

RAČUNARSTVO I INFORMATIKA – ISM

ČAS 13



- Složeniji programi se mogu podeliti na odgovarajuće logičke i funkcionalne celine, takozvane potprograme.
- Potprogram omogućava organizovanje određenog broja instrukcija u celine koje obavljaju određene zadatke.
- Potprogram preuzima ulazne veličine i vrši njihovu obradu u cilju dobijanja određenih rezultata.
- Potprogram možemo pozivati više puta u toku izvršavanja programa.



PREDNOSTI KORIŠĆENJA POTPROGRAMA

- Omogućava koncentrisanje na izvršavanje samo određenog zadatka
- Različiti članovi razvojnog tima mogu razvijati različite potprograme koji se kasnije sklapaju u jednu celinu
- Omogućava izvršavanje istog zadatka na više mesta u programu jednostavnim pozivanjem odgovarajućeg potprograma



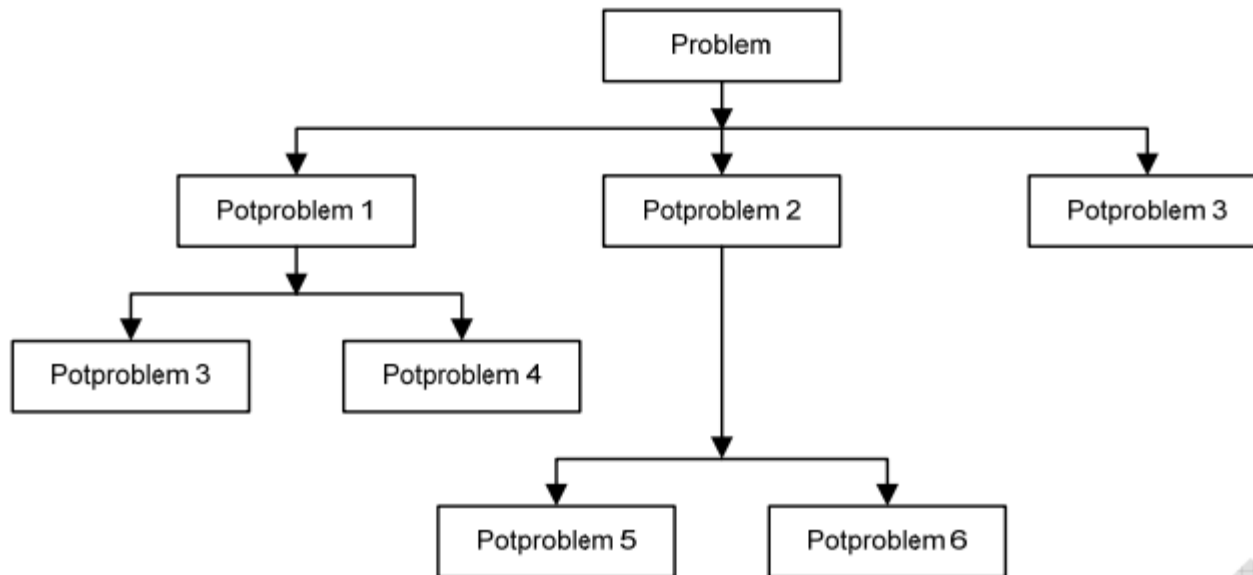
- Funkcija je samostalan deo programa koji obavlja određeni zadatak
- Preko svog naziva i liste parametara, delu programa iz koga je pozvana vraća odgovarajuće rezultate
- Svaka funkcija ima jedinstveni naziv preko koga se može pozvati proizvoljan broj puta iz bilo kog dela programa u cilju izvršenja tog zadatka.



- Procedura je potprogram sličan funkciji sa tom razlikom što procedura preko svog imena ne vraća nikakvu vrednost delu programa iz koga je pozvana.



