

```



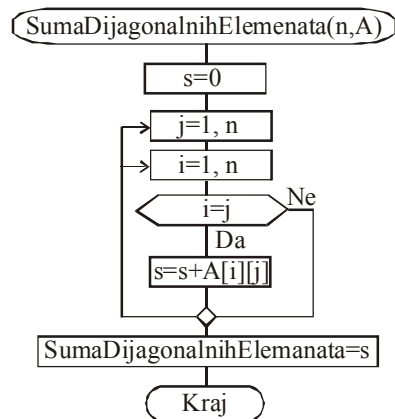
end.

45. Napisati program (funkciju) kojom se za realnu matricu A (NxN) izracunava suma svih elemenata (clanova) matrice na glavnoj dijagonali.

```

function SumaDijagonalnihElemenata(n:integer; A:matrica) :
real;
var
  s : real;
  i, j : integer;
begin
  s:=0;
  for i:=1 to n do
    for j:=1 to n do
      if i=j then
        s:=s+A[i,j];
        SumaDijagonalnihElemenata:=s
end;

```



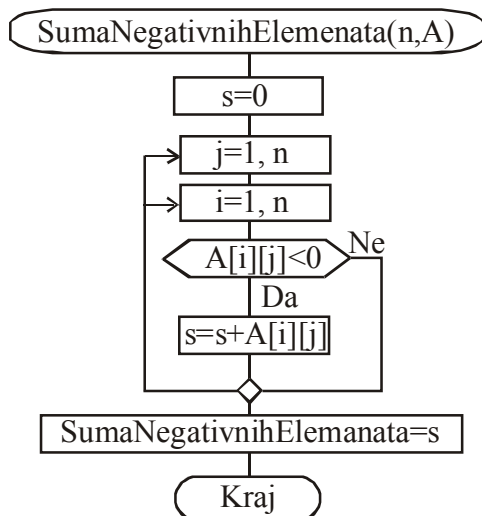
46. Napisati program (funkciju) kojom se za realnu matricu A (NxN) izracunava suma svih negativnih elemenata (clanova) matrice.

```

function SumaNegativnihElemenata(n:integer; A:matrica) : real;
var
  s : real;
  i, j : integer;
begin

```

```
s:=0;
for i:=1 to n do
  for j:=1 to n do
    if A[i,j]<0 then
      s:=s+A[i,j];
  SumaNegativnihElemenata:=s
end;
```



STRINGOVI

Funkcije i procedure kod stringova su sledeće:

Funkcije:

LENGTH(str) -vraća broj karaktera u stringu str.

CONCAT(str1, str2,...) –vraća string [koji se dobija spajanjem stringova str1, str2,...](#)

COPY(str, pos, len) –vraća podstring iz [stringa str, koji počinje od pozicije pos, sa dužinom len.](#)

POS(str1, str2)-vraća poziciju prvog pojavljivanja karaktera str1 u str2. Ako se str1 ne sadrži u str2 vraća se nula.

Procedure:

INSERT(str1, str2, pos) -ubacuje string str1 u string str2, od mesta pos.

DELETE(str, pos, len) -briše podstring iz stringa str počevši od pozicije pos dužine len.

EOLN je logička funkcija se koristi za utvrđivanje kraja linije pri čitanju ulaznog fajla. Funkcija ima vrednost false sve dok se ne dostigne kraj reda kada postaje true. Ova funkcija je korisna kada se ne zna koliko karaktera ima u redu. Za ovu funkciju se može koristiti petlja:

```
WHILE NOT EOLN DO
BEGIN
READ(Ch);
...
END;
```

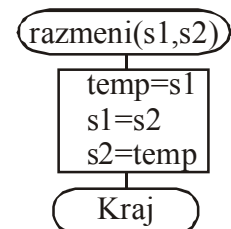
Druga Bulova funkcija je EOF koja se koristi za određivanje kraja datoteke. Funkcija daje vrednost FALSE sve dok se ne dođe do kraja datoteke kada dobija vrednost TRUE. Obično se koriste funkcije EOF i EOL u kombinaciji kako bi se analiziralo više [redova teksta.](#)

```
WHILE NOT EOF DO
BEGIN
WHILE NOT EOLN DO
BEGIN
READ(Ch);
... { obrada podataka }
END; { Kraj linije }
READLN { pozicioniranje na početak novog reda }
END; { kraj fajla }
```

47. Napisati program kojim se učitavaju prezimena studenata iz jedne grupe, a zatim se ispisuju u alfabetskom poretku.

```
program Alfabet;
uses WinCrt;

const k=50;
type maxstring=string[30]; {maksimalni broj slova u prezimenu}
var i, j, n : integer;
    prezime : array[1..k] of maxstring;
```



```

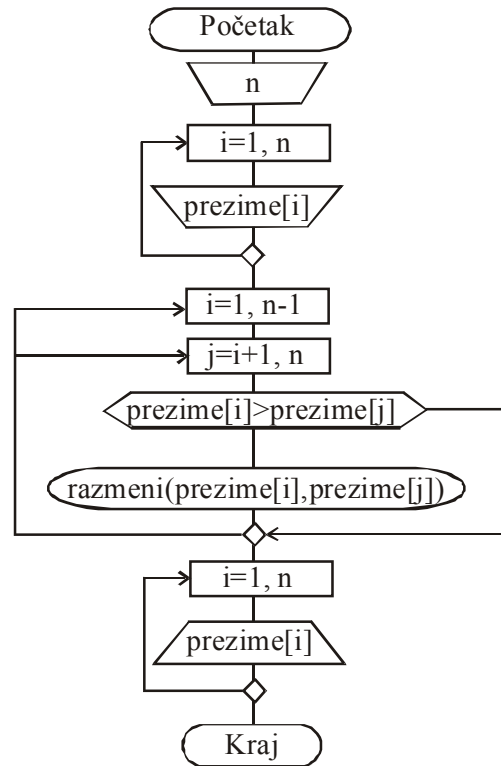
procedure razmeni(var s1,s2 : maxstring);
var
  tmp : string[30];
begin
  tmp:=s1;
  s1:=s2;
  s2:=tmp;
end;

```

```

begin
write('Unesite broj studenata: ');
readln(n);
writeln('Unesite prezimena: ');
for i:=1 to n do
  readln(prezime[i]);
for i:=1 to n-1 do
  for j:=i+1 to n do
    if prezime[i]>prezime[j]
    then razmeni(prezime[i], prezime[j]);
writeln('Prezimana studenata uredjena po
alfabetskom redosledu su: ');
for i:=1 to n do
  writeln(prezime[i]);
end.

```



48. Napisati program kojim se u zadatom stringu s broji pojavljivanje datog znaka x.

