

prof: Grozda Ćirjanić  
učenik: Lazar Vučković

**Skripta i rešeni zadaci iz programskog jezika**  
**P A S C A L**

## 1. UVOD U PASCAL

Programski jezik Pascal razvio je Niklaus Wirth, 1969/71.godine. Pascal je prvobitno zamišljen kao nastavno sredstvo za učenje veštine strukturiranog programiranja, gde se program može razbiti na manje celine koje se opet logički povezuju u program.

### 1.1 Pravila pisanja programa

Program se sastoji od zaglavlja i bloka naredbi. U zaglavlju se definiše ime programa. Blok se sastoji od deklaracija ili definicija i odeljka naredbi. Odeljak naredbi počinje sa rezervisanom reči **BEGIN**, a završava sa **END**.

**PROGRAM** Ime\_programa (Lista\_datoteka);

Zaglavlje programa

**LABEL** { Deklaracija labela };  
**CONST** { Deklaracije konstanti };  
**TYPE** { Deklaracije tipa };  
**VAR** { Deklaracije promenljivih };

Odeljak deklaracija i definicija. Promenljive, konstante, labele, i tipovi moraju se definisati pre početka izvršnog dela programa.

{Definicije podprograma }

**BEGIN**

naredba1;  
 naredba2;  
 .....;  
 naredban;

Odeljak naredbi ili izvršni deo programa. Naredbe i izrazi u Pascalu se završavaju sa znakom ;

**END.**

### 1.2 IME PROGRAMA

Reč PROGRAM je rezervisana reč u Pascalu kojom počinje svaki program. Iza reči PROGRAM programer definiše ime programa. Imena se u Pascalu nazivaju *identifikatorima*. Ime mora početi slovom abecede. Ne sme imati praznine ili znake interpunkcije i ne može početi brojem. Ime ne može biti ključna reč. Ime programa završava tačka- zarezom. Sledi tabela ključnih reči u Pascalu:

KLJUČNE REČI U PASCALU			
<b>and</b>	<b>end</b>	<b>nil</b>	<b>shr</b>
<b>asm</b>	<b>file</b>	<b>not</b>	<b>string</b>
<b>array</b>	<b>for</b>	<b>object</b>	<b>then</b>
<b>begin</b>	<b>function</b>	<b>of</b>	<b>to</b>
<b>case</b>	<b>goto</b>	<b>or</b>	<b>type</b>
<b>const</b>	<b>if</b>	<b>packed</b>	<b>unit</b>
<b>constructor</b>	<b>implementation</b>	<b>procedure</b>	<b>until</b>
<b>destructor</b>	<b>in</b>	<b>program</b>	<b>uses</b>
<b>div</b>	<b>inline</b>	<b>record</b>	<b>var</b>
<b>do</b>	<b>interface</b>	<b>repeat</b>	<b>while</b>
<b>downto</b>	<b>label</b>	<b>set</b>	<b>with</b>
<b>else</b>	<b>mod</b>	<b>shl</b>	<b>xor</b>

Tabela1: Ključne reči

Imena u Pascalu mogu biti proizvoljne dužine, međutim prevodilac uzima u obzir identifikatore dužine do osam znakova. Ime može imati više od osam znakova, ali prevodilac te znakovce ignoriše.

pravilno napisana imena	nepravilno napisana imena
aritmetika	aritmetika dva
prvi drugi	prvi drugi
max1	1max

**SAVET:** Prilikom definisanja identifikatora, kako za ime programa tako i za promenljive, konstante, procedure i funkcije preporučuje se izbor preglednih imena u skladu sa pojmom koji predstavljaju.

## 1.2 KOMENTARI

Komentari se sastoje od teksta za koji se odlučuje programer. Komentari su namenjeni korisnicima radi boljeg razumevanja funkcije programa, dok računar komentar prilikom prevođenja programa ignoriše. Komentari se pišu u vitičastim zagradama :

{program izračunava površinu kruga} ili  
(\*program izračunava površinu kruga\*)

Komentar se može postaviti bilo gde u programu, ali se obično postavlja ispod ili u liniji koda koju objašnjava.

## 1.3 TIPOVI PODATAKA U PASCALU

U svakom programu potrebno je deklarisati tipove podataka sa kojima se radi. Razlikujemo celobrojni tip podatka, realni tip, znakovni tip i logički tip podatka.

### 1) CELOBROJNI TIP PODATKA

*Celobrojni tip podatka:*

Tip podatka	Rezervisana reč	Najmanja vrednost	Najveća vrednost
celi brojevi (samo +)	byte	0	255
celi brojevi (-i +)	shortint	-128	127
celi brojevi (-i +)	integer	-32768	+32768
celi brojevi (samo +)	word	0	65535
celi brojevi (-i +)	longint	-2147483648	+2147483647

Promenljive u Pascalu imaju slične su promenljivim u matematici. Sve promenljive koje se koriste u programu MORAJU biti deklarisanе. Karakteristika promenljive je da se njena vrednost može menjati tokom izvršenja programa (zbog čega se u izvornom kôdu predstavljaju ne svojim iznosom već simboličkom oznakom, *imenom promenljive* ili indentifikatorom)

Naredba za deklaraciju promenljivih je VAR, promenljiva ima svoje ime i tip podatka kome pripada. Definisanjem tipa podatka kome promenljiva pripada definišu se i operacije koje se mogu izvršavati nad promenljivim

Pravilo pisanja:

```
var ime_promenljive1, ime_promenljive2:tip_podatka;
```

**Primer 1:** Da bi se promenljivu a deklarisanala kao celobrojna koristi se naredba:

```
var a: integer;
```

**Var** je naredba za deklarisanje promenljivih, **a** je promenljiva, **integer** ključna reč (standardni identifikator) koja označava celobrojni tip podatka

Operacije karakteristične za celobrojni tip podataka su:

- 1) sabiranje (+)
- 2) oduzimanje (-)
- 3) množenje (\*)
- 4) div (celobrojno deljenje)
- 5) mod (ostatak celobrojnog deljenja)

**Primer 2:** Ako je  $x=15$  i  $y=4$  i ako je

$z := x \text{ DIV } y$	$z := 15 \text{ DIV } 4$
$j := x \text{ MOD } y$	$j := 15 \text{ MOD } 4$

Rezultat je:  $z=3$ ,  $j=3$ .

## 2) REALNI TIP PODATAKA

Vrednosti koje sadrže ceo i decimalni deo nazivaju se realnim.

