

# Praktikum

Turbo Pascal 1.5  
sa rješenim zadacima

Pregled i značenje **nekih** funkcija i procedura TurboPascala

1.	Opis funkcije	Funkcija	Primjer funkcije	Rezultat
2.	Cjelobrojno dijeljenje	<b>DIV</b>	3 DIV 2 1 DIV 2	1 0
3.	Ostatak cjelobrojnog djeljenja	<b>MOD</b>	1 MOD 2 2 MOD 2	1 0
4.	Kvadrat ( $x^2$ )	<b>SQR</b>	SQR (2) SQR (3)	4 9
5.	Kvadratni korijen $\sqrt{x}$	<b>SQRT</b>	SQRT (3) SQRT (9)	1,73 3
6.	Vraća cio dio realnog broja.	<b>TRUNC</b>	TRUNC(2.3) TRUNC(2.7)	2 2
7.	Zaokružuje na najbliži cijeli broj	<b>ROUND</b>	ROUND(-1.49) ROUND(1.5)	-1 2
8.	Apsolutna vrijednost	<b>ABS</b>	ABS(-2.99) ABS(-2.49)	2,99 2,49
9.	Generiše slučajni broj	<b>RANDOM</b>		
10.	Vraća redni broj upisanog znaka	<b>ORD</b>		
11.	Vraća ASCII vrijednost koja odgovara brojnoj vrijednosti	<b>CHR</b>		
12.	Vraća prijedhodnu vrijednost rednog tipa.	<b>PRED</b>		
13.	Vraća slijedeću rednu vrijednost.	<b>SUCC</b>		
14.	Ispituje kraj fajla (da li znak prijedstavlja kraj fajla) <i>Vrijednost vraćene promenljive je true, ako je kao tekući -ispitivani- označen element iz posljednjeg nepraznog ( tj. novi ). U suprotnom ova funkcija vraća false.</i>	<b>EOF</b>		
15.	Ispituje kraj linije -reda (da li znak prijedstavlja kraj linije)	<b>EOLN</b>		
16.	Direktan (neposredan) i prijevod tekućeg bloka. Ako je komanda u glavnom programu znači prekid -kraj programa. Ako se nalazi u pp procedura završava potprogram u kome je upotrebljena.	<b>EXIT</b>		
17.	Prekid izvršenja programa i odlazak u OS	<b>HALT</b>		
18.	čita vrijednost tastera ( <i>Bez čekanja Entera</i> )	<b>Readkey</b>		
19.	Daje dužinu stringa (broj znakova stringa)	<b>LENGTH</b>		
20.	Procedura povezivanja fajla i varijable – povezivanje datotečne promenljive sa konkretnom fizičkom datotekom na disku.	<b>ASSIGN</b>		procedura za rad sa tekstualnim fajlovima
21.	<b>Kreira novi fajl</b> ako ne postoji, a ako postoji briše savsadržaj.	<b>REWRITE</b>		
22.	<b>Otvora postojeći fajl</b>	<b>RESET</b>		
23.	<b>Otvora postojeći fajl za dodavanje</b> novih elemenata	<b>APPEND</b>		
24.	<b>Zatvara otvoren fajl</b>	<b>CLOSE</b>		
25.	Pozicionira –postavlja kursor na zadane koordinate	<b>GotoXY</b>		za poziv procedura neophodan (treba ga deklarirati) modul WinCrt unit
26.	Daje X koordinatu kursora	<b>WhereX</b>		
27.	Daje Y koordinatu kursora	<b>WhereY</b>		
28.	Procedura koja daje sistemski datum Potrebno deklarirati: <b>VAR y, m, d, dw : WORD;</b>	<b>GetDate</b>		za poziv procedura neophodan modul WinDos unit
29.	Procedura koja daje sistemsko vrijeme Potrebno deklarirati: <b>VAR h, m, s, hund : WORD;</b>	<b>GetTime</b>		

### Struktura programa u Pascal-u

Programi u Pascal-u se sastoje iz sljedećih elemenata, redom:

