

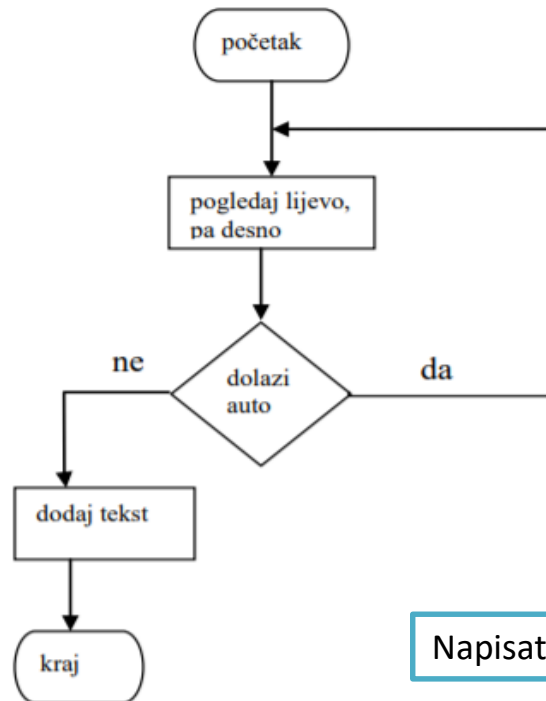
# Algoritmi i pseudokod

ssploce

# Zadaci

## 1. zadatak:

Nacrtati algoritam za prelazak ceste na zebri bez semafora.



Napisati pseudokod za zadanu sliku

## Nacrtati DTP i napisati pseudokod

### **2. zadatak:**

Algoritam za izračunavanje hipotenuze pravokutnog trokuta.

### **3. zadatak:**

Algoritam od dva broja uvijek ispisuje veći broj.

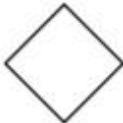
### **4. zadatak:**

Algoritam preračunava kune u eure ili US dolare.

### **5. zadatak:**

Algoritam zbraja brojeve koje korisnik upisuje. Kada upiše 0 program ispisuje ukupni zbroj.

## Prisjetimo se

|                                                                                     |                                 |                                                                                               |                            |
|-------------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------|----------------------------|
|    | Početak ili kraj algoritma      | Početak<br>Kraj.                                                                              |                            |
|    | Unos podataka, prikaz rezultata | Ulaz(x);<br>Ulaz(x,y);<br>Izlaz(x);<br>Izlaz('Ispisani tekst');<br>Izlaz('Vrijednost je', x); |                            |
|    | Jednostavna radnja (naredba)    | C:=x+y;<br>Ili<br>C:=abs(x-y);                                                                | Abs – apsolutna vrijednost |
|    | Odluka ili grananje             | Ako uvjet onda naredba 1<br>Inače naredba 2;                                                  |                            |
|    | Tok programa                    |                                                                                               |                            |
|  | Nastavak dijagrama              |                                                                                               |                            |

## 1. zadatak:

Algoritam za izračunavanje hipotenuze pravokutnog trokuta.

**početak**

**ulaz(a,b);**

**c:=sqrt(a<sup>2</sup>+b<sup>2</sup>);**

**izlaz(c);**

**kraj**

Ispravite!!



## 2. zadatak:

Algoritam od dva broja uvijek ispisuje veći broj.

**početak**

**ulaz(x);**

**ulaz(y);**

**ako x>y onda izlaz(x)**

**inače izlaz(y);**

**kraj**

### **3. zadatak:**

Algoritam preračunava kune u eure i US dolare.

**početak**

**ulaz(kune);**

**izlaz('Želiš eure?');**

**ulaz(odgovor);**

**ako odgovor='Da' onda izlaz(kune/7,1)**

**inače izlaz(kune/5,3);**

**kraj**

**4. zadatak:**

Algoritam izračunava trajanje utrke na temelju polaznog vremena (sat, minuta, sekunda) i završnog vremena.

- a) rezultat je prikazan u sekundama  
b) rezultat je prikazan u minutama i sekundama

npr. početak je u 13:40:20, a kraj u 14:30:10 → utrka je trajala 49 minuta i 50 sekundi

Očito je da i jedno i drugo vrijeme moramo prikazati u sekundama, a zatim izračunati razliku.