```

program BrojZnakova;
uses WinCrt;

```

```

var i, brojac : integer;
    s : string;
    x : char;

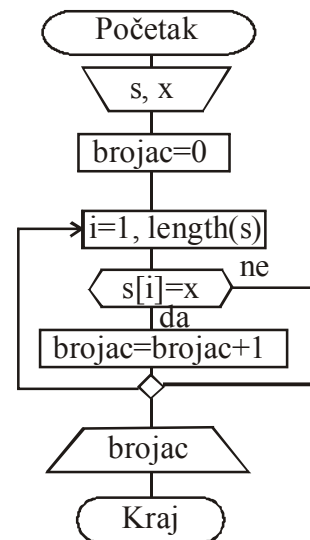
```

```

begin
write('Unesite tekst: ');
readln(s);
writeln('Unesite znak x: ');
readln(x);
brojac:=0;

for i:=1 to length(s) do
  if s[i]=x then
    brojac:=brojac+1;
writeln(x, ' se pojavljuje ', brojac, ' puta. ');
end.

```

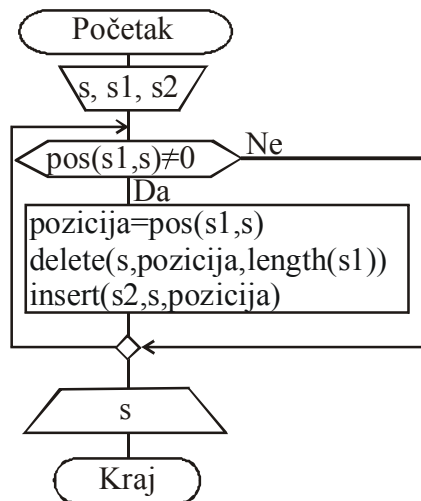


49. Napisati program kojim se u zadatom stringu s zamenjuje svako pojavljivanje stringa s1 sa s2.

```

program Zamena;
uses WinCrt;
var
  pozicija : integer;
  s, s1, s2 : string;
begin
  write('Unesite tekst: ');
  readln(s);
  writeln('Zameniti string (s1): ');
  readln(s1);
  writeln('Sa stringom (s2): ');
  readln(s2);
  while pos(s1,s) <> 0 do
  begin
    pozicija:=pos(s1,s);
    delete(s,pozicija,length(s1));
    insert(s2,s,pozicija);
  end;
  writeln(s);
end.

```

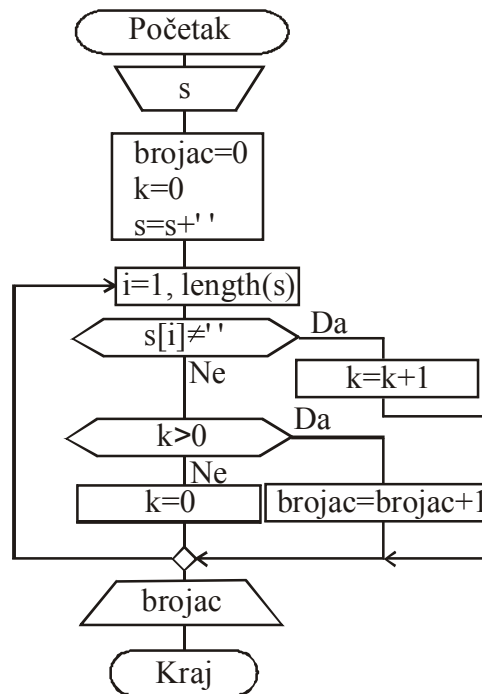


50. Napisati program koji broji reci u zadatom stringu. Rec je niz znakova izmedju praznina (razmaka).

```

program BrojReci;
uses WinCrt;
var i, brojac, k : integer;
    s : string;
begin
  write('Unesite string: ');
  readln(s);
  brojac:=0;
  k:=0;
  s:=s+' ';
  for i:=1 to length(s) do
  if s[i] <> ' '
  then k:=k+1
  else
  begin
    if k>0 then brojac:=brojac+1;
    k:=0;
  end;
  end;
  writeln('Broj reci u stringu je: ', brojac);
end.

```



SLOGOVI

Slogovi su skupovi podataka međusobno povezanih koji mogu biti različitog tipa. Svaki element zapisa naziva se polje. Deklaracija slogova se sastoji u sledećem:

type

identifikator = **RECORD**

lista polja

END;

Npr. zapis o zaposlenima:

TYPE

Zaposleni = RECORD

Ime :STRING[25];

Adresa :STRING[40];

tel :STRING[12];

Plata :REAL;

END;

Pristup poljima podataka se vrši tako što se ime sloga i identifikator polja odvajaju tačkom. Na primer:

Zaposleni.Ime := 'Pero Peric';

Zaposleni.Plata := 22500;

Ako treba da se odštampa npr. podatak o imenu:

WRITELN('Ime zaposlenog: ', Zaposleni.Ime);

WITH izraz

WITH izraz omogućava da se pristupi poljima sloga bez korišćenja promenljivih polja. Efekat upotrebe with izraza je da se pripoji svako ime polja imenu sloga. Ako neka od promenljivih u bloku nije identifikator polja, neće se promeniti sa with izrazom. Ako iza with ima samo jedan izraz, nema potrebe da se koristi BEGIN-END blok. With se može koristiti pri pozivanju procedure npr:

WITH ZaposleniRec DO

PrikaziRezultat(Ime, Plata);

Ovo je ekvivalentno sa:

PrikaziRezultat(ZaposleniRec.Ime, ZaposleniRec.Plata);

WITH izraz ima opšti oblik:

WITH identifikator-sloga **DO** izraz;

51. Deklarisati slog osoba sa sledecim podacima: Ime, adresa, starost, zanimanje(ponuditi sledeca zanimanja - student, inzenjer, menadzer, ekonomista). Napisati program za citanje slogova kojim se opisuju dve osobe i ispisuju informacije o starijoj osobi.

program Stariji;

uses WinCrt;

{Program cita podatke o dve osobe i ispisuje podatke o starijoj osobi}

type

licnost=record

