

3.1.1 Grananja

Razgranati algoritmi realizuju se u programu primenom naredbe `if`. Naredba `if` može se javiti u tri oblika: jednogranom, dvogranom i višegranom (slika 3.2).

```
if uslov:
    blok_1
naredba_posle

if uslov:
    blok_1
else:
    blok_2
naredba_posle

if uslov_1:
    blok_1
elif uslov_2:
    blok_2
elif uslov_3:
    .....
    .....
elif uslov_n-1:
    blok_n-1
else:
    blok_n
naredba_posle
```

Slika 3.2: Naredba `if` i varijante grananja: jednograna (levo), dvograna (sredina) i višegrana (desno) struktura.

Jednograna naredba `if`

Struktura jednograne naredbe `if` prikazana je na slici 3.2, levo. Uslov između ključne reči `if` i dvotačke (`:`), predstavlja *logički izraz* čija vrednost može biti istinita (`True`), ili lažna (`False`). Logički izrazi grade se najčešće uz pomoć relacija poređenja i logičkih operacija `and`, `or` i `not`. Na primer, izraz koji testira da li tačka (x,y) pripada prvom kvadrantu glasi: $x \geq 0$ and $y \geq 0$. Da bi izraz bio tačan (tačka pripada prvom kvadrantu), oba uslova povezana operacijom `and` moraju biti ispunjena.

Ako je odgovarajući uslov naredbe `if` ispunjen, tok programa prebacuje se na *blok* naredbi `blok_1`. Blok predstavlja uzastopni niz naredbi koje čine jednu smislenu programsku celinu. Program, kao jedinstvena celina, tretira se kao sveobuhvatni blok. Sve naredbe bloka `if` moraju biti uvučene udesno, za jednak broj mesta, u odnosu na početak reda u kome se nalazi `if`. Prva naredba van bloka (`naredba_posle`), počinje pomerenom sleva u odnosu na blok, poravnato sa naredbom `if`. Ako uslov nije ispunjen, `blok_1` se preskače i izvršava se `naredba_posle`. Ova naredba izvršava se i kada je uslov ispunjen, ali pošto se prvo izvrše sve naredbe u bloku.

Naglašava se da svaki blok naredbe `if` može da sadrži proizvoljan broj *umetnutih* `if`

Naglašava se da svaki blok naredbe `if` može da sadrži proizvoljan broj *umetnutih* 1 blokova. Na taj način, tok programa nastavlja da se grana, u zavisnosti od odgovarajući logičkih uslova.

Prilikom formiranja novog bloka, po opšteprihvaćenoj *konvenciji*, njegove naredbe pomeraju se za *četiri mesta* u odnosu na roditeljski blok. Po navođenju dvotačka i pritisku na taster <Enter>, IDLE prelazi u novi red i formira novi blok, pomere nadesno za četiri mesta. Program, kao sveobuhvatni blok, *mora* započeti u prvoj kolozi

- ! Pomeranje (*indentacija*) svih naredbi bloka za jednak broj mesta udesno, u odnosu na roditeljski blok, predstavlja kontroverznu sintaksnu odliku jezika. Većina modernih jezika označava blok korišćenjem vitičastih zagrada (leva za početak, a desna za kraj bloka), čime se ostavlja programeru da po volji započne redove u izvornom kodu. Ukoliko se pravilo indentacije ne poštuje, Pajton interpreter prijavljuje sintaksnu grešku!

Problem 3.1 — Par-nepar. Igra par-nepar odvija se tako što računar “pita” igrača za celobrojni ulog koji stavlja u opticaj. Potom računar zamišlja broj x , a igrač pogađa da li je x paran ili ne. Pogodak igraču donosi, a promašaj uzima novac u visini uloga. Napisati program koji realizuje ovu igru. ■

Da bi se igra realizovala potrebno je da računar *može* da “zamisli” *slučajan* broj. Slučajni brojevi razmatraće se u glavi 6, a ovde će slučajnost biti *improvizovana* tako što će zamišljeni broj biti jednak unetoj visini uloga. Ovakvo ponašanje programa nije unapred poznato igraču, ali se predloženo pravilo može naslutiti posmatranjem ishoda većeg broja odigranih partija. U idealnom slučaju, kada računar zaista zamišlja slučajan broj, verovatnoća pogoška iznosi 50%. Kako igrači imaju tendenciju da nude “okrugle” uloge (10, 100, 150...), porednička strategija tipuje na paran broj. Sledi realizacija programa:

```
# igra par-nepar
```

Program 3.1 — Par-nepar.

```
1 # igra par-nepar
2 ulog = int(input('koliki je ulog (ceo broj): '))
3 pokusaj = int(input('par (0) ili nepar (1) '))
4
5 if pokusaj == 0 or pokusaj == 1:
6     if pokusaj == ulog % 2:
7         print('zaradili ste', ulog)
8     else:
9         print('izgubili ste', ulog)
10 else:
11     print('parnost mora biti 0 ili 1!')
```

Dvograna naredba `if` podrazumeva postojanje bloka naredbe `else`, koji se izvršava kada uslov nije ispunjen. Ako unos parnosti nije adekvatan (r5), izvršava se blok `else`, u kome se ispisuje prigodna poruka (r11). Za korektan unos testira se parnosti i ispisuje ishod partije (r6-9). Tom prilikom koristi se druga struktura tipa `if-else` koja je umetnuta u blok prve naredbe `if`.

