

1.	<pre> var x: string; function kaj(var a: string): boolean; begin   if length(a) = 0 then kaj := false   else begin     if length(a) = 1 then kaj := true     else begin       delete(a, 1, 2);       kaj := kaj(a);     end;   end; end;  begin   x := 'cesta';   writeln(kaj(x));   writeln(x); end. </pre> <p>Program izpise: True a</p> <p>Funkcija nam vrne, ali je parameter sode (false) ali lihe (true) dolžine. Če je lihi, v parametru ne ostane nič, če pa je sod, pa v njem ostane zadnja črka. To doseže z brisanjem do zadnje ali nictje crke.</p>
2.	<pre> var a, i: integer; procedure prva(var a: integer; i: integer); procedure druga(a: integer); var i: integer; begin   i:=4; a:=a+i;   writeln('Druga ',a,' ',i); end; begin   i := i+a; writeln('Prva ',a,' ',i);   druga(a);   a := a+i; writeln('Prva ',a,' ',i); end;  begin   a:=5; i:=2;   prva(a, i);   writeln('Glavni ',a,' ',i); end.  </pre> <p>Program izpise: Prva 5 7 Druga 9 4 Prva 12 7</p>

	<p>Glavni 12 2</p> <p>Treba je pazati na prenasanje vrednosti po spremenljivkah (kdaj je var kdaj ni)</p>
3.	<pre>function kaj(a, b: word): word; begin   if a = b then kaj := a   else begin     if a&gt;b then kaj := kaj(a-b, b)     else kaj := kaj(a, b-a);   end; end;  begin   writeln(kaj(18,4));   writeln(kaj(12,30)); end.</pre> <p>Funkcija 'kaj' vedno odsteje manjsi parametr od vecjega dokler nista enaka takrat pa jih vrne v glavni program kjer se izpiseta.</p>
4.	<pre>var x: string; procedure kaj(x: string); begin   x[0] := chr(ord(2));   writeln(x); end;  begin   x := 'BALADA'; kaj(x);   writeln(x);   x := copy(x, pos('A', x), 2);   writeln(x); kaj(x); end.</pre> <p>Program izpise:</p> <p>BA BALADA AL AL</p>
5.	<pre>function kaj(a: word): word; begin   if a = 1 then kaj := 1   else     if a = 2 then kaj := 5     else       kaj := kaj(a-1)+kaj(a-2); end;  begin   writeln(kaj(4));   writeln(kaj(10));</pre>

	<p>end.</p> <p>Program izpise:</p> <p>11 191</p> <p>Program vrne vrednost zaporedja na mestu parametra.</p>												
6.	<pre> var a, b: string; procedure prva(var a: string; b: string); var c: string; procedure druga(var a: string); var b: string; begin   b := copy(a, pos('T', a), 2);   a[0] := chr(3);   writeln('Druga ',a,' ',b); end; begin   c := a+b;   a := concat(b, a);   b := a;   writeln('Prva ',a,' ',b,' ',c);   druga(b);   writeln('Prva ',a,' ',b,' ',c); end;  begin   a := 'AVTO'; b := 'CESTA';   prva(a, b);   writeln('Glavni ',a,' ',b); end. </pre> <p>Program izpise:</p> <p>Prva CESTAAVTO CESTAAVTO AVTOCESTA    Druga CES TA    Prva CESTAAVTO CES AVTOCESTA    Glavni CESTAAVTO CESTA</p> <p>Sintaksa uporabljenih podprogramov za delo z nizi:</p> <table border="1"> <tr> <td>Length(S)</td> <td>Vrne trenutno dolzino niza: lahko tudi z Ord(S[0]);</td> </tr> <tr> <td>Copy(S,Index,Count)</td> <td>iz niza S vrne podniz, ki se zacne na poziciji Index in ima Count znakov.</td> </tr> <tr> <td>Pos(SubStr,S)</td> <td>Poisce, kje v nizu S se prvic pojavi podniz SubStr. Ce se ne vren funkcija vrednost 0.</td> </tr> <tr> <td>Delete(S,Index,Count)</td> <td>Iz niza S zbrise na poziciji Index, Count znakov.</td> </tr> <tr> <td>Insert(Source,S,Index)</td> <td>V niz S vrine na pozicijo Index niz Source</td> </tr> <tr> <td>Concat</td> <td>zdruzi zaporedje nizov v nov niz, lahko uporabljamo tudi +</td> </tr> </table>	Length(S)	Vrne trenutno dolzino niza: lahko tudi z Ord(S[0]);	Copy(S,Index,Count)	iz niza S vrne podniz, ki se zacne na poziciji Index in ima Count znakov.	Pos(SubStr,S)	Poisce, kje v nizu S se prvic pojavi podniz SubStr. Ce se ne vren funkcija vrednost 0.	Delete(S,Index,Count)	Iz niza S zbrise na poziciji Index, Count znakov.	Insert(Source,S,Index)	V niz S vrine na pozicijo Index niz Source	Concat	zdruzi zaporedje nizov v nov niz, lahko uporabljamo tudi +
Length(S)	Vrne trenutno dolzino niza: lahko tudi z Ord(S[0]);												
Copy(S,Index,Count)	iz niza S vrne podniz, ki se zacne na poziciji Index in ima Count znakov.												
Pos(SubStr,S)	Poisce, kje v nizu S se prvic pojavi podniz SubStr. Ce se ne vren funkcija vrednost 0.												
Delete(S,Index,Count)	Iz niza S zbrise na poziciji Index, Count znakov.												
Insert(Source,S,Index)	V niz S vrine na pozicijo Index niz Source												
Concat	zdruzi zaporedje nizov v nov niz, lahko uporabljamo tudi +												
7.	function kaj(x: string): boolean; begin												

```

if(length(x) = 1) or (length(x) = 0) then
  kaj := true
else begin
  if x[1] = x[length(x)] then begin
    delete(x,1,1);
    delete(x,length(x),1);
    kaj := kaj(x);
  end else
    kaj := false;
  end;
end;

begin
  writeln(kaj('ROMOR'));
  writeln(kaj('SOPHOS'));
end.