```

ime:string;
adresa:string;
starost:integer;

```

```

zanimanje:(student,inzenjer,menadzer,ekonomista)
end;

```

```

var
  starija, osoba1,osoba2:licnost;

```

```

procedure citanjeSloga(var osoba:licnost);
{Procedura koja cita slog}

```

```

var
  simbol:char;
  pravilno : boolean;
begin
  with osoba do
  begin
    write('Unesite ime i prezime osobe: ');
    readln(ime);
    write('Unesite adresu osobe: ');
    readln(adresa);
    repeat
      write('Unesite starost osobe: ');
      readln(starost);
      pravilno:=(starost>=0) and (starost<=120);
      if not pravilno then writeln('Greska u unosenu
godina!');
    until pravilno;
    writeln('Unesite S, ako je osoba student');
    writeln('Unesite I, ako je osoba inzenjer');
    writeln('Unesite M, ako je osoba menadzer');
    writeln('Unesite E, ako je osoba ekonomista');
    readln(simbol);
    case simbol of
      'S' : zanimanje:=student;
      'I' : zanimanje:=inzenjer;
      'M' : zanimanje:=menadzer;
      'E' : zanimanje:=ekonomista;
    end;
    writeln
  end
end;

```

```

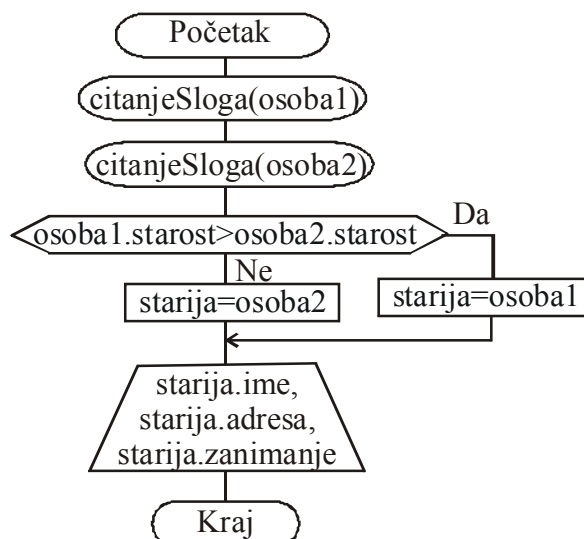
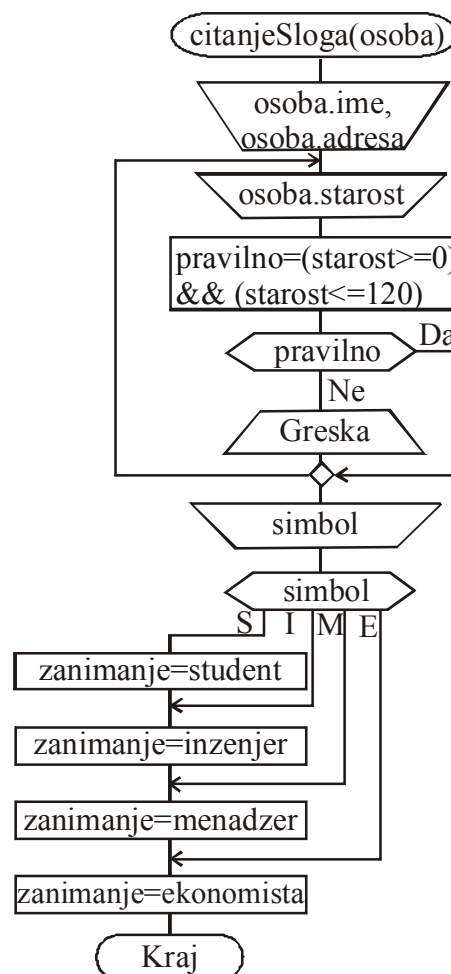
begin
  citanjeSloga(osoba1);
  citanjeSloga(osoba2);
  if osoba1.starost>osoba2.starost

```

```

begin
  citanjeSloga(osoba1);
  citanjeSloga(osoba2);
  if osoba1.starost>osoba2.starost

```



```

then starija:=osoba1
else starija:=osoba2;
with starija do
begin
writeln('Starija osoba je: ');
writeln(ime);
writeln(adresa);
case zanimanje of
student : writeln('Student');
inzenjer : writeln('Inzenjer');
menadzer : writeln('Menadzer');
ekonomista : writeln('Ekonomista');
end
end
end.

```

52. Napisati program kojim se citaju slogovi koji sadrze prezime, adresu i broj telefona, a zatim ih ispisuju u afabetskom poretku.

Zadatak se moze razbiti na tri modula:

1. Citanje slogova
2. Sortiranje slogova
3. Ispisivanje sortiranih slogova

Da bi program bio jednostavniji citanje, ispisivanje i sortiranje slogova pisemo u obliku procedura.

```

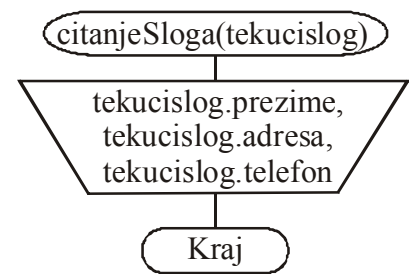
program Imenik;
uses WinCrt;
{Program ispisuje alfabetски uredjen imenik}
const
maks=50; {Maksimalan broj imena u imeniku}
type
licnost=record
prezime:string;
adresa:string;
telefon:string;
end;
niz=array[1..maks] of licnost;
var
i, brojac : integer;
spisak : niz;
simbol : char;

```

```

procedure citanjeSloga(var tekuciSlog:licnost);
{Procedura za ucitavanje sloga}

```



```

begin
with tekucislog do
begin
write('Unesite prezime osobe: ');
readln(prezime);
write('Unesite adresu osobe: ');
readln(adresa);
write('Unesite telefon: ');
readln(telefon);
end
end;

```

```

procedure ispisSloga(var tekuciSlog:licnost);
{Procedura za ispisivanje sloga na ekranu}
begin
writeln("");
with tekuciSlog do
begin
writeln(prezime);
writeln(adresa);
write('Tel. ', telefon);
end
end;

```