Na primer, realne konstante su: **7.23**      **5.88**      **-9.10**      **40.555**

Realna konstanta u Pascalu ne sme počinjati ili završavati decimalnom tačkom, na primer:

**652.**      **.210**

Ispravno je napisati: **652.0**      **0.210**

Realni brojevi mogu da se izraze u

- a) eksponencijalnom zapisu
- b) pozicionom ili formatiranom zapisu

Eksponencijalni zapis ili eksponenciojalna notacija koristi se prilikom ispisa veoma velikih ili veoma malih vrednosti. Na ovaj način izbegava se pisanje velikog broja nula. ( 3000000000.0 ili 0.0000000003)

Primer:

$$3.5E + 5 = 3.5 * 10^5$$

E – oznaka za eksponencijalni oblik, ispred se piše mantisa a iza E eksponent. Opšti oblik pisanja je:

$$\pm m E \pm ex$$

Eksponent pokazuje za koliko mesta treba decimalnu tačku pomeriti u levo ili desno. Ako je eksponent pozitivan tačka se pomera udesno, a ako je negativan tačka se pomera ulevo.

Primer:

$$231E - 1 = 231.0E - 1 = 23.1$$

$$1E2 = 1.0E2 = 100.0$$

### Realni tipovi

Tip podatka	Rezervisana reč	Najmanja vrednost	Najveća vrednost
realni brojevi	real	2.9E-39	1.7E38
	single	1.5E-45	3.4E38
	double	5.0E-324	1.7E308
	extended	3.4E-4932	1.1E4932

Operacije nad realnim tipom podatka su:

- 1) sabiranje (+)
- 2) oduzimanje (-)
- 3) množenje (\*)
- 4) deljenje (/)

**Primer deklarisanja promenljive x kao realne:**

**var x: real;**

Standardni identifikator je **real**.

### 3) ZNAKOVNI TIP PODATAKA

Ovu grupu podataka čine znakovi: slova abecede, numerički znakovi od 0 do 9, znakovi interpunkcija i specijalni znakovi.

*Znakovna konstanta* sastoji se od znaka ograđenog jednostrukim navodnicima, na primer:

'B' 'B' 'Z' '#' '@' '10'

Jednostruki navodnici su neophodni da bi računar znao, na primer, da \* PREDSTAVLJA znakovnu konstantu za razliku od oznake množenja, ili da je 10 znakovna konstanta a ne celobrojni podatak 10.

Više znakova kao što su reči ili rečenice nazivaju se *nizovi* (engl. string). U programu se niz znakova može koristiti kao konstanta s time da se niz znakova omeđi navodnicima.

Primer: ' Programski jezik PASCAL' ili ' Unesi x'

Standardni identifikator za znakovni tip je **char**.

Primer deklarisanja promenljive Y znakovnog tipa:

**var Y: char ;**

### 4) LOGIČKI TIP PODATKA

Računar u radu proverava različite uslove i u skladu sa rezultatom ispitivanja nastavlja rad. U tu svrhu se upotrebljava tip podataka koji ima dve vrednosti

False	True
Laž	istina

Ove vrednosti zovu se logičke vrednosti, a u Pascalu ih nazivaju Booleove vrednosti, po engleskom matematičaru Georgu BOOLE, koji je razvio logičku algebru u 19. veku. U računaru se Booleove vrednosti prikazuju jednim bitom, koji je 0 za laž, a 1 za istinito. Postoje tri osnovne operacije nad logičkim tipom podataka:

- 1) **Negacija (not)**
- 2) **Disjunkcija (or)**
- 3) **Konjukcija (and)**

Identifikator kojim se označava logički tip je **boolean**.

Primer deklarisanja logičke promenljive Y :

**var Y: boolean;**

### KONSTANTE U PASCALU

Konstante su podaci čija je osnovna karakteristika da se njihova vrednost ne sme menjati.

Konstante se definišu u odeljku za definisanje konstanti. Naredba za definisanje je CONST.

Pravilo pisanja:

**CONST ime\_konstante= vrednost;**

Primer:

**const e=2.71;**  
**const pi=3.14159;**  
**const b='';**  
**const x=56.23;**

Ime konstante (e i pi) se koristi u programu umesto konstante, time se povećava preglednost programa, i olakšava promena vrednosti konstante.

## UNOS PODATAKA U PASCALU – ULAZNE NAREDBE

U Pascalu postoje dve naredbe za unos podataka : **READ** i **READLN**. Učitavanje može biti preko ulazne datoteke ili preko tastature. Prilikom unosa promenljive treba voditi računa da unesene vrednosti odgovaraju definisanom tipu podataka za te promenljive. Opšte značenje naredbe je prekid izvršavanja programa i čekanje da se unesu vrednosti navedenih promenljivih.

Pravilo pisanja je:

**READ(ime promenljive, ime promenljive);**

**Primer:** Ako želimo uneti vrednosti za stranice pravougaonika – promenljive a i b, u programu pišemo:

**READ (a,b);** učitava promenljive a i b

**READLN(a,b);** učitava promenljive a i b i prelazi u naredni red

U naredbi READ podaci se učitavaju po redu kako su u naredbi navedeni.

Osnovna razlika između ovih naredbi je što naredba READLN učitava vrednosti kako su specificirane promenljivama, a nakon toga prelazi na sledeći red, tj. zanemaruju se svi ostali podaci u tekućem redu.

## NAREDBA ZA ISPIS NA EKLAN

Naredba za ispis omogućava prikazivanje teksta ili vrednosti promenljive ili rezultata obrade na ekran.

Naredba je:

**WRITE (ime promenljive ili tekst u polunavodnicima);**

**WRITELN(ime promenljive ili tekst u polunavodnicima);**

Ova naredba ispisuje vrednost koja je napisana u zagradi iza reči WRITE odnosno WRITELN.

Ako želite da se preskoči red pri ispisu, piše se samo WRITELN, na primer:

**WRITELN;**

- A) Ispis teksta **WRITELN('Pascal')**<sup>1</sup>;  
Na ekranu dobijamo tekst Pascal
- B) Ispis vrednosti promenljive **WRITELN(O,P);**  
Ispisuje vrednost promenljive O i P
- C) Kombinovan ispis teksta i promenljive **WRITELN('Zbir je:',a+b);**  
Ispisuje tekst Zbir je i vrednost izraza a+b
- D) Formatirani ispis – koristi se kod realnih brojeva

**Primer za formatirani ispis:** Vrednost promenljive  $x= 3,1485693$  a želimo ispis samo na dve decimale

**WRITE(x:5:2);**

za ispis promenljive x rezervisano je **ukupno 5 mesta**, od toga su **dva za decimalni deo** (ispisuje 3.14)

	3	.	1	4
--	---	---	---	---

E) Ispis celih brojeva

Primer:

**WRITELN(14:8);** Ispisuje se vrednost 14 a za ispis je potrebno 8 mesta.

							1	4
--	--	--	--	--	--	--	---	---

<sup>1</sup> Tekst se OBAVEZNO smešta u polunavodnike

## NAREDBE DODELE ILI PRIDRUŽIVANJE VREDNOSTI

Promenljive u memoriji imaju neku slučajnu vrednost. Naredba dodele ili pridruživanja omogućava da promenljiva dobije novu vrednost koja se smešta u memorijsku lokaciju i postaje vrednost odgovarajuće promenljive.