```

Program izpise:

TRUE  
FALSE

Program prepoznava palindrome.

```

8. uses crt;
type tabela = array['A'..'Z'] of byte;
var t: tabela;

procedure nastavi(var t: tabela);
var i: char;
begin
  for i := 'A' to 'Z' do t[i] := 0;
end;

procedure izpis(t: tabela);
var i: char;
begin
  for i := 'A' to 'Z' do
    if t[i] > 0 then writeln(i, '...', t[i]);
end;

procedure kaj(var t: tabela; x: string);
var i: byte;
begin
  for i := 1 to length(x) do inc(t[x[i]]);
end;

begin
  clrscr;
  nastavi(t); kaj(t, 'PLAVALKA'); izpis(t);
  nastavi(t); kaj(t, 'TRST'); izpis(t);
end.

```

	<p>Program izpise:</p> <p>A...3 K...1 L...2 P...1 V...1 R...1 S...1 T...2</p> <p>Procedura kaj v tabelo zapiše število ponovitev določene črke iz parametra.</p>
9.	<pre> var x,y: string; procedure izpis(var a: string; b: string); var i: integer;  function kaj(a, b: string; c: byte): boolean; begin   if pos(b, a) = c then kaj := true else kaj := false; end;  begin   for i := 1 to 3 do begin     writeln(kaj(a, 'T', length(a)));     delete(a, 1, 1);   end;   b := a;   writeln(a, ' ', b); end;  begin   x := 'TRST';   y := 'JE';   izpis(x, y);   writeln(x, ' ', y);   readln; end. </pre> <p>Program izpise:</p> <p>FALSE TRUE TRUE T T T JE</p> <p>Funkcija kaj vrne true, če se string b pojavi v stringu a na c-tem mestu.</p>
10	<pre> type tab = array[1..6] of string; var a: tab;  procedure izpis(a: tab); var i, j: byte; </pre>

```

function kaj(x: string): boolean;
begin
  kaj := pos(x[1], copy(x, 2, length(x) - 1)) > 0;
end;

begin
  for i := 5 downto 1 do begin
    for j := length(a[i]) downto 1 do
      write(a[i,j]);
    writeln(' ', kaj(a[i])); { tukaj kličemo z pravilno obrnjenim stringom }
  end;
end;

begin
  a[1] := 'VALVAZOR';
  a[2] := 'KERSNIK';
  a[3] := 'CANKAR';
  a[4] := 'CAMUS';
  a[5] := 'KAFKA';
  izpis(a);
end.

```

Program izpise:

AKFAK TRUE  
 SUMAC FALSE  
 RAKNAC FALSE  
 KINSREK TRUE  
 ROZAVLAV TRUE

Funkcije kaj preverja ali se prvi znak niza ponovi se kje v nizu, ce ne vrne FALSE.

11

```

procedure kaj(a: string; b: boolean);
begin
  if a <> " then begin
    if b then begin
      b := not(b);
      write(a[1]);
      delete(a,1,1);
      kaj(a,b);
    end else begin
      b := not(b);
      write(a[length(a)]);
      delete(a, length(a),1);
      kaj(a,b);
    end;
  end;
end;

begin
  kaj('MOJCA', odd(length('MOJCA'))); writeln;