```

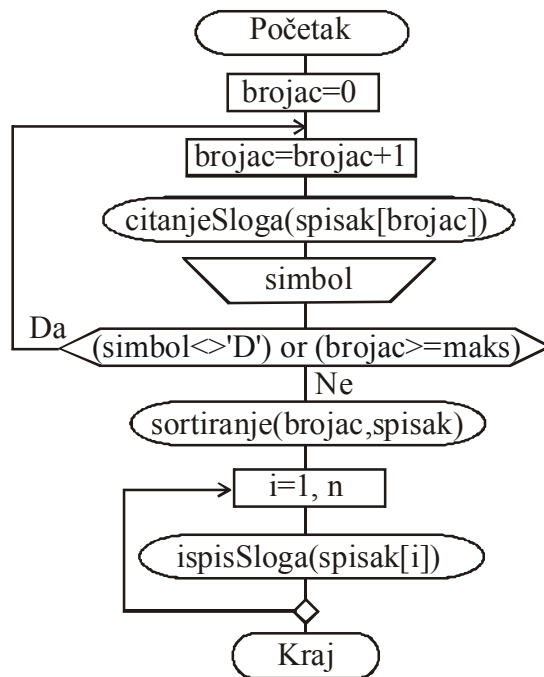
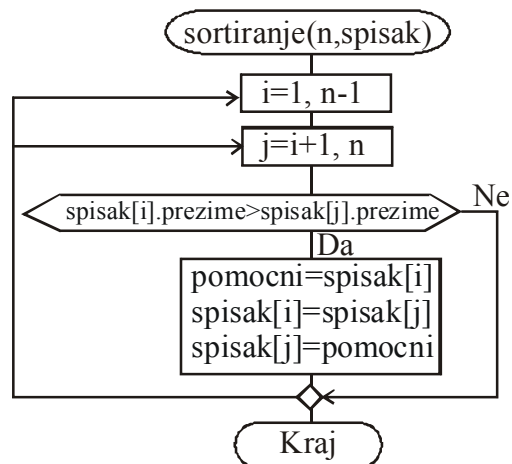
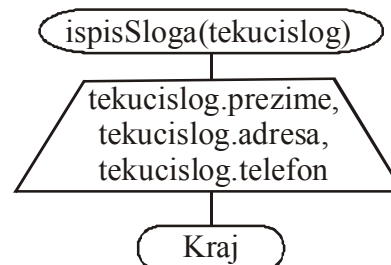
procedure sortiranje(n:integer; var spisak:niz);
{Procedura sortira slogove po prezimenu}
var
i, j : integer;
pomocni : licnost;
begin
for i:=1 to n-1 do
for j:=i+1 to n do
if spisak[i].prezime>spisak[j].prezime
then
begin
pomocni:=spisak[i];
spisak[i]:=spisak[j];
spisak[j]:=pomocni
end
end;

```

```

begin
brojac:=0;
repeat
brojac:=brojac+1;
citanjeSloga(spisak[brojac]);
writeln('Ima li jos slogova (D/N): ');

```



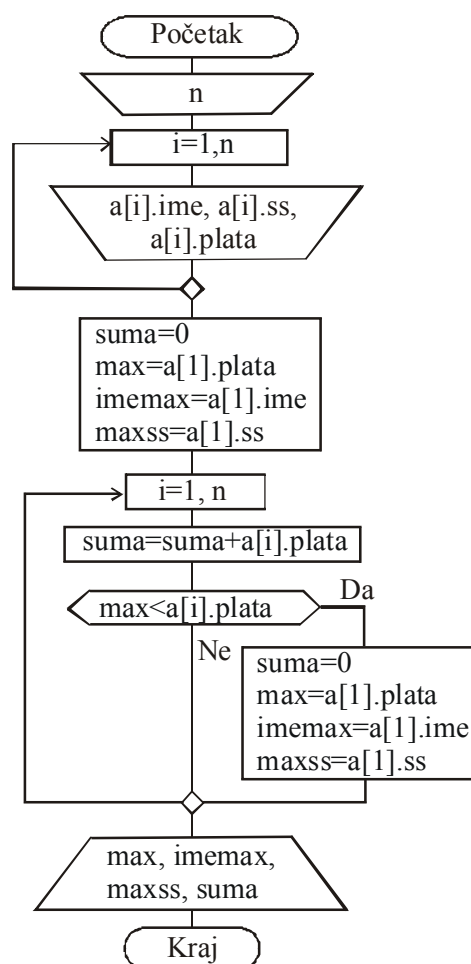
```

    readln(simbol);
until (simbol<>'D') or (brojac>=maks);
sortiranje(brojac, spisak);
for i:=1 to brojac do
begin
    write('Pritisnite <ENTER> za ispis sledeceg sloga');
    read(simbol);
    ispisSloga(spisak[i])
end;
writeln('Ispisani su svi slogovi.');
```

53. Nacrtati algoritam i napisati program koji za zadati slog o zaposlednima sa poljima ime, stručna sprema i plata određuje ukupnu sumu novca potrebnu za isplatu plata i ispisati ime radnika sa najvišom platom.

```

PROGRAM slog1;
uses wincrt;
TYPE
Zaposleni = RECORD
    Ime :STRING[25];
    ss :STRING[40];
    Plata :REAL;
END;
type
    niz=array[1..20] of Zaposleni;
VAR
i,n:integer;
suma,max:real;
a:niz;
radnik:string;
BEGIN
    writeln('Uneti broj zaposlenih');
    readln(n);
    for i:=1 to n do
        begin
            writeln('Uneti ime');
            readln(a[i].Ime);
            writeln('Uneti strucnu
spremu');
            readln(a[i].ss);
            writeln('Uneti platu');
            readln(a[i].Plata);
        end;
    suma:=0;
    max:=a[1].plata;
    radnik:=a[1].ime;
    for i:=1 to n do
```



```

begin
  suma:=suma+a[i].plata;
  if max<a[i].plata then
  begin
    max:=a[i].plata;
    radnik:=a[i].ime;
  end
end;
writeln ('Ime radnika sa najvecom platom je: ', radnik);
writeln ('Ukupna suma novca za isplatu plata: ', suma:10:2);
END.