Operator pridruživanja u Pascalu je **:=**, a naredba pridruživanja je oblika:

**promenljiva := izraz;**

Kada računar izvrši naredbu pridruživanja izraz na desnoj strani postaje vrednost promenljive na levoj strani.

Operator **:=** se može čitati kao "postaje".

Primeri:

X:=15; čitaj X postaje 15

B:=10; čitaj B postaje 10

X:=X\*B; čitaj X pomnoži sa B, rezultat sačuvaj u X

**VAŽNO.** Izraz koji se nalazi sa desne strane znaka za pridruživanje izračuna se a zatim dodeli promenljivoj sa leve strane.

Vrednost pridružena promenljivoj mora biti istog tipa kao što je određeno za promenljivu u deklaraciji promenljive.

Ovo pravilo ima jedan izuzetak. Za svaku celobrojnu vrednost postoji odgovarajuća realna vrednost. Na primer, za celobrojnu vrednost 13 odgovarajuća realna vrednost je 13.0; za 210 je 210.0 ...

Pascal dozvoljava označavanje celobrojnih vrednosti realnim promenljivama. Nakon pridruživanja celobrojna vrednost se pretvara u adekvatnu realnu.

a:=13; postaje nakon pridruživanja 13.0 ili

c:=5555; postaje 5555.0

**VAŽNO** Realnoj vrednosti ne može se pridružiti celobrojna vrednost

## LINIJSKA STRUKTURA

1. **Zadatak** Napisati program koji definiše dve konstante nad njima sprovodi operacije karakteristične za:

- celobrojne podatke
- realne

Rešenje pod a:		Rešenje pod b:
Program konstante1; const a=5, b=9; begin writeln(a+b); writeln(a-b); writeln(a*b); writeln(a div b); writeln(a mod b); end.	konstante a i b su definisane na početku – dodeljene su im vrednosti (5 i 9) tako da dalje u programu koristimo njihova imena (a i b). a i b se ponovo ne učitavaju	Program konstante2; const a=5.23, b=9.00; begin writeln(a+b); writeln(a-b); writeln(a*b); writeln(a/b); end.

2. **Zadatak:** Napisati program koji vrednost u eurima pretvara u dinare.

**Rešenje:**

program pretvaranje; var dinari, eura, kurs:real; begin	Potrebne su tri promenljive. Dve promenljive (kurs i eure) unosimo a treću (dinare) izračunavamo u naredbi dodele.
---	--

```
writeln('Unesi kurs eura:->');
readln(kurs);
writeln('Unesi iznos u eurima:->');
readln(eura);
dinari:=eura*kurs;
writeln('Iznos u dinarima je',dinari:5:2);
readln;
end.
```

**3. Zadatak:** Putovanje između mesta M i N traje n sekundi. Izračunaj koliko je to sati, minuta i sekundi.

**Rešenje:**

```
program sekunde;
var n,sat,min,sek : integer;
begin
  Write('Unesi sekunde :-> ');
  Readln (n);
  sat := n div 3600;
  sek := n mod 3600;
  min := sek div 60;
  sek := sek mod 60;
  Writeln('To je =',sat:3,'sati',min:6,' ',',',minuta,' ',',',sek:4,' ',',',sekundi);
end.
```

3888 sekundi je 1 sat, 4 minuta i 48 sekundi...

**4. Zadatak:** Napisati program kojim se određuje vreme prizemljenja aviona koji je uzleteo u a sati, b minuta, c sekundi, i u letu proveo d sekundi.

**Rešenje:**

```
program avion;
var sati,minuti,sekunde,vreme:integer;
    satpriz, minutpriz, sekundpriz:longint;
begin
  write('Unesi sate,minute,sekunde');
  readln(sati,minuti,sekunde);
  write('Duzina leta u sekundama');
  readln(vreme);
  sekundpriz:=sekunde+vreme;
  minutpriz:=minuti+ sekundpriz div 60;
  sekundpriz:= sekundpriz mod 60;
  satpriz:=sati+ minutpriz div 60;
  minutpriz:= minutpriz mod 60;
  satpriz:=satpriz mod 24;
  writeln('Vreme prizemljenja',satpriz,'sati',minutpriz,'minuta',sekundpriz,'sekundi');
end.
```

**5. Zadatak** Date su dve tačke u koordinatnom sistemu, zadate svojim koordinatama. Izračunati njihovo rastojanje od koordinatnog početka redom.

**Rešenje:**

```
program rastojanje;
var x1,x2,y1,y2,d1,d2:real;
{x1,y1 koordinate prve tacke}
```



```
{x2, y2 koordinate druge tacke}
{d1, d2rastojanje tacaka od koordinatnog pocetka}
```

```
begin
```

```
writeln('Unesi koordinate prve tacke');
```

```
readln(x1,y1);
```

```
writeln('Unesi koordinate druge tacke');
```

```
readln(x2,y2);
```

```
d1:=sqrt(sqr(x1)+sqr(y1));
```

```
d2:=sqrt(sqr(x2)+sqr(y2));
```

Formula po kojoj se izracunava rastojanje od koordinatnog pocetka do tacke.

```
writeln('Rastojanje izmedju prve tacke i koordinatnog pocetka',d1:8:3);
```

```
writeln('Rastojanje izmedju druge tacke i koordinatnog pocetka',d2:6:4);
```

```
end.
```

### **6. Zadatak**

Napisati program kojim ce se izracunati udaljenost dve tacke u ravni. Poznate su koordinate tih tacaka T1(x1, y1) i T2(x2, y2).

Formula glasi:  $d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

#### **Rešenje:**

```
program rastojanje;
```

```
var x1,x2,y1,y2,d:real;
```

```
{ x1,y1 koordinate prve tacke}
```

```
{ x2, y2 koordinate druge tacke}
```

```
{ d rastojanje}
```

```
begin
```

```
writeln('Unesi koordinate prve tacke');
```

```
readln(x1,y1);
```

```
writeln('Unesi koordinate druge tacke');
```

```
d:=sqrt(sqr(x2-x1) + sqr(y2-y1));
```

```
writeln('Rastojanje izmedju tacaka je:', d:5:3);
```

```
readln;
```

```
end.
```

**7.Zadatak** . Date su tri tacke u koordinatnom sistemu i zadate svojim koordinatama. Izracunati obim i površinu trougla čija su one temena.