1. odeljak deklaracija **obilježja**,
2. odeljak definicija **konstanti**,
3. odeljak definicija **tipova**,
4. odeljak deklaracija **promenljivih**,
5. odeljak deklaracija **procedura i/ili funkcija**, i
6. izvršni dio programa obuhvaćen između begin i end.

```

program STRUKTprimjer;
{Dodatni moduli koje će program koristiti, npr: Wincrt, WinDOS... -->}
uses wincrt;
{Konstante -->}
const konstanta = 128;
{Definisanje korisničkih tipova podataka -->}
type tip_podatka = array [1..24] of integer;
{Varijable uz obavezno deklarisanje njihova tipa -->}
var broj : real;
    dan : tip_podatka;
{Procedura -->}
procedure p;
begin
    {...}
end;
{Funkcija i njezin tip Vrijednosti -->}
function f: byte;
begin
    {...}
end;

```

Naredbe za definisanje tipova i konstanti i deklarisanje nisu izvršne i služe kao dodatne informacije kompajleru (prevodiocu) Pascala.

Deklaracijom varijabli izabranim je imenima je pridruženo svojstvo da prijedstavljaju varijable određenog tipa, s nedefinisanom vrijednošću. Vrijednost im se mora pridružiti posebnim naredbama: za inicijalizaciju, dodjeljivanje, unos i ispis (: =, read, write).

Programiranje ne bi imalo smisla kad ne bi postojao ispis vrijednosti ("rezultata").

Kategorija "cjelobrojni tip" INTEGER definisana je sa:

SHORTINT | BYTE | INTEGER | WORD | LONGINT

Ova standardna imena označava podskupove cijelih brojeva iz domena prijema tabli:

Ime tipa	Domena
shortint	-128 do 127
byte	0 do 255
integer	-32768 do 32767
word	0 do 65535
longint	-2147483648 do 2147483647

Kao i u slučaju cjelobrojnog tipa, postoji nekoliko realnih tipova s različitom domenom (preciznošću) vrijednosti, što je prikazano u sljedećoj tabli:

Tipa	Domena	Broj cifri
real	$2.9 \times 10^{-39}$ do $1.7 \times 10^{38}$	11-12
single	$1.5 \times 10^{45}$ do $3.4 \times 10^{38}$	7-8
double	$5.0 \times 10^{324}$ do $1.7 \times 10^{308}$	15-16
extended	$3.4 \times 10^{4932}$ do $1.1 \times 10^{4932}$	19-20
comp	-263 + 1 do 263 - 1	19-20

### Zapisi -SLOG (record)

Zapisi (slogovi) su strukture podataka koje se, za razliku od nizova, sastoje od komponenti (segmenata) različitog tipa. Komponente sloga se obično nazivaju polja.

Svako polje poseduje ime i tip.

Imena polja se grade kao i drugi identifikatori. Zapisi se u programu definišu prijeko eksplicitnih definicija odgovarajućih strukturnih tipova podataka ili prijeko anonimnih opisa tipova.

Sljedeći primjer ilustruje eksplicitne definicije strukturnog tipa podataka koji obuhvata zapise:

```
type datum = record
```

```
mesec : (jan,feb,mar,apr,maj,jun,jul,avg,sep,okt,nov,dec);
```

```
dan : 1..31;
```

```
godina : 1900..2100;
```

```
end;
```

*Zapis (record) ne mora biti zapisan na nekom disku u nekoj datoteci, on može postojati samo za vrijeme izvršavanja programa, u memoriji.*

---

## Datoteke

Sa stanovišta Pascal-a datoteka se sastoji od sekvencijalno poredanih elemenata.

Karakteristično je to da je u svakom trenutku jedan element markiran i zove se tekući element (trenutno aktivni element). Tekući element može biti jedan od postojećih ili novi (tj. jedan iza poslednjeg postojećeg u sekvenci).

I prazna datoteka ima tekući element.

Procedure i funkcije se izvršavaju implicitno nad tekućim elementom, npr. prilikom pisanja ili čitanja uvek se pristupa tekućem elementu.