Višegrana naredba `if-elif-else`

Kada postoji potreba da se, u zavisnosti od složene kombinacije uslova, izvrši *samo jedan* od tri ili više mogućih blokova, koristi se višegrana naredba `if-elif-else`. Njena primena biće ilustrovana na već poznatom problemu iz prethodne glave (2.4), gde je trebalo odrediti veći od dva uneta broja. Nedostatak prikazanog rešenja ogledao

Višegrana naredba if-elif-else

Kada postoji potreba da se, u zavisnosti od složene kombinacije uslova, izvrši *samo jedan* od tri ili više mogućih blokova, koristi se višegrana naredba if-elif-else. Njena primena biće ilustrovana na već poznatom problemu iz prethodne glave (2.4), gde je trebalo odrediti veći od dva uneta broja. Nedostatak prikazanog rešenja ogledao

se u nemogućnosti da se prikaže adekvatna poruka, za slučaj kada su brojevi jednaki. Varijanta koja koristi višegrnu naredbu if data je u sledećem prikazu:

```
# ispisuje veći od dva uneta broja
# ako su brojevi jednaki ispisuje prigodnu poruku

x = float(input('x= '))
y = float(input('y= '))

if x > y:
    print('veći broj je', x)
elif x < y:
    print('veći bro je', y)
else:
    print('brojevi su jednaki!')
```

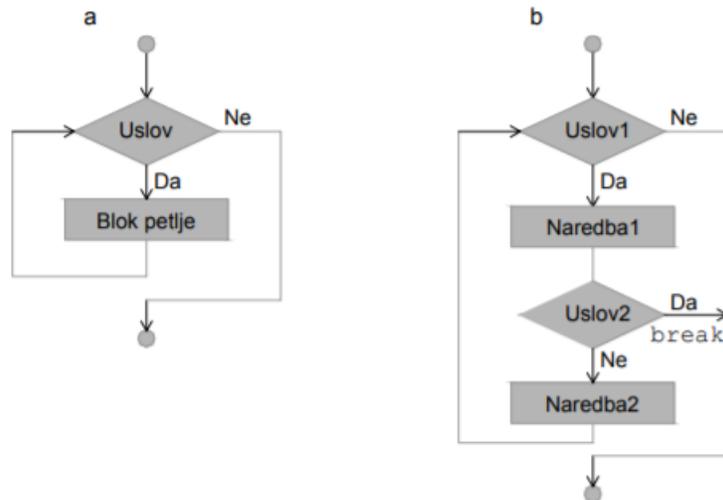
3.1.2 Petlje

Postupci koji zahtevaju višestruko ponavljanje pojedinih operacija realizuju se pomoću naredbi while i for.

3.1.2 Petlje

Postupci koji zahtevaju višestruko ponavljanje pojedinih operacija realizuju se pomoću naredbi `while` i `for`.

Petlja `while`



Slika 3.3: Dijagram toka petlje `while`: (a) testiranje uslova ostanka u petlji obavlja se pri vrhu; (b) petlja se može napustiti pre vremena, ako je ispunjen uslov za izvršavanje komande `break`.

Naredba `while` omogućava da se odgovarajući programski blok ponavlja u *petlji*,

Naredba `while` omogućava da se odgovarajući programski blok ponavlja u *petlji*,

sve dok je ispunjen *uslov ostanka* koji se navodi neposredno iza `while` (slika 3.3-a). Svaki prolaz kroz naredbe bloka petlje zove se *iteracija*. Program može da se zadrži u petlji beskonačno dugo, ako je uslov ostanka uvek ispunjen. Zbog toga je potrebno obezbediti da se ispunjenost uslova može *promeniti* unutar bloka petlje `while`. Petlja se uopšte ne izvršava samo ako uslov nije ispunjen pri prvom testiranju.

Problem 3.2 — Najveći broj u nizu. Napisati program kojim se sa tastature učitava niz brojeva. Unosi se broj po broj, sve dok se ne unese tekst koji označava pozitivnu beskonačnost ('inf'). Po završetku unosa treba prikazati najveći uneti broj. ■

Program 3.2 — Najveći broj u nizu.

```
1 # maksimalan broj u nizu
2
3 inf = float('inf') # inf ukazuje na specijalni objekat tipa
4                   # float koji označava + beskonačno(Infinity)
5 broj = float(input('unesite broj (inf za kraj) '))
6 if broj == inf:
7     print('niz je prazan')
8 else:
9     maks = broj
10    while broj != inf:
11        if broj > maks:
12            maks = broj
13        broj = float(input('unesite broj (inf za kraj) '))
14    print('najveći broj u nizu je', maks)
```