#### **Rešenje:**

```
program trougao;
```

```
var x1, x2, y1, y2, x3, y3, a, b, c, o, p:real;
```

```
{x1,y1 koordinate prve tacke}
```

```
{x2,y2 koordinate prve tacke}
```

```
{x3,y3 koordinate prve tacke}
```

```
{a,b,c dužine stranica trougla}
```

```
{o obim trougla, p je površina trougla}
```

```
begin
```

```
writeln('Unesi koordinate x1 i y1');
```

```
readln(x1,y1);
```

```
writeln('Unesi koordinate x2 i y2');
```

```
readln(x1,y1);
```

```
writeln('Unesi koordinate x3 i y3');
```

```
readln(x3,y3);
```

```
a:=sqrt(sqr(x1-x2)+sqr(y1-y2));
```

```

b:= sqrt(sqr(x1-x3)+sqr(y1-y3));
c:= sqrt(sqr(x3-x2)+sqr(y3-y2));
o:=a+b+c;
p:=sqrt((o/2)*((o/2)-a)*((o/2)-b)*((o/2)-c));
writeln('Obim iznosi',o:5:3);
writeln('Povrsina iznosi',p:4:3);
end.

```

**8. Zadatak:**

Napisati program koji za uneseni napon  $U$  i struju  $I$  sa tastature izračunava otpor i snaga kola. Rezultat prikazati u formatu od 6 mesta sa 3 decimale.

**Rešenje:**

```

program snaga;
var u,i,r,p : real;
begin
  Write('Unesi napon U kola : ');
  ReadLn(u);
  Write('Unesi struju I kola : ');
  ReadLn(i);
  r := u/i;
  p := u*i;
  WriteLn('Otpor kola je : ',r:6:2,' Snaga kola je : ',p:6:3);
  ReadLn
End.

```

**9. Zadatak:**

Neka je data jednačina  $ax + b = 0$ . Promenljive  $a$  i  $b$  se unose sa tastature. Napiši program koji za svaku vrednost  $a$  i  $b$  daje rešenja jednačine. Rezultat prikaži u formatu od 7 mesta sa 3 decimale.

**Rešenje:**

Rešenje je jednačine  $x = -b/a$

```

program prog;
var a,b,x : real;
begin
  Write('Unesi parametar a : ');
  ReadLn(a);
  Write('Unesi parametar b : ');
  ReadLn(b);
  x := -b/a;
  Writeln('Resenje jednacine je x=',x:7:3);
End.

```

**10 Zadatak:** .Ako se stranice  $a$  i  $b$  i visina  $h$  jednakokrakog trapezu unose, izracunati njegovu površinu.

**Rešenje:**

```

program trapez;
var a,b,h,p:real;
begin
  writeln('Unesi a');
  readln(a);
  writeln('Unesi b');

```

```

readln(b);
writeln('Unesi h');
readln(h);
p:=(b+a)/2*h;
writeln(p:5:2)
end.

```

**11 Zadatak:** Napisati program koji unetu velicinu u metrima pretvara u centimetre.

**Rešenje:**

```

program pretvaranje;
var metri, centimetri:real;
begin
writeln('Unesi metre koje pretvaras');
readln(metri);
centimetri:=metri*100;
writeln(centimetri:4:2);
end.

```

**12 Zadatak:** Ako se unose duzine stranice a i hipotenuze c pravouglog trougla, izracunati površinu tog trougla.

**Rešenje:**

```

program trougao;
var a, b,c,p:real;
begin
writeln('Unesi katetu a i hipotenuzu c');
readln(a,c);
b:=sqrt(sqr(c)-sqr(a));
p:=a*b/2;
writeln('Povrsina je',p:4:1);
end.

```

**13 Zadatak** Ako se unosi fajl određenog kapaciteta u Mb, napisati program koji izračunava slobodan prostor na disketi(standardna 3<sup>1/2</sup> inčna disketa). Rezultat prikazuje u kilobajtima.

**Rešenje:**

```

program kilobajti;
{Program računa slobodan prostor na disketi}
var slobodno,zauzeto:real;
const kapacitet=1.44 ;
begin
    writeln('Unesi kapacitet fajla u megabajtima');
    readln(zauzeto);
    slobodno:=kapacitet-zauzeto;
    writeln('Slobodan prostor je:',slobodno:6:2);
    writeln('To je u kilobajtima:',slobodno*1024:8:3,'KB');
end.

```

```

8 bita= 1 Bajt
1024 Bajta=1 KB
1024 KB=MB
1024 MB=GB

```

**14 Zadatak:** napisati program koji računa i prikazuje ceo i decimalan deo realnog broja.

**Rešenje:**

```

program celi_decimalni;
var x,d:real;

```

```

c:integer;
begin
    writeln('Unesi realan broj ->:');
    readln(x);
    c:=trunc(x); {izračunavanje celog dela broja}
    d:=x-c; { izračunavanje decimalnog dela broja}
    writeln('Celi deo broja je',c);
    writeln('Decimalni deo je:',d:4:2);
readln;
end.

```

**15 Zadatak** napisati program koji na osnovu unete vrednosti kredita i kamate, računa puni iznos plaćanja.

**Rešenje:**

```

program kredit; {program računa kamatu i ukupan iznos}
{ovaj program racuna kamatu i total u kreditu}
var iznos,kamata,dcmlkam,iznkam,total:real;
begin
    read(iznos,kamata);
    dcmlkam:=kamata/100.0;
    iznkam:=iznos*dcmlkam;
    total:=iznos+iznkam;
    writeln('Iznos kredita:',iznos:8:2);
    writeln('Kamata:',kamata:4:1);
    writeln('Iznos kamata',iznkam:8:2);
    writeln('Puni iznos placanja:',total:8:2);
end.

```

**16 Zadatak** Napisati program koji unetu temperaturu u Celzijima pretvara u Farenhajte.

**Rešenje:**

```

program cel_Farenhajt;
{ Program unetu temperaturu u Celzijima pretvara u Farenhajte}
var tempC, tempF: real;
begin
    writeln('Unesi temperaturu u Celzijusima');
    readln(tempC);
    tempF:=tempC*1.8+32;
    writeln('Temperatura po Forenhajtu', tempF:8:2)
end.

```

**17 Zadatak:** Napisti program koji na osnovu unetog kapaciteta rezervoara i pređenog puta, izračunava potrošnju goriva na 100km.