Pascal raspoznaje tri različita tipa datoteka, od kojih standardnu primjenu imaju prve dvije:

-tekstualna datoteka

-tipizirana datoteka

-netipizirana datoteka

## Tekstualne datoteke -procedure i komande za rad sa datotekama-

U Pascalu se ovaj tip datoteke intenzivno koristi. Tekstualna datoteka prijedstavlja poseban tip podataka i postoji posebno rezervirano ime za ovaj tip: text.

Primjer deklarisanja tekstualne datoteke:

```
var datoteka:text;
```

Osnovni element tekstualne datoteke je znak (char). Pošto ovakva datoteka sadrži tekst (ne formalne podatke), u nju se piše i iz nje se čita pomoću stringova.

Datoteke ove vrste postoje nezavisno od programa, imaju spoljnje identifikatore i **u programu se moraju povezivati sa spoljnim imenom koje je nezavisno od programa.**

**To se postize procedurom Assign.**

Pristup datoteci se vrši pomoću procedure Rewrite, Reset i Append.

Procedura **Rewrite** formira datoteku na spoljnjem mediju ako ona još ne postoji, a ako već postoji onda briše sav njen sadržaj. Posle izvršavanja ove procedure datoteka se sastoji od jedne prazne lokacije koja je tekuća. Treba još napomenuti da ova procedura datoteku otvara u "write-only" režimu, tj. kada otvorimo datoteku ovom procedurom u nju je moguće samo upisivanje.

Primjer korišćenja procedure Rewrite:

```
var f:text; {deklarisanje tekstualne datoteke}
begin
  assign(f,'C:\datoteka.txt'); {povezivanje datoteke sa spoljnjim imenom}
  rewrite(f); {izvršavanjem ove procedure pravi se datoteka "datoteka.txt" u C:\}
  ....
close(f); {posle upotrebe datoteka mora da se zatvori}
end.
```

Procedura **Reset** se koristi pod prijetpostavkom da datoteka već postoji nezavisno od toga da li sadrži neke podatke ili je prazna. U svakom slučaju ona (potencijalne) podatke ne dira i postavlja prvi element za tekući element. Ova procedura datoteku otvara u "read-only" režimu, što znači da je ovom procedurom moguće samo čitanje iz datoteke.

Primjer korišćenja procedure Reset:

```
var f:text; {deklarisanje tekstualne datoteke}
begin
  assign(f,'C:\datoteka.txt'); {povezivanje datoteke sa spoljnjim imenom}
  reset(f); {otvaranje datoteke za čitanje; datoteka mora da postoji!!!}
  ....
close(f); {posle upotrebe datoteka mora da se zatvori}
end.
```

Procedura **Append** otvara datoteku kao da je izvršena procedure Reset, ali se marker tekućeg elementa postavlja na kraj datoteke i u nju sada možemo upisivati nove podatke iza postojećih.

*Naravno i ovdje se podrazumeva da datoteka već postoji.*

Primjer korišćenja procedure Append:

```
var f:text; {deklarisanje tekstualne datoteke}
begin
  assign(f,'C:\datoteka.txt'); {povezivanje datoteke sa spoljnjim imenom}
  append(f); {datoteka mora postojati!otvorena za nastavljnje upisivanja u nju }
  ....
close(f); {poslije upotrebe datoteka mora da se zatvori}
end.
```

*Napomena: prije korišćenja datoteke potrebno je izvršiti Reset, Rewrite ili Append!!!*

**Funkcija EoF** (End of File) je ugrađena funkcija koja vraća promenljivu logičkog tipa (boolean). Vrijednost vraćene promenljive je true, ako je kao tekući označen element iza poslednjeg **nepraznog** (tj. novi). U suprotnom ova funkcija vraća false.

**Procedura Close** se koristi za zatvaranje datoteke.

**Funkcija eoln** pokazuje kada nađemo na kraj reda. Ova funkcija je u Pascalu izgubila značaj, jer postoji readln koja učitava cijeli red odjednom.

## Primjeri i zadaci

Sa punim rezervoarom kapaciteta V litara pređen je put od S kilometara.

Napisati program kojim se računa potrošnja automobila u litrima na 100 km. Ispis prikazati sa tačnošću (u formatu) sa dvije decimale.

```
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.
```

Date su dvije tačke u koordinatnom sistemu, zadate svojim koordinatama.