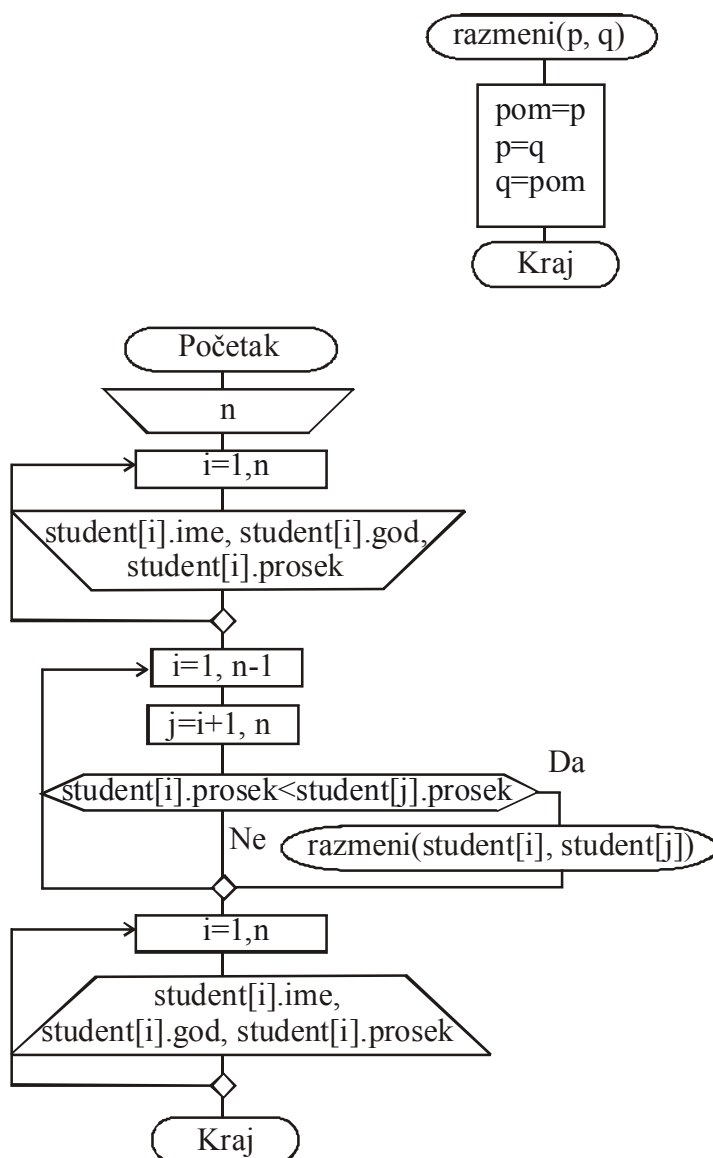
```

54. Napisati program i nacrtati algoritam koji za zadati slog student sa podacima ime, godina studija i prosek sortira sutdente prema proseku i ispisuje tako sortiran spisak.

```

program student1;
uses wincrt;
type
  slog=record
    ime:string;
    god:integer;
    prosek:real
  end;
type
  niz=array[1..30] of slog;
var
  student:niz;
  pom:slog;
  n,i,j:integer;
procedure razmeni(var p,q:integer);
var pom:integer;
begin
  pom:=p;
  p:=q;
  q:=pom
end;
begin
  write('unesi broj studenata:');
  readln(n);
  writeln('unesi podatke o
studentima:');
  for i:=1 to n do
  begin
    write('Ime ');
    readln(student[i].ime);
    write('Godina studija ');
    readln(student[i].god);

```



```

        write('Prosecna ocena ');readln(student[i].prosek);
    end;
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if student[i].prosek< student[j].prosek then
                razmeni(student[i], student[j]);
        writeln;
    writeln('Sortirani podaci po prosečnoj oceni su:');
    for i:=1 to n do
        begin
            writeln('Ime ',student[i].ime);
            writeln('Godina studija ',student[i].god);
            writeln('Prosecna ocena ',student[i].prosek:4:2);
        end;
    writeln;
end.

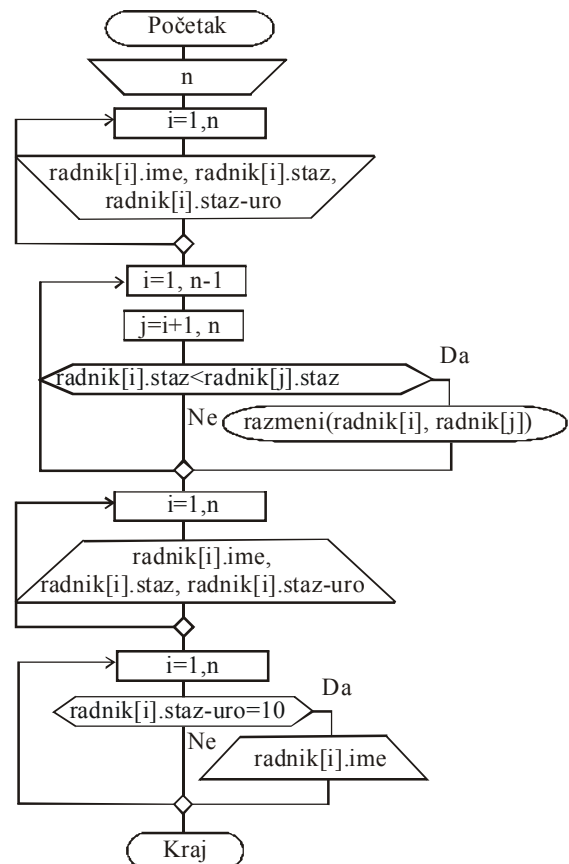
```

55. Nacrtati algoritam i napisati program za zadati slog o radnicima koji se sastoji od polja: ime, staz proveden u radnoj organizaciji i ukupan staz. Izvršiti sortiranje radnika prema ukupnom stazu i ispisati imena radnika koji imaju 10 godina staza u radnoj organizaciji.

```

program ro;
uses wincrt;
type
    slog=record
        ime:string;
        staz:integer;
        staz_uro:integer
    end;
type
    niz=array[1..30] of slog;
var
    radnik:niz;
    pom:slog;
    n,i,j:integer;
procedure razmeni(var p,q:integer);
    var pom:integer;
    begin
        pom:=p;
        p:=q;
        q:=pom
    end;
begin
    write('unesi broj radnika:');
    readln(n);
    writeln('unesi podatke o radnicima:');
    for i:=1 to n do

```