**Rešenje:**

```

program potrosnja_goriva; {Program računa potrošnju goriva}
var V, S,P:real;
begin
    writeln('Unesi kapacitet rezervoara');
    readln(V);
    writeln('Unesi predjeni put');
    readln(S);
    P:=V*100/S;

```

writeln('Potrosnja na 100 km je', P:5:3);  
**end.**

## STANDARDNE FUNKCIJE U PASCALU

Oznaka u pascalu	NAZIV	oznaka u matematici	objašnjenje
<b>abs (x)</b>	Apsolutna vrednost	$ x $	Izračunava apsolutnu vrednost stvarnog parametra, na primer, abs (8) je 8, abs (- 8) je 8.
<b>sqr (x)</b>	Kvadrat	$x^2$	Izračunava kvadrat stvarnog parametra, na primer, sgr (4) je 12.
<b>sqrt (x)</b>	Koren	$\sqrt{x}$	Izračunava kvadratni koren stvarnog parametra, na primer, sqrt (16) je 4.0, sqrt (2.25) je 1,5.
<b>trunc (x)</b>	Odsecanje		Pretvara realni broj u celobrojni zanemarujući decimalni deo, na primer, trunc (5.25) je 5.
<b>round (x)</b>	Zaokruživanje		Pretvara realni broj u celobrojni zaokružujući realnu vrednost na celobrojnu, na primer, round (3.25) je 3, round (3.5) je 4.
<b>sin (x)</b>	Sinus	$\sin x$	Izračunava trigonometrijsku funkciju, a stvarni parametar mora biti u radijanima.
<b>cos (x)</b>	Cosinus	$\cos x$	
<b>arctan (x)</b>	Arkus tangens	$\arctg x$	
<b>Ln (x)</b>	Prirodni logaritam	$\ln x$	Izračunava prirodni logaritam stvarnog parametra, na primer, ln (10) je 2,3.
<b>exp (x)</b>	Eksponencijalna	$e^x$	Izračunava eksponencijalnu funkciju čiji je eksponent x; na primer exp (2) je 7,34

**10. Zadatak** Na osnovu stranice jednakostraničnog trougla izračunati njegovu površinu i visinu

**Rešenje:**

```

program trougao;
var a, pov, visina :REAL;
begin
  writeln ('Upisi stranicu jednakostranicnog trougla --> ');
  readln (a);
  pov:= (sqr (a) * sqrt (3)) / 2;
  visina:= (sqr (a) * sqrt (3)) / 4;
  writeln ('Povrsina trougla je --> ', pov :6 :2);
  writeln ('Visina trougla je --> ', visina :6 :2);
end.

```

**11. Zadatak** Definisati konstantu x, zatim izračunati kvadrat, koren, apsolutnu vrednost, prirodni logaritam.

**Rešenje:**

```

program standard;
const x=25.0;

```

```

var kv,ko,aps,prirodni:real;
begin
  aps:=abs(x);
  prirodni:=ln(x);
  kv:=sqr(x);
  ko:=sqrt(x);
  writeln('apsolutna=',aps);
  writeln('kvadrat=',kv:4:2);
  writeln('koren=',ko:4:2);
  writeln('Prirodni logaritam je =',prirodni:5:3);
end.

```

**12. Zadatak.** Alfa je ugao u stepenima. Izračunati sinus i kosinus ugla.

**Rešenje:**

```

program trigono;
var alfa,beta,s,c:real;
{alfa – ugao u stepenima}
{beta – ugao u radijanima}
{s , c –sinus i kosinus ugla}
const pi=3.14159;
begin
  write('Ugao u stepenima');
  readln(alfa);
  beta:=pi/180*alfa;
  s:=sin(beta);
  c:=cos(beta);
  writeln('sin(',alfa:3:1,')=',s:1:2);
  writeln('cos(',alfa:3:1,')=',c:1:2);
  readln;
end.

```

Ugao pretvaramo u radijane.

## Razgranata struktura programa

### USLOVNE NAREDBE

Jednostruki izbor podrazumeva da računar prvo proverava vrednost bulovog izraza, a zatim izvršava određenu naredbu. Ako je izraz istinit naredba se izvršava, a ako nije naredba se ne izvršava, već se izvodi naredna naredba u programu.

U pascalu jednostruki izbor je predstavljen pomoću IF THEN naredbe, koja je oblika:

a) **IF logički izraz THEN**

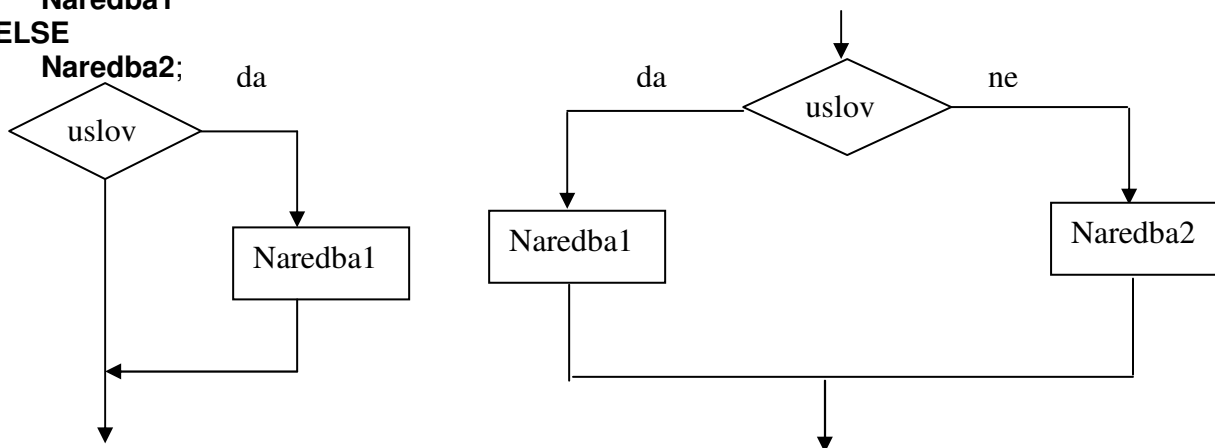
Naredba;

b) **IF logički izraz THEN**

Naredba1

**ELSE**

Naredba2;



**1 Zadatak:** Napisati niz naredbi kojima se sabiraju samo pozitivne vrednosti promenljivih a,b,c.

**Rešenje:**

```

S:=0;
If a>0 then s:=s+a;
If B>0 then s:=s+b;
If c>0 then s:=s+c;
  
```

**2 Zadatak:**

Napiši naredbu uslovnog prelaska kojom se za zadato  $x$  izračunava  $f$  po formuli:

$$f = \begin{cases} 8x, & \text{za } x \leq -1 \\ \sqrt{x^2 + 4x}, & \text{za } -1 < x < 4 \\ |x|, & \text{za } x \geq 4 \end{cases}$$