**početak**

```
ulaz(satp,minp,sekp);
```

```
ulaz(satz,minz,sehz);
```

```
sekundep:=satp*3600+minp*60+sekp;
```

```
sekundez:=satz*3600+minz*60+sehz;
```

**trajanje:=sekundez-sekundep;**

izlaz ('Utrka je trajala ',trajanje,'sekundi); a)

```
mintrajanje:=trajanje div 60;
```

**sektrajanje:=trajanje mod 60;**

izlaz ('Utrka je trajala ',mintrajanje,'minuta i ',sektrajanje,' sekundi);      b)

**kraj**

**div je operator cjelobrojnog dijeljenja → daje rezultat**

**mod je operator cjelobrojnog dijeljenja → daje ostatak**

Algoritam izračunava s koliko novčanica od 100, 10, 5 i 1 kune treba isplatiti gotovinu na bankomatu.

**početak**

**ulaz(kune);**

```
sto:=kune div 100;  
ostatak1:=kune mod 100;  
deset:= ostatak1 div 10;  
ostatak2:=ostatak1 mod 10;  
pet:=ostatak2 div 5;  
jedan:= ostatak2 mod 5;  
izlaz('100 * ', sto);  
izlaz('10 * ', deset);  
izlaz('5 * ', pet);  
izlaz('1 * ', jedan);
```

**kraj.**

Unesite vrijednost i ispišite izlaze



## Programske petlje prikazane pseudokodom

Programskom petljom smatramo isječak programa u kojem se određeni niz instrukcija ponavlja određeni broj puta. Broj ponavljanja može biti unaprijed zadan ili može ovisiti o nekom uvjetu.

Razlikujemo 3 vrste programskih petlji:

- petlja sa zadanim brojem ponavljanja
- petlja s uvjetom na vrhu (na početku)
- petlja s uvjetom na dnu (na kraju)

### 1. petlja sa zadanim brojem ponavljanja

**za kv:=početna vrijednost do završna vrijednost činiti**

**početak**

**instrukcije;**

**kraj**

- kv je kontrolna varijabla čija vrijednost se mijenja od početne do završne vrijednosti
- instrukcije se izvode za svaku vrijednost kontrolne varijable

## 2. petlja s uvjetom na vrhu

**dok uvjet činiti**  
**početak**  
**instrukcije;**  
**kraj**

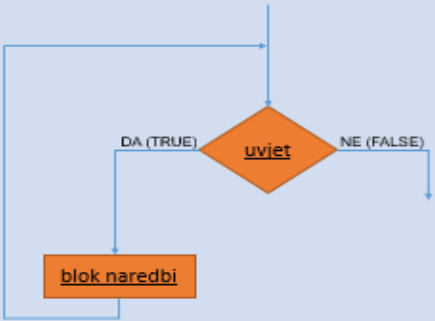
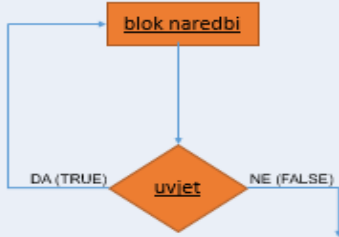
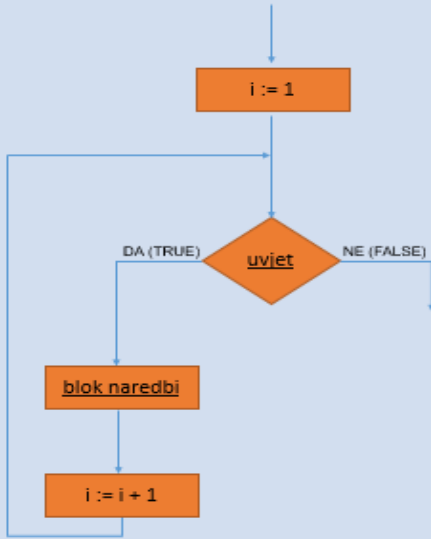
- instrukcije se izvode ako je uvjet istinit
- petlja se ponavlja sve dok je uvjet istinit
- može se desiti situacija da u prvom provjeravanju uvjet ne bude istinit tako da se instrukcije neće izvesti niti jednom