```
begin
  with radnik[i] do
    begin
      write('Ime '); readln(ime);
      write('Ukupan radni staz ');
      readln(staz);
      write('Staz u radnoj organizaciji ');
      readln(staz_uro);
      writeln
    end
  end;
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if radnik[i].staz< radnik[j].staz then
        razmeni(radnik[i], radnik[j]);
      writeln('Sortirani podaci po ukupnoj duzini radnog staza:');
      for i:=1 to n do
        with radnik[i] do
          begin
            writeln('Ime: ',ime);
            writeln('Ukupan staz: ',staz);
            writeln('Staz u radnoj organizaciji: ',staz_uro);
            writeln
          end;
        end;
      writeln;
      writeln('Imena radnika sa 10 god radnog staza_uro: ');
      for i:=1 to n do
        if radnik[i].staz_uro=10 then writeln (radnik[i].ime);
      end.
end.
```

DATOTEKE

FILE ili datoteka se definiše kao skup [povezanih celina koji se nalaze na disku ili drugom medijumu](#). Nad datotekom se mogu izvoditi sledeće operacije:

1. čitanje iz fajla
2. upisivanje u fajl.

Datoteke mogu biti tekstualne ili binarne. Tekstualne datoteke su skupovi struktura i svih njegovih karaktera tipa char. Tekstualne datoteke se sastoje od linija karaktera razdvojenih sa krajem linije (EOLN). Fajl na disku izgleda isto kao i tekst na ekranu koji se unosi preko tastature.

Pristup tekstualnoj datoteci

Da bi pristupili tekstualnoj datoteci, moramo deklarirati promenljivu file sa tipom text. Npr.:

```
var
```

```
f:text;
```

Otvaranje datoteke

Da bi čitali podatke iz datoteke, moramo je prvo otvoriti sa standardnom procedurom reset:

```
reset(f);
```

Da bi **povezali promenljivu** f sa datotekom koja se nalazi na disku koristimo proceduru assign. Ukoliko znamo ime datoteke navodimo u ovom proceduri:

```
assign(f,c: \config.sys);
```

```
reset(f);
```

Važno je naglasiti da kada se navodim ime fajla, ukoliko nije u istom direktorijumu u kome se program izvršava, potrebno je navesti putanju datoteke.

Zatvaranje datoteke vrši se [procedurom close, kada završimo sa otvaranjem i manipulacijom](#) sa podacima u datoteci.

```
close(f);
```

Opšta formula za [pripremu tekstualne datoteke je sledeća:](#)

```
{Zaglavlje programa}
```

```
PROGRAM Ime;
```

```
{Deklaracija promenljivih}
```

```
f:TEXT;
```

```
{Povezivanje imena datoteke sa imenima promenljivih file}
```

```
ASSIGN(f, ime-file);
```

```
{Otvaranje datoteke}
```

```
RESET(f);
```

```
{Zatvaranje datoteke}
```

```
CLOSE(f);
```

Čitanje [podataka iz datoteke vrši se pomoću procedura read i readln](#).

```
READLN(f, lista podataka);
```

```
READ(f, lista podataka);
```

Funkcije za određivanje kraja linije i kraja datoteke su:

```
EOLN(f)
```

```
EOF(f)
```

Ispisivanje sadržaja [tekstualne datoteke vrši se pomoću write, ali prethodno se podaci moraju pročitati iz datoteke](#)

```
READLN(f,Ch);
```

```
WRITE(Ch);
```

Da bi se razdvojile linije u datoteci pri ispisu na ekranu nepohodno je koristiti naredbu writeln, kada se utvrdi kraj jednog reda.

Kreiranje datoteke vrši se sa rewrite:

REWRITE(f);

Upisivanje u datoteku vrši se sa procedurom write ili writeln:

WRITELN(f, lista promenljivih);

or

WRITE(f, lista promenljivih);

Definisanje tipa datoteke:

type

identifikator=file of tip-podatka;

Tip podataka može biti [integer](#), [array \(strukturni tip\)](#) ili [slog](#).

Dodavanje podataka u datoteku

Za dodavanje podataka u datoteci koristi se sledeći oblik:

APPEND(promenljive u datoteci);

Sa append se pozicioniramo na kraj datoteke pa se novi podaci dodaju na kraju.

Brisanje sadržaja datoteke se vrši uz pomoć procedure ERASE, koja ima oblik:

ERASE(promenljive);

56. Napisati program koji upisuje niz celih brojeva u binarnu datoteku, a zatim očitava vrednosti iz te datoteke i ispisuje ih na ekran.

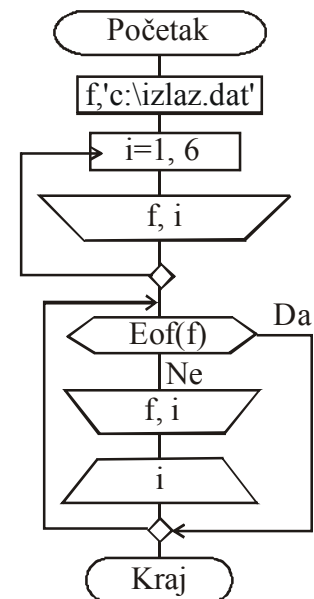
procedura Assign dodeljuje ime spoljasnje datoteke datotecznoj promenljivoj (datotecznom pokazivacu). (Dodeljuje ime spoljašnjeg fajla promenljivoj tipa fajl.)

procedura Reset otvara postojeću datoteku.

procedura Rewrite kreira i otvara novu datoteku.

procedura Close zatvara prethodno otvoreni fajl.

```
program upisivanjeBrojevaUDatoteku;
uses WinCrt;
var
  f: file of Integer;
  i,j: Integer;
begin
  Assign(f,'c:\izlaz.dat');
  Rewrite(f);
  for i := 1 to 6 do
    Write(f,i);
  Writeln('U fajlu su upisani brojevi:');
  Reset(f);
  while not Eof(f) do
  begin
    Read(f,i);
    Writeln(i);
  end;
  Close(f);
end.
```



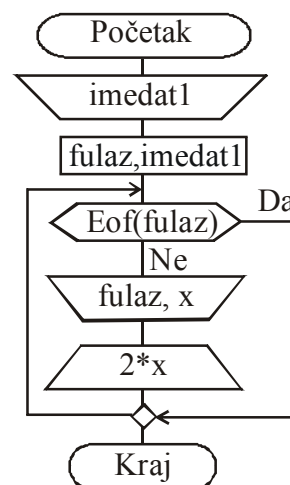
57. Napisati program koji upisuje niz celih brojeva u binarnu datoteku, a zatim očitava vrednosti iz te datoteke i ispisuje ih na ekran.

```

program citanjeBrojevaizDatoteke;
uses WinCrt;
type
  intdat = file of Integer;

var
  fulaz : intdat;
  x : integer;
  imedat1 : string[30];
begin
  write('Upisati ime datoteke: ');
  readln(imedat1);
  assign(fulaz, imedat1);
  reset(fulaz);
  while not Eof(fulaz) do
  begin
    Read(fulaz,x);
    {x:=x*2;}
    Writeln(2*x);
  end;
  close(fulaz)
end.

```

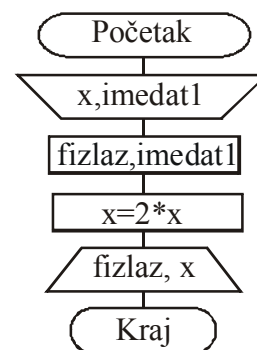


58. Napisati program koji cita broj sa ekrana, mnozi ga sa dva i upisuje rezultat u tekstualnu datoteku.