**Rešenje:**

```

if x <= 1 then f:=8*x
  else if x > -1 and x < 4 then f:=sqrt(sqr(x)+4*x)
    else f:=abs(x);
  
```

**3 Zadatak:**

Koju će vrednost imati promenljiva  $a$  nakon izvršavanja sledećih naredbi:

**a = 1.56;**

**if a < 1 then a = a+1**

**else begin a = a+4; a:=trunc(a)+7; a:=sqr(a)+10 end;**

**a:=abs(a)+2;**



**Rešenje:**

Budući da je  $a$  veće od 1 to će se izvršiti instrukcija od `if` koje su pod `else` pa imamo:

```
a=a+4=1.56+4=5.56
a=trunc(a)+7=trunc(5.56)+7=5+7=12
s=sqr(a)+10=12*12+10=144+10=154
na kraju dobijemo:
a=abs(a)+2=sbs(154)+2=154+2=156
```

**4 Zadatak:** Uneti zaradu radnika, ako je zarada veća od 15000 hiljada dinara, prikazati je, ako ne prikazati tekst 'Zarada je manja od 15000'

**Rešenje:**

```
program zarada1;
var zarada:real;
begin
writeln('Unesi zaradu');
readln(zarada);
if zarada>15000 then
writeln(zarada:5:2)
else
writeln('Zarada je manja od 15000');
end.
```

**5 Zadatak:** Uneti dva broja a zatim ispisati manji pa veći broj.

**Rešenje:**

```
Program brojevi;
Var x,y:integere;
Begin
    Writeln('Unesi dva broja');
    Readln(x,y);
    If x<y then write(x,y) else write (y,x);
End.
```

**6 Zadatak:** Napisati program koji, od uneta dva broja, veći deli sa 3, a manji množi sa 4. Rezultat napisati u formativnom zapisu.

**Rešenje:**

```
program brojevi;
var broja,brojb:integer;
begin
    writeln('Unesi dva broja');
    readln(broja,brojb);
    if broja>brojb then Writeln(broja/3,brojb*4)
    else
        writeln(broja*4,brojb/3);
end.
```

Prvo se proverava da li je **broja**>**brojb**, ako jeste **broja** se deli sa 3, a **brojb** množi sa 4. Ako je **brojb**>**broja**, onda se **brojb** deli sa 3, a **broja** množi sa 4

**7 Zadatak:** Napisati program koji proverava da li je uneti broj deljiv sa 2.

**Rešenje:**

```
Program deljenje;
var broj, rezultat :integer;
begin
writeln('Unesi broj koji ispituješ');
readln(broj);
rezultat:=broj mod 2;
if rezultat=0 then
writeln('Broj je deljiv sa dva', rezultat:3)
else
writeln('Broj nije deljiv sa dva');
end.
```

rezultat:=broj mod 2; vrednost koja se dobije prilikom deljenja smešta se u promenljivu rezultat. Ako je vrednost promenljive rezultat jednaka nuli – broj je deljiv sa 2, u protivnom nije.

**8 Zadatak:** Napisati program koji proverava da li uneta datoteka L izrazena u Kb može da stane na disketu.

**Rešenje:**

```
program disketa;
const kapacitetdis=1474.56;
var datoteka:real;
begin
writeln('Unesi kapacitet datoteke');
readln(datoteka);
if datoteka<=kapacitetdis then writeln('Datoteka se može smestiti na disketu')
else
writeln('Kapacitet datoteke je veći od kapaciteta diskete');
end.
```

**9 Zadatak:**

Neka se sa tastature unose celi brojevi  $a_1$ ,  $a_2$ , i  $a_3$ . Napiši program kojim se izračunava zbir brojeva koji se nalaze u intervalu 5 do 9. Rezultat prikaži u formatu od 3 mesta.

**Rešenje:**

```
program sumabrojeva;
var a1,a2,a3,suma : integer;
begin
  Writeln('Unesi a1 : ');
  Readln(a1);
  Writeln('Unesi a2 : ');
  Readln(a2);
  Writeln('Unesi a3 : ');
  Readln(a3);
  suma := 0;
  if ((a1 >=5) and (a1 <= 9)) then suma := suma+a1;
  if ((a2 >=5) and (a2 <= 9)) then suma := suma+a2;
  if ((a3 >=5) and (a3 <= 9)) then suma := suma+a3;
  Writeln('Suma brojeva koji se nalaze u intervalu od 5 do 9 je:',suma:3);
end.
```

**10 Zadatak:**

Neka se sa tastature unose brojevi  $a_1$ ,  $a_2$ , i  $a_3$ . Napiši program kojim se nalazi najveći od unesenih brojeva. Rezultat prikaži u formatu od 6 mesta sa 3 decimalna.

**Rešenje:**

```

program najveći;
var  n1,n2,n3,max : real;
begin
  Writeln('Unesi a1 : ');
  Readln(a1);
  Writeln('Unesi a2 : ');
  Readln(a2);
  Writeln('Unesi a3 : ');
  Readln(a3);
  max := a1;
  if (a2 >= max) then max := a2;
  if (a3 >= max) then max := a3;
  Writeln('Najveći uneseni broj je:',max:6:3);
end.

```

**11 Zadatak:**

Neka se sa tastature brojevi  $a_1$ ,  $a_2$ , i  $a_3$ . Napiši program kojim se ovi brojevi udvajaju ako je  $a_1 \geq a_2 \geq a_3$  inače se svakom broju dodeljuje njegova apsolutna vrednost.

**Rešenje:**

```

program udvajanje;
var
  a1,a2,a3 : real;
begin
  Writeln('Unesi a1 : ');
  Readln(a1);
  Writeln('Unesi a2 : ');
  Readln(a2);
  Writeln('Unesi a3 : ');
  Readln(a3);
  if ((a1 >= a2) and (a2 >= a3 )) then
  begin
    a1 := 2*a1;
    a2 := 2*a2;
    a3 := 2*a3;
  end
  else
  begin
    a1 := abs(a1);
    a2 := abs(a2);
    a3 := abs(a3)
  end;
  Writeln('a1 = ',a1:6:2, ' a2 = ',a2:6:2,' a3 = ',a3:6:2);
end.