## 3. petlja s uvjetom na dnu

**ponavljati**

**instrukcije;**  
**dok uvjet;**

- instrukcije se izvode ako je uvjet neistinit
- petlja se ponavlja sve dok je uvjet neistinit
- u trenutku kad uvjet postane istinit petlja se prekida
- instrukcije se izvode barem jednom

|                                                                                                                                           |                                                                                           |                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>WHILE</b></p> <p>Petlja kod koje <b>nije unaprijed poznat broj ponavljanja</b>, a uvjet se provjerava <b>na početku</b> petlje</p>  | <p><b>dok je <u>uvjet</u> činiti:</b><br/><b><u>blok naredbi</u>;</b></p>                 |  <pre> graph TD     Start(( )) --&gt; Uvjet{uvjet}     Uvjet -- DA TRUE --&gt; BlokNaredbi[blok naredbi]     BlokNaredbi --&gt; Uvjet     Uvjet -- NE FALSE --&gt; Exit(( )) </pre>                                                               |
| <p><b>DO-WHILE</b></p> <p>Petlja kod koje <b>nije unaprijed poznat broj ponavljanja</b>, a uvjet se provjerava <b>na kraju</b> petlje</p> | <p><b>ponavljati:</b><br/><b><u>blok naredbi</u>;</b><br/><b>dok je <u>uvjet</u>;</b></p> |  <pre> graph TD     Start(( )) --&gt; BlokNaredbi[blok naredbi]     BlokNaredbi --&gt; Uvjet{uvjet}     Uvjet -- DA TRUE --&gt; BlokNaredbi     Uvjet -- NE FALSE --&gt; Exit(( )) </pre>                                                        |
| <p><b>FOR</b></p> <p>Petlja s unaprijed <b>poznatim brojem ponavljanja</b></p>                                                            | <p><b>za i := _ do _ činiti:</b><br/><b><u>blok naredbi</u>;</b></p>                      |  <pre> graph TD     Start(( )) --&gt; I1[i := 1]     I1 --&gt; Uvjet{uvjet}     Uvjet -- DA TRUE --&gt; BlokNaredbi[blok naredbi]     BlokNaredbi --&gt; Iplus[i := i + 1]     Iplus --&gt; Uvjet     Uvjet -- NE FALSE --&gt; Exit(( )) </pre> |

## Primjer zadatka s petljama

Primjeri:

1. Učitati 10 brojeva i izračunati njihovu sumu i njihov prosjek.
2. Učitati 20 brojeva i prebrojati koliko ima negativnih.
3. Ispisati tablicu prvih 100 prirodnih brojeva, njihovih kvadrata i njihovih korijena
4. Učitavati brojeve sve dok je njihova suma manja od 1000
5. Učitavati cijele pozitivne brojeve sve dok je njihov umnožak manji od 200 000
6. Treba generirati sljedeći niz: 1, 2, 4, 8, 16, 32, ... Treba ispisati sve članove koji su manji od 4000.
7. Treba učitati rečenicu (rečenica završava točkom). Algoritam ispisuje koliko riječi ima rečenica.  
    broj:=1;  
    ponavljaj  
        ulaz(znak);  
        ako znak=' ' onda broj:=broj+1;  
    dok znak='.';  
    izlaz('U rečenici ima ', broj, ' riječi');
8. Zadan je niz 1, 3, 9, 27, ... Treba ispisati prvi član niza veći od 10000.  
    clanniza:=1;  
    ponavljaj  
        clanniza:=clanniza\*3;  
    dok clanniza>10000;
9. Učitavanje podataka o učeniku sve dok korisnik želi.  
    ponavljaj  
        ulaz(ime, prezime);  
        ulaz(ocjena);  
        ulaz(vladanje);  
        izlaz('Želiš podatke za još jednog učenika?');  
        ulaz(odgovor);  
    dok odgovor='Ne';