Izračunati njihovo rastojanje od koordinatnog početka redom. (*Pitagora*)

```
Program rastojanje;
uses winCRT;
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));

  writeln('Rastojanje prve tacke i koordinatnog pocetka ',d1:8:3);
  writeln('Rastojanje druge tacke i koordinatnog pocetka ',d2:8:3);
end.
```

Napisati program koji za unešeni napon U i struju I sa tastature izračunava otpor i snaga kola.

Rezultat prikazati u formatu od 6 mhesta sa 3 decimale.

```
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.
```

Putovanje između mjesta A i B traje n sekundi. Sračunaj koliko je to sati, minuta i sekundi.

```
program pSati;
Uses winCRT;
var
  n,sat,min,sek : integer;
begin
  ClrScr;
  Write('Unesi sekunde : ');
  ReadLn(n);
  sat := n div 3600;
  sek := n mod 3600;
  min := sek div 60;
  sek := sek mod 60;
  WriteLn('sati = ',sat:5, ' minuta = ',min:5,' sek = ',sek:5);
  ReadLn
End.
```

Napisati program kojim se vrši prevodjenje količine tečnosti iz galona u litre, ako je 1 galon = 4.54 litra. (*koliko je litara, ako znamo koliko je galona*)

Ukoliko se unese negativan broj javiti grešku.

Ispis prikazati sa tačnošću (u formatu) sa dvije decimale.

```
program IzGuLit;
uses WinCrt;
var
  g, lit : real;
begin
  write('Unesite kolicinu tecnosti u galonima: ');
  readln(g);
  IF g>=0 then
  begin
    lit := g*4,54;
    writeln('Odgovarajuca kolicina u litrima je: ', lit:8:2 );
  end
  else writeln('Greska kod unosa ');
end.
```

Nacrtati dijagram i napiši program koji kad su unešena dva broja a i b, veći dijeli sa 3, a manji množi sa 4 i ispisuje rezultate te operacije (*veći podjeljen sa 3 i manji pomnožen sa 4*).

```
program brojevi;
uses wincrt;
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.
```

Neka se sa tastature unose brojevi  $n_1$ ,  $n_2$ , i  $n_3$ . Napiši program kojim se promjenljivoj  $max$  dodjeljuje najveća vrijednost od unesenih brojeva. Rezultat prikaži u formatu od 8 mesta sa 2 decimalna.

```
program pMax3;
Uses winCRT;
var
  n1,n2,n3,max : real;
begin
  ClrScr;
  Write('Unesi n1 : ');
  ReadLn(n1);
  Write('Unesi n2 : ');
  ReadLn(n2);
  Write('Unesi n3 : ');
  ReadLn(n3);
  max := n1;
  if (n2 >= max) then max := n2;
  if (n3 >= max) then max := n3;
  WriteLn('Najveci uneseni broj je:',max:8:2);
  ReadLn
End.
```

Neka se sa tastature unose celi brojevi  $n_1$ ,  $n_2$ , i  $n_3$ . Napiši program kojim se sračunava ukupan broj negativnih brojeva. Rezultat prikaži u formatu od 5 cjelih mesta.

```
program progSng;
Uses winCRT;
var
  n1,n2,n3,s : integer;
begin
  ClrScr;
  Write('Unesi n1 : ');
  ReadLn(n1);
  Write('Unesi n2 : ');
  ReadLn(n2);
  Write('Unesi n3 : ');
  ReadLn(n3);
  s := 0;
  if n1 < 0 then s := s+1;
  if n2 < 0 then s := s+1;
  if n3 < 0 then s := s+1;
  WriteLn('Ukupan broj negativnih brojeva je :',s:5);
  ReadLn
End.
```



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

```
program progZ2;
Uses winCRT;
var
  s, i : integer;
begin
  ClrScr;
  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.
```