```

## CASE naredba

Selekcija jedne naredbe iz skupa naredbi, na osnovu vrednosti izraza može se izvršiti pomoću CASE naredbe. Sintaksa CASE naredbe je sledeća:

**CASE c OF**

1:S1;

2:S2:

3:S3:

.....

n:SN

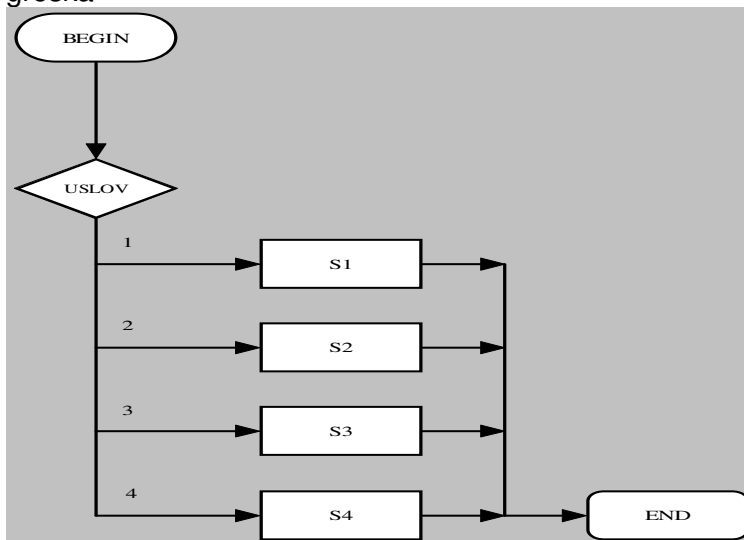
end

Izraz iza rezervisane reči CASE naziva se selektor. Vrednost selektora može biti celobrojnog, znakovnog ili Booleovog tipa. Ne sme biti realnog tipa

Case naredba izvršava se na sledeći način:

- 1) Prvo se izračunava vrednost selektora a zatim
- 2) izvršava naredba čije je obeležje jednako vrednosti selektora

Vrednost c određuje koja će se naredba izvršiti. Ako je  $c=1$  izvršiće se S1, ako je  $c=2$  S2 itd. U ovom primeru mora vrednost c biti u nizu od 1 do 4. Ako je c izvan toga niza javlja se greška



**1 Zadatak:** Napisati program kojim se za unetu ocenu ispisuje: 1 - nedovoljan, 2 - dovoljan, 3 - dobar, 4 - vrlodobar, 5 - odličan. Ovaj zadatak može se rešiti uz pomoću složene IF naredbe ili pomoću CASE.

### Rešenje:

```

program ocene;
var ocena : integer;
begin
  Writeln('Unesi ocenu : ');
  Readln(ocena);
  case ocena of
    1: Writeln('Nedovoljan');
    2: Writeln('Dovoljan');
    3: Writeln('Dobar');
    4: Writeln('Vrlodobar');
    5: Writeln('Odlican');
  end;
End.
  
```

```

program ocene;
var ocena:integer;
begin
  write('unesi ocenu: ');
  readln(ocena);
  if ocena=1 then
    writeln('nedovoljan')
  else if ocena=2 then
    writeln ('dovoljan')
  else if ocena=3 then
    writeln ('dobar')
  else if ocena=4 then
    writeln ('vrlo dobar')
  else if ocena=5 then
    writeln ('odlican')
  end.
  
```

**Rešenje pod a)**

Selektor je ocena koja se kreće u intervalu od 1 do 5. Za unešenu vrednost 1 izvršava se naredba WriteLn('Nedovoljan'), za 2 naredba WriteLn('Dovoljan'); i tako do vrednosti 5.

**Rešenje pod b)**

Korišćena je složena naredba IF gde se vrednost promenljive ocena menja od 1 do 5, a na osnovu vrednosti izraza na ekranu dobijamo poruke nedovoljan (ako je ocena jednaka 1), dovoljan, dobar i tako dalje.

**2 Zadatak** Za dati redni broj dana odrediti ime dana. Potrebno je proveriti da li je izvršen korektan unos. Zadatak rešavamo kombinacijom if i case naredbi.

**Rešenje**

```

program daniunedelji;
var dan:integer;
{promenljiva dan predstavlja redni broj dana}
begin
writelN('unesi redni broj dana');
read(dan);
if (dan<1) or (dan>7)
then writelN('Pogresno unet broj dana')
else
begin
writelN('dan u sedmici je:');
  case dan of
    1:writelN('ponedeljak');
    2:writelN('utorak');
    3:writelN('sreda');
    4:writelN('cetvrtak');
    5:writelN('petak');
    6:writelN('subota');
    7:writelN('nedelja');
  end
end
end
end.

```

**FOR PETLJA:**

Naredba FOR se koristi kada imamo potrebu da neki deo programa izvorsimo vise puta. Ona je formata:

```

FOR n1 TO n2 DO
  "linija-koda"
END;

```

n1 i n2 su celobrojne vrednosti!

n1 je pocetna, a n2 krajnja vrednost brojac. Petlja se ponavlja n2 puta (ako je n2 = 5, petlja ce se ponoviti 5 puta...). Dokle god je vrednost brojac u intervalu od n1 do n2, izvrsavaju se iskljucivo naredbe koje se nalaze u FOR petlji. Kada brojac dostigne vrednost n2, petlja se završava i PASCAL izvrsava naredbe posle FOR petlje. Ukoliko u petlji pisemo vise linija koda, moramo pisati i naredbe BEGIN i END;

Primer:

```
FOR n1 TO n2 DO
  BEGIN
    "linija-koda-1"
    "linija-koda-2"
    "linija-koda-3"
    "linija-koda-n"
  END;
```

U sklopu FOR petlje imamo i jos jednu naredbu - DOWNTO  
Ona služi za odbrojanje unazad:

```
FOR n1 DOWNTO n2 DO
  "linija-koda"
END;
```

Gde  $n_1$  predstavlja početnu, a  $n_2$  krajnju vrednost ( $n_1 > n_2$ ,  $n_1 \in \mathbb{Z}$ ,  $n_2 \in \mathbb{Z}$ )  
Brojač možemo obeležiti bilo kojim slovom. Obavezna je deklaracija brojača.

### 1 Zadatak

- Za  $n$  brojeva izračunati njihov kvadrat.
- Za brojeve od 1 do 5 izračunati njihov kvadrat

```
program fora;
var i,n:integer;
begin
write ('n= ');
readln(n);
for i:=1 to n do
writeln(i,' ',sqr(i));
end.
```

```
program forb;
var i:integer;
begin
for i:=1 to 5 do
writeln(i,' ',sqr(i));
end.
```

U primeru a deklariramo promenljivu  $n$  koja predstavlja krajnju vrednost brojača i računamo kvadrat brojeva od 1 (početna vrednost brojača) pa do unetog broja  $n$  (bilo koji broj koji unesemo sa tastature).