## Nacrtati DTP za algoritme i programe u Pythonu

### Primjeri algoritama:

1. Algoritam ispisuje poruku: Hoću 5!

**početak**

**izlaz('Hoću 5!');**

**kraj**

2. Algoritam omogućava upis poruke, a zatim poruku prikazuje na ekranu.

**početak**

**ulaz(poruka);**

**izlaz(poruka);**

3. Algoritam omogućava unos dva broja, a zatim prikazuje njihov zbroj.

**početak**

**ulaz(broj1,broj2);**

**zbroj:=broj1+broj2;**

**izlaz(zbroj);**

**kraj**

4. Algoritam omogućava unos tri broja, a zatim prikazuje njihovu srednju vrijednost.

**početak**

**ulaz(broj1,broj2,broj3);**

**srednja:=(broj1+broj2+broj3)/3;**

**izlaz(srednja);**

**kraj**

5. Algoritam omogućava unos dva broja, a zatim prikazuje zbroj njihovih kvadrata.

**početak**

**ulaz(a,b);**

**zbroj:=a\*a+b\*b;**

**izlaz(zbroj);**

**kraj**

6. Algoritam izračunava opseg i površinu kruga (unos R).

```
početak  
  ulaz(r);  
  opseg:=2*3,14*r;  
  površina:=r*r*3,14;  
  izlaz(opseg, površina);  
kraj
```

7. Algoritam izračunava opseg i površinu kvadrata (unos stranicu a).

```
početak  
  ulaz(a);  
  opseg:=4*a;  
  površina:=a*a;  
  izlaz(opseg, površina);  
kraj
```

8. Algoritam izračunava opseg i površinu pravokutnika (unos stranice a i b).

```
početak  
  ulaz(a,b);  
  opseg:=2*a+2*b;  
  površina:=a*b;  
  izlaz(opseg, površina);  
kraj
```



11. Algoritam unosi dva broja i ispisuje njihov zbroj, razliku, umnožak, kvocijent, rezultat cjelobrojnog dijeljenja i ostatak kod cjelobrojnog dijeljenja.

**početak**

**ulaz(a,b);**

**zbroj:= a+b;**

**razlika:=a-b;**

**umnožak:=a\*b;**

**kvocijent:=a/b;**

**cjrez:=a div b;**

**cjost:=a mod b;**

**izlaz(zbroj,razlika,umnožak,kvocijent,cjrez,cjost);**

**kraj**

12. Algoritam pretvara sekunde u oblik sati, minute, sekunde.

**početak**

**ulaz(sekunde);**

**sat:=sekunde div 3600;**

**ostatak:=sekunde mod 3600;**

**minute:=ostatak div 60;**

**nove\_sekunde:=ostatak mod 60;**

**izlaz(sat,minute,nove\_sekunde);**

**kraj**

13. Algoritam unosi troznamenkasti broj, a zatim ispisuje tog znamenke broja.

**početak**

**ulaz(broj);**

**stotica:=broj div 100;**

**ostatak:=broj mod 100;**

**desetica:=ostatak div 10;**

**jedinice:=ostatak mod 10;**

**izlaz(stotica,desetica,jedinica);**

**kraj**

14. Algoritam omogućava unos dva broja, a zatim prikazuje manji broj.

**početak**

**ulaz(a,b);**

**ako a<b onda izlaz(a)**

**inače izlaz(b);**



## Grananje

15. Algoritam ispisuje apsolutnu vrijednost broja.

```
početak
    ulaz(a);
    ako a<0 onda početak
        r:=-1*a,
        izlaz(r);
    kraj
    inače izlaz(a);
kraj
```

16. Algoritam unosi broj i provjerava je li djeljiv sa 3.

```
početak
    ulaz(a);
    ako a mod 3=0 onda izlaz('Broj',a, 'je djeljiv s 3')
    inače izlaz('Broj',a, 'nije djeljiv s 3');
kraj
```

17. Algoritam unosi broj i ispisuje je li broj paran ili neparan.

```
početak
    ulaz(a);
    ako a mod 2=0 onda izlaz('Broj',a, 'je paran')
    inače izlaz('Broj',a, 'nije paran');
kraj
```