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

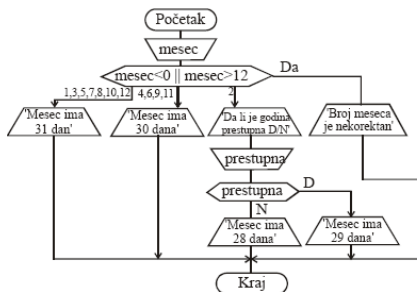
```
program prog;
Uses winCRT;
var
  s, i : integer;
begin
  ClrScr;
  s :=0;
  for i:=1 to 100 do
    if ((i mod 7 =0) and (i mod 3 = 0)) then s :=s+i;
  WriteLn("Trazeni zbir je ',s:5);
  ReadLn
End.
```

Napisati program kojim se izračunava 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.
```

Nacrtaati dijagram i napisati program koji za unijeti broj mjeseca (na primjer, 1=januar, 2=februar, itd) ispisuje njegov broj dana.

U slučaju unijete vrijednosti 2 pitati korisnika da li je godina prestopna.



```

program DaniM;
uses WinCrt;
var
  mjesec : integer;
  prestupna : char;

```

```

begin
  write('Unesite broj mjeseca od 1 do 12: ');
  readln(mjesec);

```

```

if (mjesec<1) or (mjesec>12)
  then writeln('Broj mjeseca je nekorektan')
  else
    case mjesec of
      1,3,5,7,8,10,12 : writeln('Mjesec ima 31 dan');
      4,6,9,11 : writeln('Mjesec ima 30 dana');
      2 :
        begin
          write('Da li je godina prestupna (D/N)?');
          readln(prestupna);
          if (prestupna='D') or (prestupna='d')
            then writeln('Mjesec ima 29 dana')
            else writeln('Mjesec ima 28 dana');
          end;
        end;
    end;
end;

```

```
end.
```

String može da se shvati kao niz karaktera. Upišite neki string sa tastature, a onda napišite program koji će ispisati predhodno unešeni string tako da u jednom redu bude redni broj znaka i znak (od početka do kraja stringa, svako slovo novi red. *Koristi komandu za dužinu stringa i petlju za ispis*)

```

PROGRAM PrLString;
USES WinCRT;
VAR
  i : Integer;
  s : String;
BEGIN
  Write('Unesi string ');
  Readln(s);
  FOR i := 1 TO LENGTH(s) DO
    Writeln(i, '-ti znak = ', s[i]);
  END.

```

Nacrtni dijagram i napisati program kojim se u zadatom tekstu - stringu s (koga unesemo sa tastature), broji pojavljivanje datog znaka x. (Izračunaj koliko je slova u tekstu)

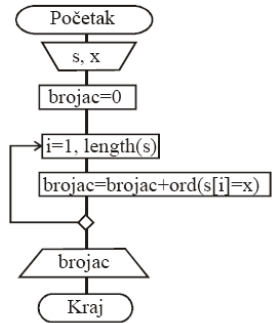
```

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
brojac:=brojac+ord(s[i]=x);
writeln(x,' se pojavljuje ',brojac,' puta. ');
end

```



Neka su data dva niza xx i yy od po pet članova. Napiši program kojim se računa ukupan broj parova nizova koji na istom indeksu imaju jednake elemente.

```

program pet;
Uses winCRT;
type niz=array[1..100] of real;
var
xx, yy : niz;
i, br: integer;
begin
ClrScr;
for i := 1 to 5 do
begin
Write('Unesi - ',i:3,' clan niza xx : ');
ReadLn(xx[i])
end;
for i := 1 to 5 do
begin
Write('Unesi - ',i:3,' clan niza yy : ');
ReadLn(yy[i])
end;
br := 0;
for i := 1 to 5 do
if (xx[i] = yy[i]) then br := br+1;
WriteLn('Trazeni broj parova je: ',br:5);
ReadLn
End.

```

---

Naredni zadaci ilustruju otvaranje nove datoteke, te dodavanje i čitanje podataka iz postojeće datoteke.

---

Napisati program za unos podataka o automobilima.

Na početku definisati broj automobila  $n$  i podatke o vrsti (marka auta), godini proizvodnje, cijeni i registarskom broju smjestiti u **datoteku auto.txt**.