U primeru b program prilikom svakog izvršavanja računa kvadrate brojeva od 1 do 5, odnosno 1, 4, 9, 16, 25 su vrednosti koje dobijamo na izlazu.

### 2 Zadatak Napiši program koji za rezultat daje sledeći ispis na ekran:

Unesi gornju granicu za X i Y : 2 3

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
```

**Rešenje:**

```

PROGRAM for_xy;
VAR x,y,do_x,do_y,proizvod:integer;
BEGIN
write ('Unesi gornju granicu za X i Y : ');
readln(do_x,do_y);
for x:=1 to do_x do
  for y:=1 to do_y do
    begin
      proizvod:=x*y;
      writeln(x:2,' * ',y:2,' = ',proizvod:3)
    end
  end.

```

U prvom prolazu x ima vrednost 1, y takođe 1. U drugom prolazu x je 1, y je 2, u trećem x je 1, y je 3. Kada y dobije vrednost 3, x se menja i postaje 2, y opet menja vrednost od 1 do 3.

**3 Zadatak:**

Neka je data sledeća grupa naredbi.

```

a := 5;
for i :=1 to 5 do
a :=a-i;
a :=a-4;

```

kolika će biti vrednost promenljive a posle izvršenja datog koda ?

**Rešenje:**

1. prolaz:

$$a = a - i = 5 - 1 = 4$$

2. prolaz:

$$a = a - i = 4 - 2 = 2$$

3. prolaz:

$$a = a - i = 2 - 3 = -1$$

4. prolaz:

$$a = a - i = -1 - 4 = -5$$

5. prolaz:

$$a = a - i = -5 - 5 = -10$$

i na kraju se izvršava naredba:

$$a = a - 4 = -10 - 4 = -14$$

Primitite da je u telu ciklusa samo naredba  $a := a - i$  koja će biti izvršena pet puta, dok će naredba  $a := a - 4$  biti izvršena samo jednom.

**4 Zadatak:**

Napisati program koristeći instrukciju for kojom se računa suma brojeva od 1 do 100 koji su deljivi sa 7 i sa 3.

**Rešenje:**

```

program suma1;
var
  suma, i : integer;
begin
  suma :=0;
  for i:=1 to 100 do
    if ((i mod 7 =0) and (i mod 3 = 0)) then
      suma :=suma+i;
    Writeln('Trazeni zbir je ',suma:5);
  end.

```

Brojevi koji zadovoljavaju dati uslov su: 21, 42, 63 i 84 čiji je zbir 210.

**5 Zadatak:**

Napisati program koristeći instrukciju for kojom se računa zbir brojeva od 1 do 100 čija je zadnja cifra 9.

**Rešenje:**

```

program prog;
var
  s, i : integer;
begin
  s :=0;
  for i:=1 to 100 do
    if ((i mod 7 =0) and (i mod 3 = 0)) then
      s :=s+i;
  WriteLn('Trazeni zbir je ',s:5);
  ReadLn
End.

```

Brojevi koji zadovoljavaju ovaj uslov su 9, 19, 29, 39, 49, 59, 69, 79, 89 i 99 čiji je zbir 540.

**6 Zadatak:**

Neka se sa tastature unosi 10 realnih brojeva. Napisati program kojim se izračunava njihov zbir i njihov proizvod.

**Rešenje:**

```

program prog2;
var
  a, zbir, proizvod : real;
  i : integer;
begin
  zbir :=0;
  proizvod :=1;
  for i :=1 to 10 do
    begin
      Write('Unesi ',i,' - broj :');
      ReadLn(a);
      zbir :=zbir+i;
      proizvod :=proizvod*i;
    end;
  WriteLn('Zbir brojeva je ',zbir:8:2,' Proizvod je : ',proizvod:8:2);
  ReadLn
End.

```

Primitite da je početni uslov za zbir nula, dok je to za proizvod 1.

**7 Zadatak:**

Poznato je u matematici da je faktorijel nekog celog broja definisan kao:

$$\text{faktorije} = n! = 1 * 2 * \dots * n$$

Napiši program koristeći instrukciju for kojom se računa faktorijel broja n unesenog sa tastature.

**Rešenje:**

```

program prog;
var faktorijel : real;
  n, i : integer;
begin
  faktorijel :=1;
  Write('Unesi zadati broj n :');

```



```

ReadLn(n);
for i :=1 to n do
begin
faktorijel :=faktorijel*i
end;
WriteLn('Faktorijel broja ',n,' je = ',faktorijel:13:2);
End.

```

**8 Zadatak:**

Napisati program koristeći instrukciju for kojom se računa zbir niza koji je definisan kao:

$$s=0-1+2=3+4\dots(-1)^n n$$

za broj n koji se unosi sa tastature.

**Rešenje:**

```

program progniz;
var zbir : real;
    i, n,pred : integer;
begin
  Write('Unesi zadati broj n :');
  ReadLn(n);
  pred := -1;
  zbir :=0;
  for i :=1 to n do
  begin
    zbir := zbir+(pred)*i;
    pred := -pred
  end;
  WriteLn('Trazeni zbir niza je ',zbir:8:2);
End.

```

za rešavanje ovog zadatka mora se definisati promenljiva pred, koja može imati vrednost 1 ili - 1 i u svakom ciklusu se menja i to za neparan i jednaka je - 1 a za paran 1.

**9 Zadatak:**

Neka je dat niz:

$$s=\sin(x)+\sin^2(x)+\sin^3(x)+\dots+\sin^n(x).$$

Napisati program koristeći instrukciju for kojom se računa ovaj zbir ako se sa tastature unosi n i broj x.

**Rešenje:**

```

program prog;
var
  zbir, pro, x : real;
  n, i : integer;
begin
  Write('Unesi zadati broj n :');
  ReadLn(n);
  Write('Unesi broj x :');
  ReadLn(x);
  zbir :=0;
  pro := 1;
  for i :=1 to n do
  begin
    pro :=pro*sin(x);
    zbir := zbir+pro;

```

promenljivoj zbir dodeljuje se nula, promenljivoj proizvod jedinica. Naredbe dodele se postavljaju ispred for petlje.

```
end;  
WriteLn('Trazeni zbir niza je ',zbir:8:2);  
End.
```

**10 Zadatak.** Napisati program koji ce celobrojnoj vrednosti A deset puta dodati vrednost b koja se svaki put povecava za 2, ako je pocetna vrednost  $b = 3$ .

**11 Zadatak.** Napisati program koji omogucava unos n brojeva i sabira sve brojeve vece od 5.

**12 Zadatak.** Napisati program koji unazad ispisuje brojeve pocevsi od broja n.