18. Algoritam unosi tri broja i provjerava zadovoljavaju li Pitagorin uvjet. Ispisuje odgovarajuću poruku.

```
početak
    ulaz(a,b,c);
    ako a*a +b*b=c*c onda izlaz('Brojevi',a, b,c,'su Pitagorini brojevi')
    inače izlaz('Brojevi',a, b,c,'nisu Pitagorini brojevi');
kraj
```

15. Algoritam ispisuje apsolutnu vrijednost broja.

**početak**

**ulaz(a);**

**ako  $a < 0$  onda početak**

**$r := -1 * a$ ,**

**izlaz(r);**

**kraj**

**inače izlaz(a);**

**kraj**

16. Algoritam unosi broj i provjerava je li djeljiv sa 3.

**početak**

**ulaz(a);**

**ako  $a \bmod 3 = 0$  onda izlaz('Broj',a, 'je djeljiv s 3')**

**inače izlaz('Broj',a, 'nije djeljiv s 3');**

**kraj**

17. Algoritam unosi broj i ispisuje je li broj paran ili neparan.

**početak**

**ulaz(a);**

**ako  $a \bmod 2 = 0$  onda izlaz('Broj',a, 'je paran')**

**inače izlaz('Broj',a, 'nije paran');**

**kraj**

18. Algoritam unosi tri broja i provjerava zadovoljavaju li Pitagorin uvjet. Ispisuje odgovarajuću poruku.

**početak**

**ulaz(a,b,c);**

**ako  $a^2 + b^2 = c^2$  onda izlaz('Brojevi',a, b,c,'su Pitagorini brojevi')**

**inače izlaz('Brojevi',a, b,c,'nisu Pitagorini brojevi');**

**kraj**

## Petlje

19. Algoritam unosi prirodni broj  $n$ , a zatim  $n$  puta ispisuje poruku: Dobar dan.

```
početak
    ulaz(n);
    za kv:=1 do n činiti
        izlaz('Dobar dan');
kraj
```

20. Algoritam unosi broj  $n$  i ispisuje sve brojeve od 1 do  $n$ .

```
početak
    ulaz(n);
    za kv:=1 do n činiti
        izlaz(kv);
kraj
```

21. Algoritam unosi prirodni broj  $n$  i ime korisnika, a zatim  $n$  puta ispisuje to ime.

```
početak
    ulaz(n);
    ulaz(ime);
    za kv:=1 do n činiti
        izlaz(ime);
kraj
```

22. Algoritam ispisuje sve prirodne brojeve od 1 do 100 i njihove kvadrate.

```
početak
    za broj:=1 do 100 činiti
        početak
            kvadrat:=broj*broj;
            izlaz(broj,kvadrat);
        kraj
kraj
```

23. Algoritam unosi prirodni broj  $n$ , a zatim ispisuje sve parne brojeve od 1 do tog broja.

```
    početak
      ulaz(n);
      za broj:=1 do n činiti
        početak
          ako broj mod 2 =0 onda izlaz(broj);
        kraj
      kraj
```

24. Algoritam unosi prirodni broj  $n$ , a zatim ispisuje sve brojeve djeljive sa 7 od 1 do tog broja.

```
    početak
      ulaz(n);
      za broj:=1 do n činiti
        početak
          ako broj mod 7 =0 onda izlaz(broj);
        kraj
      kraj
```

22. Algoritam ispisuje sve prirodne brojeve od 1 do 100 i njihove kvadrate.

```
početak
  za broj:=1 do 100 činiti
    početak
      kvadrat:=broj*broj;
      izlaz(broj,kvadrat);
    kraj
  kraj
```

23. Algoritam unosi prirodni broj n, a zatim ispisuje sve parne brojeve od 1 do tog broja.

```
početak
  ulaz(n);
  za broj:=1 do n činiti
    početak
      ako broj mod 2 =0 onda izlaz(broj);
    kraj
  kraj
```

24. Algoritam unosi prirodni broj n, a zatim ispisuje sve brojeve djeljive sa 7 od 1 do tog broja.

```
početak
  ulaz(n);
  za broj:=1 do n činiti
    početak
      ako broj mod 7 =0 onda izlaz(broj);
    kraj
  kraj
```