```
program UnosAuta;
uses wincrt;

type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cijena :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('Unijeti broj automobila: ');
  readln(n);

  for i:=1 to n do
    begin
      write('Unijeti vrstu automobila: '); readln(a[i].vrsta);
      write('Godina proizvodnje: '); readln(a[i].godproiz);
      write('Cijena automobila: '); readln(a[i].cijena);
      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].cijena);
      writeln(f, a[i].registar);
    end;
  close(f)
end.
```

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

```
program DodavanjeAuta;
uses wincrt;

type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cijena :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');
  readln(n);
  writeln('Unesi podatke');
  writeln;

assign(f, 'auto.txt');
append(f);
  for i:=1 to n do
    begin
      write('Unesi vrstu automobila: '); readln(a[i].vrsta);
      writeln(f,a[i].vrsta);
      write('Unijeti godinu proizvodnje: '); readln(a[i].godproiz); writeln(f,a[i].godproiz);
      write('Unijeti cijenu: ');
      readln(a[i].cijena); writeln(f,a[i].cijena);
      write('Unijeti registarski broj: '); readln(a[i].registar); writeln(f,a[i].registar);
      writeln;
    end;
  close(f);
end.
```

Iz formirane datoteke auto.txt pročitati podatke o automobilima i izračunati ukupnu vrijednost svih automobila zadate vrste automobila. Ispisati na ekranu vrstu, godinu proizvodnje i cijenu automobila koji su proizvedeni poslije zadate godine.

```
program Automobil;
uses wincrt;

type
  auto = record
    vrsta :string[18];
    godproiz :integer;
    cijena :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('Vrsta auta');
  readln(ime);
  writeln('Unijeti godinu proizvodnje od koje se ispisuje vrsta i cijena automobila');
  readln(godina);

  assign(f, 'auto.txt');
  reset(f);
  zbir:=0;
  n:=1;
  writeln;
  writeln('Automobli proizvedeni poslije ',godina,' su: ');

  while not eof(f) do

begin
  readln(f,a[n].vrsta);
  readln(f,a[n].godproiz);
  readln(f,a[n].cijena);
  readln(f,a[n].registar);
  if a[n].vrsta=ime then
    zbir:=zbir+a[n].cijena;
    if a[n].godproiz>godina then
      writeln(a[n].vrsta, ': ',a[n].godproiz, ' god. ', a[n].cijena, ' KM');
    n:=n+1;
  end;
  writeln;
  writeln('Ukupna vrijednost svih automobila tipa ',ime, ' je ',zbir);
close(f)
end.
```

---

Napiši program kojim se u tekstualnu datoteku ucenik unose podaci: prezime i ime, mjesto stanovanja i adresa. Program završi kada se za prezime unese riječ 'kraj'.

```
program prog;  
Uses winCRT;  
Var  
  ucenik : text;  
  ime : string[25];  
  mjesto : string[15];  
  adresa : string[20];  
  x : real;  
begin  
  ClrScr;  
  assign(ucenik,'c:\ucenik.dat');  
  rewrite(ucenik);  
  while not (ime='kraj') do  
    begin  
      ClrScr;  
      Write('1. Prezime i ime : ');  
      ReadLn(ime);  
      WriteLn;  
      Write('2. Mjesto stanova. : ');  
      ReadLn(mjesto);  
      WriteLn;  
      Write('3. Adresa      : ');  
      ReadLn(adresa);  
      if not (ime='kraj') then  
        Write(ucenik,ime,mjesto,adresa);  
    end;  
  close(ucenik);  
  readln;  
end.
```

Napiši program kojim se iz prethodno formirane tekstualne datoteku ucenik, prikazuju na ekranu uneseni podaci.

```
program progdt1;  
Uses winCRT;  
Var  
  ucenik : text;  
  ime : string;  
  mjesto : string;  
  adresa : string;  
begin  
  ClrScr;  
  assign(ucenik,'c:\ucenik.dat');  
  reset(ucenik);  
  repeat  
    ReadLn(ucenik, ime, mjesto, adresa);  
    WriteLn;  
    Write(ime,' ',mjesto,' ', adresa);  
  until eoln(ucenik);  
  close(ucenik);  
  readln;  
end.
```