```

program upisBrojevaUDatoteku;
uses WinCrt;
var
  fizlaz : text;
  x : integer;
  imedat1 : string[30];
begin
  write('Upisati broj: ');
  readln(x);
  write('Upisati ime datoteke: ');
  readln(imedat1);
  assign(fizlaz, imedat1);
  reset(fizlaz);
  rewrite(fizlaz);
  x:=2*x;
  write(fizlaz, x);
  close(fizlaz)
end.

```



59. Napisati program koji kopira sadržaj zadate datoteke u datoteku c:\nova.txt.

```

program kopidatoteke;
uses WinCrt;

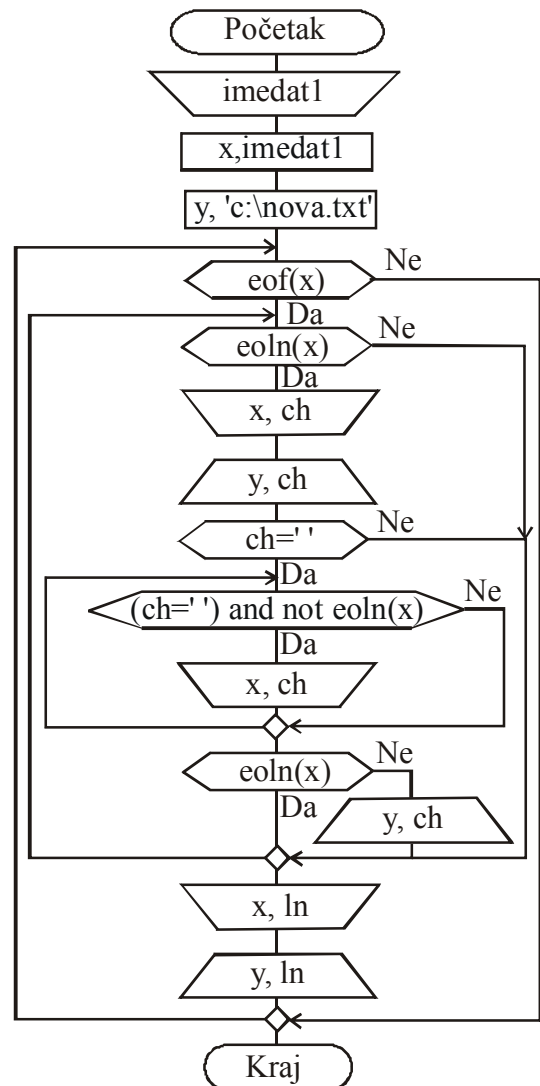
var
  ch:char;
  x,y:text;
  imedat1, imedat2:string[20];

```

```

begin
  write('Upisati ime datoteke (celu putanju): ');
  readln(imedat1);
  assign(x, imedat1);
  assign(y, 'c:\nova.txt');
  reset(x);
  rewrite(y);
  while not eof(x) do
    begin
      while not eoln(x) do
        begin
          read(x,ch);
          write(y,ch);
          if ch=' ' then
            begin
              while(ch=' ') and not eoln(x) do
                read(x, ch);
              if not eoln(x) then write(y, ch)
            end
          end;
          readln(x);
          writeln(y)
        end;
      close(x);
      close(y);
      writeln('Datoteka je iskopirana');
    end.

```



60. Napisati program za unos podataka o automobilima. Na početku **definisati broj** automobila n i podatke o vrsti, godini proizvodnje, ceni i registarskom broju smestiti u datoteku auto.txt.

```

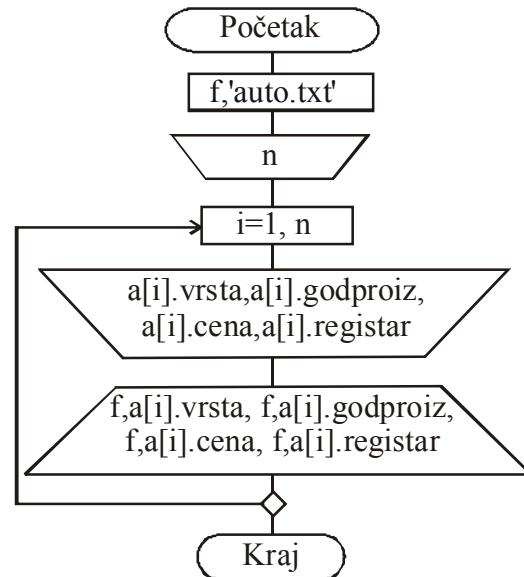
program UnosAuta;
uses wincrt;
type
  auto = record

```

```

vrsta :string[18];
godproiz :integer;
cena :integer;
registar :string[10];
end;
type
  niz=array[1..20] of auto;
var
  a:niz;
  f :text;
  n,i:integer;
begin
  assign(f, 'auto.txt');
  rewrite(f);
  writeln('Uneti broj automobila: ');
  readln(n);
  for i:=1 to n do
  begin
    write('Uneti vrstu automobila: ');
    readln(a[i].vrsta);
    write('Godina proizvodnje: '); readln(a[i].godproiz);
    write('Cena automobila: '); readln(a[i].cena);
    write('Registarski broj: '); readln(a[i].registar);
    { Smestanje informacija u fajl }
    writeln(f, a[i].vrsta);
    writeln(f, a[i].godproiz);
    writeln(f, a[i].cena);
    writeln(f, a[i].registar);
  end;
  close(f)
end.

```

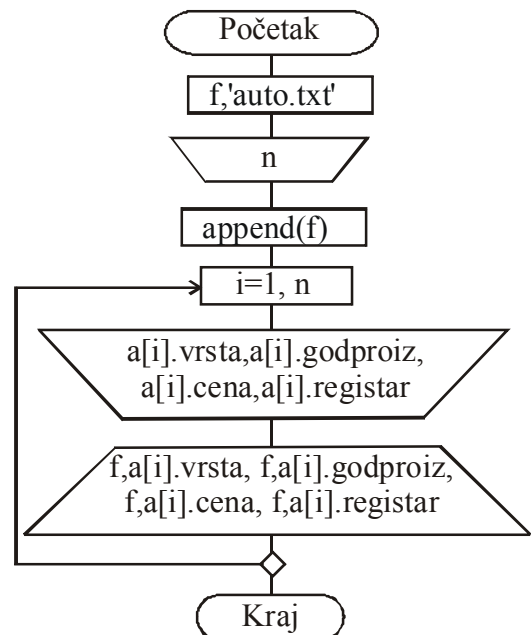


61. Napisati program za dodavanje n automobila u datoteku auto.txt.