25. Algoritam unosi 10 brojeva i ispisuje njihovu sumu.

```
početak
  suma:=0;
  za kv:=1 do 10 činiti
    početak
      ulaz(broj);
      suma:=suma+broj;
    kraj
  izlaz(suma);
kraj
```

26. Algoritam zbraja sve brojeve od 1 do n (n unosi na početku) i ispisuje njihovu sumu.

```
početak
  suma:=0;
  ulaz(n);
  za broj:=1 do n činiti
    početak
      suma:=suma+broj;
    kraj
  izlaz(suma);
kraj
```

27. Algoritam unosi 10 brojeva i ispisuje njihov umnožak.

```
početak
  umnožak:=1;
  za kv:=1 do 10 činiti
    početak
      ulaz(broj);
      umnožak:=umnožak*broj;
    kraj
  izlaz(umnožak);
kraj
```

28. Algoritam zbraja sve neparne brojeve od 1 do n (n unosi na početku) i ispisuje njihovu sumu.

**početak**

**ulaz(n);**

**.....suma = 0;**

**za broj:=1 do n činiti**

**ako broj mod 2 = 1 onda suma:=suma+broj;**

**izlaz(suma);**

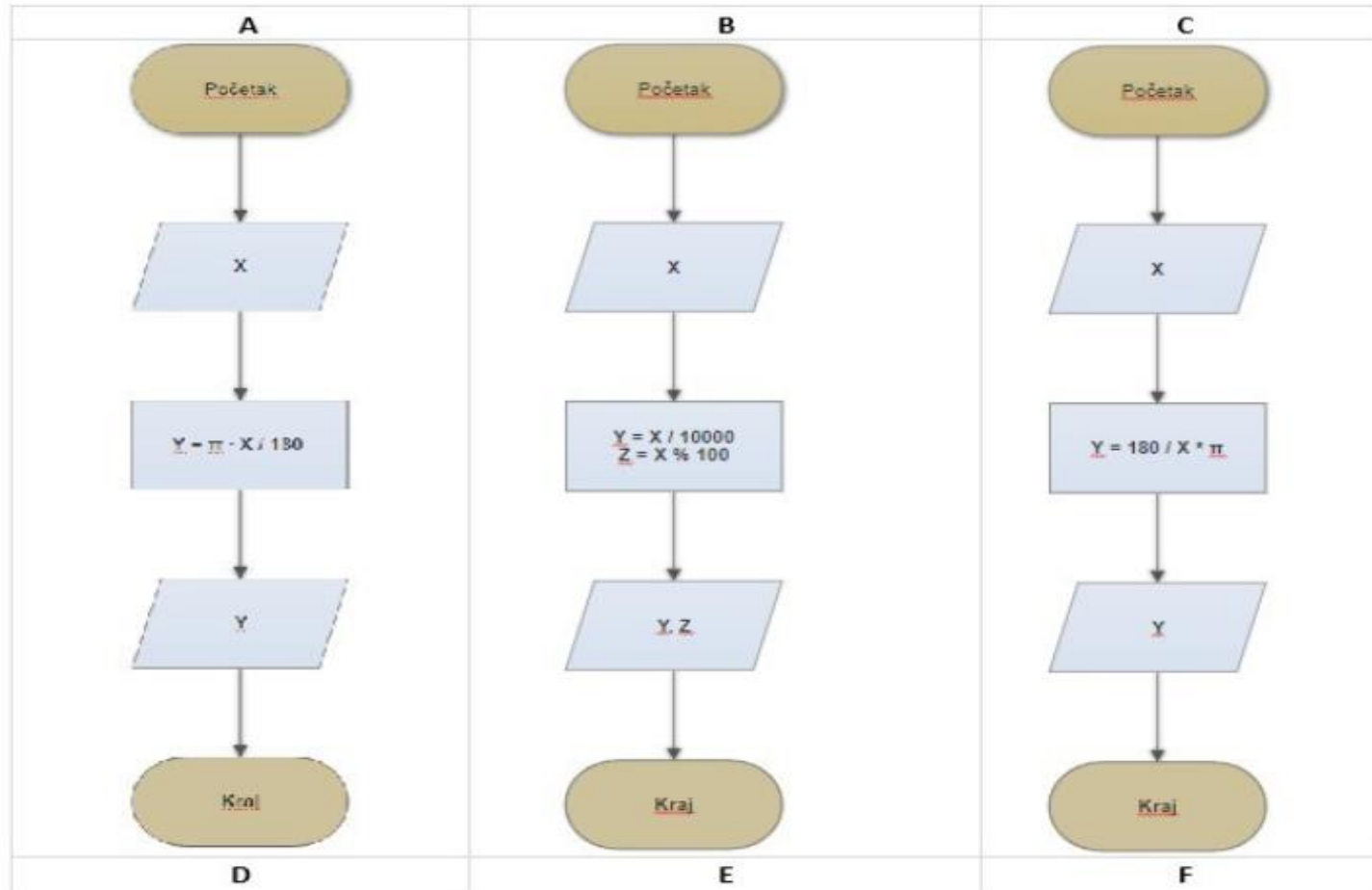
**kraj**



## Spajalica

Spoji tekst zadatka s pripadnim dijagramom toka.

- 1: Program učitava kut u stupnjevima te ga pretvara u radijane.
- 2: Program učitava četveroznamenkasti broj, a ispisuje samo znamenke stoticu i desetice.
- 3: Program učitava peteroznamenkasti broj, a ispisuje samo prvu i zadnju znamenku.



## Zadatak B

Spoji tekst zadatka s pripadnim algoritmom.

1: Maja je uočila da su čokolade na sniženju. Odlučila je kupiti što je moguće više čokolada, a ostatak novca ubaciti u kutiju na kojoj piše "za gladnu djecu Afrike". Program učitava broj kuna koje Maja ima te cijenu 1 čokolade, a ispisuje koliko je novaca Maja ubacila u kutiju.

2: Gusari su na otoku s blagom pronašli K zlatnih dukata. Osim kapetana ima još G gusara. Program učitava K i G i ispisuje koliko dukata će dobiti svaki gusar nakon što se blago pravedno raspodijeli na sve (svi dobivaju isti broj dukata, a kapetan dobiva svoj dio i ostatak koji se ne može razdijeliti).

3: Ivana je za rođendan od bake dobila B kuna, a od tete T kuna. Kako želi kupiti pernicu koja košta P kuna zamolila je djeda da joj posudi D kuna za tu kupovinu. Program ispisuje koliko kuna će joj djed posuditi.