```

	<pre>kaj('DISK', odd(length('DISK'))); writeln; end.</pre> <p>Program izpise:</p> <p>MAOCJ KDSI</p> <p>Funkcija 'kaj' izpiše in odstrani prvo črko, nato izpiše in odstrani zadnjo črko in nato spet prvo in zadnjo, dokler ni parameter prazen. Ali pa najprej izpiše in odstrani zadnjo črko in nato prvo. Izbor poteka glede na sodost/lihost parametra.</p>
12	<pre>. var x: integer; procedure izpis(var a: integer); function preveri(x: byte): boolean; begin   preveri := odd(x); end;  begin   while a &lt;&gt; 0 do begin     writeln(preveri(a mod 10), ' ');     a := a div 10;   end;   inc(a); end;</pre> <pre>begin   x := 261; izpis(x); writeln(x);   x := 10; izpis(x); writeln(x); end.</pre> <p>Program izpise:</p> <p>TRUE FALSE FALSE 1 FALSE TRUE 1</p> <p>Procedura za vsako stevko parametra izpiše, ali je soda ali liha od zadnje proti prvi.</p>
13	<pre>. type Tprvi = object   a, b: byte;   constructor init(va, vb: byte);   procedure pisi;   function racun: integer; end;</pre> <pre>constructor Tprvi.init(va, vb: byte); begin</pre>

	<pre> a := va; b := vb; end;  procedure Tprvi.pisi; begin   writeln('Izpis: ', racun); end;  function Tprvi.racun: integer; begin   racun := a+b; end;  type Tdrug = object(Tprvi)   d: byte;   constructor init(va, vb, vd: byte);   function racun: integer; end;  constructor Tdrug.init(va, vb, vd: byte); begin   Tprvi.init(va, vb);   d := vd; end;  function Tdrug.racun: integer; begin   racun := a+b+d; end;  var x: Tprvi; y: Tdrug; begin   x.init(3,4); x.pisi;   y.init(6,7,8); y.pisi; end. </pre> <p>Program izpise:</p> <p>Izpis: 7</p> <p>Izpis: 13</p>
14	<pre> type barva = (bela, rumena, zelena, modra); var a, b: barva;  procedure naredi(a, b: barva); var c: barva;   procedure p2(a: barva);   var b: barva;   begin     b := a; a := succ(a);     writeln('P2 ', ord(a), ' ', ord(b));   end; </pre>

```

begin
  a := pred(b);
  for c := a to modra do
    write(ord(c), ' ');
  writeln;
  p2(a);
  writeln('naredi ', ord(a), ' ', ord(b));
end;

begin
  a := bela; b := zelena;
  naredi(a, b);
  writeln('glavni ', ord(a), ' ', ord(b));
end.

```

Program izpise:

```

1 2 3
P2 2 1
naredi 1 2
glavni 0 2

```

Podprogrami za delo z tipi:

inc(x)	spremenljivi poveca vrednost za ena, priredi ji naslednji element
dec(x)	spremenljivki zmanjsa vrednost za ena, priredi ji prejšni element
succ(x)	vrne vrednost naslednjega elementa
predd(x)	vrne vrednost prejšnjega elementa
ord(x)	vrne zaporedno stevilko vrednosti x
ime_tipa(N)	izpise tip, ki se nahaja na N tem mestu

16 type barva = (bela, rumena, zelena, modra)  
 . var a, b: barva;

```

procedure naredi(a, b: barva);
var c: barva;
  procedure p2(a, c: barva);
  var b: barva;
  begin
    b := a; a := c; c := b;
    writeln('P2 ', ord(a), ' ', ord(c));
  end;
begin
  for c := a to b do p2(a, c);
  c := a; a := b; b := c;
  writeln('naredi ', ord(a), ' ', ord(c));
end;

begin
  a := bela; b := rumena;
  naredi(a, b);
  writeln('glavni: ', ord(a), ' ', ord(b));
end.

```

	Program izpis: P2 0 0 P2 1 0 naredi 1 0 glavni: 0 1
17 .	<pre> type Tniz5 = string[5]; knjiga = object   avtor: Tniz5;   cena: integer;   procedure init(na: Tniz5; nc: integer);   function rabat: integer;   function prodaja: integer; end; ucbenik = object(knjiga)   predmet: Tniz5;   procedure init(na, np: Tniz5; nc: integer);   function rabat: integer; end;  procedure knjiga.init(na: Tniz5; nc: integer); begin   avtor := na; cena := nc; end;  function knjiga.rabat; begin   rabat := 0; end;  function knjiga.prodaja: integer; begin   prodaja := cena + rabat; end;  procedure ucbenik.init(na, np: Tniz5; nc: integer); begin   knjiga.init(na, nc); predmet := np; end;  function ucbenik.rabat: integer; begin   if predmet = 'RSM' then rabat := round(0.5 * cena)     else rabat := round(0.2 * cena); end;  var x: knjiga;   y: ucbenik; begin   x.init('CANKAR', 6000); </pre>

```
writeln(x.prodaja);
Y.init('MS', 'RSM', 4000);
writeln(y.prodaja);
readln;
end.
```

Program izpise:  
6000  
4000