```

program DodavanjeAuta;
uses wincrt;
type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cena :integer;
    registar :string[10];
  end;
type
  niz=array[1..20] of auto;
var

```



```

a:niz;
f :text;
n,i:integer;
ime:string;
begin
writeln('Koliko automobila dodajete u datoteku');
readln(n);
writeln('Uneti podatke');
writeln;
assign(f, 'auto.txt');
append(f);
for i:=1 to n do
begin
write('Uneti vrstu automobila: '); readln(a[i].vrsta); writeln(f,a[i].vrsta);
write('Uneti godinu proizvodnje: '); readln(a[i].godproiz); writeln(f,a[i].godproiz);
write('Uneti cenu: '); readln(a[i].cena); writeln(f,a[i].cena);
write('Uneti registarski broj: '); readln(a[i].registar); writeln(f,a[i].registar);
writeln;
end;
close(f);
end.

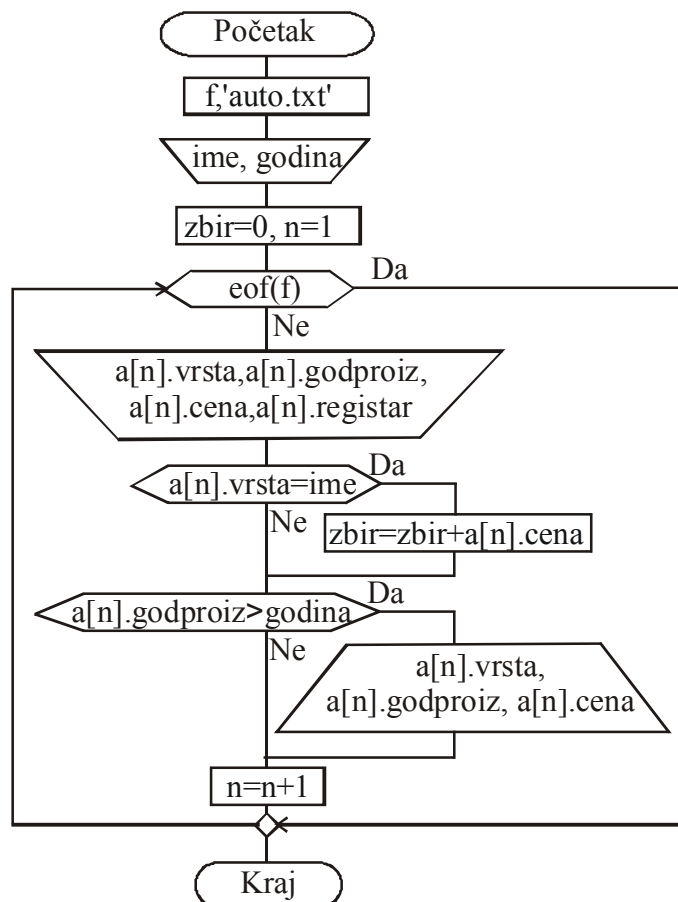
```

62. Iz formirane datoteke auto.txt procitati podatke o automobilima i izracunati ukupnu vrednost svih automobila zadate vrste automobila. Ispisati na ekranu vrstu, godinu proizvodnje i cenu automobila koji su proizvedeni posle zadate godine.

```

program Automobil;
uses wincrt;
type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cena :integer;
    registar :string[10];
  end;
type
  niz=array[1..20] of auto;
var
  a:niz;
  f :text;
  n,i,godina,zbir:integer;
  ime:string;
begin

```



```

writeln('Uneti vrstu automobila ciju ukupnu vrednost trazite');
readln(ime);
writeln('Uneti godinu proizvodnje od koje ce se ispistati vrsta i cena automobila');
readln(godina);
assign(f, 'auto.txt');
reset(f);
zbir:=0;
n:=1;
writeln;
writeln('Automobli proizvedeni posle ',godina,' su: ');
while not eof(f) do
begin
  readln(f,a[n].vrsta);
  readln(f,a[n].godproiz);
  readln(f,a[n].cena);
  readln(f,a[n].registar);
  if a[n].vrsta=ime then
    zbir:=zbir+a[n].cena;
  if a[n].godproiz>godina then
    writeln(a[n].vrsta, ': ',a[n].godproiz, ' god. ', a[n].cena,' din');
  n:=n+1;
end;
writeln;
writeln('Ukupna vrednost svih automobila tipa ',ime, ' je ',zbir);
close(f)
end.

```

63. Iz formirane datoteke auto.txt sortirati podatke o automobilima prema ceni koštanja i ispisati podatke na ekranu.

```

program SortiranjeAutomobila;
uses wincrt;
type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cena :integer;
    registar :string[10];
  end;
type
  niz=array[1..20] of auto;
var
  a:niz;
  f:text;
  n,i,j,pomc, pomg:integer;
  pomv,pomr:string;
begin

```

```

assign(f, 'auto.txt');
reset(f);
n:=1;
while not eof(f) do
begin
  readln(f,a[n].vrsta);
  readln(f,a[n].godproiz);
  readln(f,a[n].cena);
  readln(f,a[n].registar);
  n:=n+1;
end;
for i:=1 to n-1 do
for j:=i+1 to n do
begin
  if (a[i].cena)>(a[j].cena) then
  begin
    pomc:=a[i].cena;
    a[i].cena:=a[j].cena;
    a[j].cena:=pomc;
    pomv:=a[i].vrsta;
    a[i].vrsta:=a[j].vrsta;
    a[j].vrsta:=pomv;
    pomg:=a[i].godproiz;
    a[i].godproiz:=a[j].godproiz;
    a[j].godproiz:=pomg;
    pomr:=a[i].registar;
    a[i].registar:=a[j].registar;
    a[j].registar:=pomr;
  end;
end;
writeln(n);
for i:=1 to n do
begin
  writeln;
  write(a[i].vrsta,' ');
  write(a[i].godproiz,' ');
  write(a[i].cena,' ');
  write(a[i].registar,' ');
end;
close(f)
end.

```