| A                                                                                                | B                                                                              | C                                                                                                           |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br><u>izlaz</u> (a mod b);                                  | <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br>c := a div (b+1);<br><u>izlaz</u> (c); | <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br><u>ulaz</u> (c);<br>c := c - a;<br>c := c - b;<br><u>izlaz</u> (c); |
| D                                                                                                | E                                                                              | F                                                                                                           |
| <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br><u>ulaz</u> (c);<br>d := c - a - b;<br><u>izlaz</u> (d); | <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br>c := a mod b;<br><u>izlaz</u> (c);     | <u>ulaz</u> (a);<br><u>ulaz</u> (b);<br>c := (a+1) div b;<br><u>izlaz</u> (c);                              |

## Zadatak C

1: Program od tri unesena slova ispisuje prvo po abecedi.

```
a, b, c = input().split()
if a < b and a < c:
    print(a)
elif b < a and b < c:
    print(b)
else:
    print(c)
```

```
a, b, c = input().split()
if a > b and a > c:
    print(a)
elif b > a and b > c:
    print(b)
else:
    print(c)
```

2: Petar želi znati može li smjestiti šibicu na dno kutije ili ne. Napišite program koji će na temelju učitane duljine i širine kutije te duljine šibice ispisati True ako šibica stane u kutiju, a False ako šibica ne stane u kutiju.

```
a = int(input())
b = int(input())
c = int(input())
if (a**0.5+b**0.5)**2 <= c:
    print(True)
else:
    print(False)
```

```
a = int(input())
b = int(input())
c = int(input())
if (a**2+b**2)**0.5 >= c:
    print(True)
else:
    print(False)
```