- Funkcije i procedure omogućavaju realizaciju programiranja "odozgo na dole". Ovaj metod podrazumeva razbijanje problema na manje "potprobleme", a zatim i novodobijenih "potproblema" na još manje celine, sve do trenutka dok se glavni problem ne svede na rešavanje elementarnih problema.





```
function <naziv>  
[(lista parametara)]:<tip_funkcije>;  
[deklaracija_lokalnih_promenljivih]  
<blok_naredbi>
```

```
procedure <naziv>  
[(lista parametara)];  
[deklaracija_lokalnih_promenljivih]  
<blok_naredbi>
```



- Napisati program koji izračunava i štampa rastojanje između dve tačke čije su koordinate date na ulazu.

```
program Rastojanje;  
var x1,y1,x2,y2,r:real;  
  
function rast(x1,y1,x2,y2:real):real;  
begin  
    rast:=sqrt(sqr(x2-x1)+sqr(y2-y1));  
end;  
procedure stampaj(d:real);  
begin  
    writeln('Rastojanje izmedju tacaka je ',d:10:4);  
end;
```




begin

```
writeln('Unesite koordinate prve tacke:');
```

```
read(x1,y1);
```

```
writeln('Unesite koordinate druge tacke:');
```

```
read(x2,y2);
```

```
r:=rast(x1,y1,x2,y2);
```

```
stampaj(r);
```

end.



- Definisati funkciju **BrojDel** koja za dati broj određuje broj delilaca. Napisati program koji, koristeći funkciju **BrojDel** za uneti broj ispisuje da li je broj prost ili ne.

```
program Zad2;  
var n:integer;  
  
function BrojDel(n:integer):integer;  
var i,b:integer;  
begin  
    b:=0;  
    for i:=1 to n do  
        if n mod i = 0 then b:=b+1;  
    BrojDel:=b  
end;
```



- Definisati funkciju **BrojDel** koja za dati broj određuje broj delilaca. Napisati program koji, koristeći funkciju **BrojDel** za uneti broj ispisuje da li je broj prost ili ne.

```
begin
  readln(n);
  if BrojDel(n) <= 2 then
    writeln('Broj je prost')
  else
    writeln('Broj je slozen')
end.
```



- Definisati funkciju **F** koja je zadata na sledeći način:

$$F(x, y, z) = \max\{x, y\} \cdot \min\{y, z\}$$

Napisati program koji, koristeći funkciju **F**, za unete brojeve **a**, **b**, **c** računa

$$d = \max\{a + b, ac\} \cdot \min\{ac, b\}$$



```
program Zad3;
var a,b,c,d:integer;

function max(x,y:integer):integer;
begin
    if x>y then max:=x
    else max:=y
end;

function min(x,y:integer):integer;
begin
    if x<y then min:=x
    else min:=y
end;
```



```
function F(x,y,z:integer):integer;  
begin  
    F:=max(x,y)*min(y,z)  
end;  
  
begin  
    readln(a,b,c);  
    d:=F(a+b,a*c,b);  
    writeln(d:6)  
end.
```



- Definirati funkciju **Proizv** koja za m unetih celih brojeva izračunava proizvod savršenih među njima (broj je savršen ako je jednak sumi svojih delitelja ne uključujući njega samog, npr. $28=1+2+4+7+14$).



```
program Zad4;  
var m:integer;  
  
function Savrsen(n:integer):boolean;  
var i,s:integer;  
begin  
    s:=0;  
    for i:=1 to n-1 do  
        if (n mod i=0) then s:=s+i;  
    if n=s then Savrsen:=true  
    else Savrsen:=false;  
end;
```




```
function Proizv(m:integer):integer;
var i,p,x:integer;
begin
    p:=1;
    for i:=1 to m do begin
        readln(x);
        if Savrsen(x)then p:=p*x;
    end;
    Proizv:=p;
end;

begin
    readln(m);
    writeln(Proizv(m));
end.
```



Definisati funkciju **Stepen** koja za realna broj x i prirodan broj n izračunava x^n . Napisati program koji za unetih n realnih brojeva a_0, a_1, \dots, a_n i realan broj x , koristeći funkciju **Stepen** izračunava vrednost polinoma $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$



```
program Polinom;
var i,n:integer;
    a,x,p:real;

function Stepen(a:real; n:integer):real;
var i:integer;
    s:real;
begin
    s:=1;
    for i:=1 to n do s:=s*a;
    Stepen:=s;
end;
```



```
begin
  writeln('Unesite redom x i n:');
  readln(x,n);
  write('a0=');
  readln(a);
  p:=a;
  for i:=1 to n do
    begin
      write('a',i,'=');
      readln(a);
      p:= p + a * Stepen(x,i);
    end;
  writeln(p:10:5);
end.
```