Zbirka+zadataka+za+6.razred

December 3, 2018

1 Zbirka zadataka: 6.razred

By:IVezjak

1.Napisi program koji unosi dva broja te ispisuje poruku jesu li brojevi jednaki ili razliciti.

```
In [8]: a=int(input('upisi prvi broj: '))
        b=int(input('upisi drugi broj: '))
        if a==b:
            print('brojevi su jednaki')
        else:
            print('brojevi su razliciti')
```

```
upisi prvi broj: 56
upisi drugi broj: 78
brojevi su razliciti
```

2.Napisi program koji unosi dva broja (input) te ispisuje poruku koji od ta dva broja je veci.

```
In [7]: a=int(input('upisi prvi broj: '))
        b=int(input('upisi drugi broj: '))
        if a>b:
            print('broj %s je veci!'%(a))
        else:
            print('broj %s je veci!'%(b))
```

```
upisi prvi broj: 567
upisi drugi broj: 342
broj 567 je veci!
```

3.Napisi program koji unosi jedan broj te ispisuje je li taj broj veci ili manji od 100.

```
In [6]: a=int(input('upisi broj: '))
        if a>100:
            print ('broj je veci od 100')
        else:
            print ('broj je manji od 100')
```

upisi broj: 567
broj je veci od 100

4. Napisi program koji unosi duljine stranica a i b te ispisuje poruku radi li se o kvadratu ili pravokutniku.

```
In [5]: a=int(input('duljina stranice a: '))
        b=int(input('duljina stranice b: '))
        if a==b:
            print ('Kvadrat')
        else:
            print ('Pravokutnik')
```

duljina stranice a: 6
duljina stranice b: 6
Kvadrat

5. Napisi program koji unosi rezultat nogometne utakmice za Spanjolsku i Hrvatsku te ispisuje tekst 'Spanjolska je pobjednik' u slucaju da je Spanjolska pobijedila, tekst 'Hrvatska je pobjednik' u slucaju da je Hrvatska pobijedila ili tekst 'Nerijeseno' u slucaju da su obje ekipe osvojile jednak broj bodova.

```
In [11]: h=int(input('Rezultat HR: '))
         s=int(input('Rezultat SP: '))
         if h>s:
             print ('Hrvatska je pobjednik')
         elif h<s:
             print ('Spanjolska je pobjednik')
         elif h==s:
             print ('Nerijeseno')
```

Rezultat HR: 5
Rezultat SP: 3
Hrvatska je pobjednik

6. Napisi program koji unosi jedan broj te ispisuje poruku je li ucitani broj paran ili ne.

```
In [1]: a=int(input('Upisi broj: '))
        if a%2==0:
            print ('Broj je paran')
        else:
            print ('Broj je neparan')
```

Upisi broj: 78
Broj je paran

7. Napisi program koji unosi jedan broj te ispisuje poruku je li ucitani broj djeljiv s 5 ili ne.

```
In [2]: a=int(input('Upisi broj: '))
        if a%2==0:
            print ('Broj je djeljiv s 5')
        else:
            print ('Broj nije djeljiv s 5')
```

Upisi broj: 50
Broj je djeljiv s 5

8. Napisi program koji unosi dob putnika te ispisuje poruku o cijeni karte prema pravilima: * osobe mlađee od 8 godina voze se besplatno * osobe od 8-18 godina placaju kartu 5 kn * osobe starije od 65 godina placaju kartu 1 kn * sve ostale osobe placaju kartu 8 kn

```
In [3]: a=int(input('Upii starosnu dob putnika: '))
        if a<8:
            print('karta je besplatna')
        elif a>=8 and a<18:
            print('karta je 5kn')
        elif a>65:
            print('karta je 1kn')
        else:
            print('karta je 8kn')
```

Upii starosnu dob putnika: 12
karta je 5kn

9. Napisi program koji provjerava duljinu lozinke. Ako upisana lozinka ima manje od 8 znakova, ispisat ce se poruka: Slaba lozinka. Ako lozinka ima 8 ili vise znakova ispisat ce se poruka: Jaka lozinka.

```
In [4]: a=input('Upisi lozinku: ')
        if len(a)<8:
            print ('Slaba lozinka')
        else:
            print ('Jaka lozinka')
```

Upisi lozinku: luka123
Slaba lozinka

10. Prosiri prethodni program za provjeru lozinke tako da se poruka 'Jaka lozinka' ispise u slucaju ako lozinka ima vise od 8 znakova i od toga barem jedan broj!

```
In [5]: a=input('Upisi lozinku: ')
        if len(a)<8 or a.isalpha():
            print ('Slaba lozinka')
        else:
            print ('Jaka lozinka')
```

```
Upisi lozinku: perouciinformatiku
Slaba lozinka
```

11. Ana i Tin sudjelovali su u kvizu znanja. Svako od njih odgovarao je na tri pitanja i za svako pitanje osvojili su 1,2 ili 3 boda. Napisi program koji e izracunati koliko bodova su ukupno osvojili te tko je pobjedio u kvizu! Osvojene bodove unesi pomocu tipkovnice.

```
In [1]: a1=int(input('Prvo pitanje Ana: '))
        a2=int(input('Drugo pitanje Ana: '))
        a3=int(input('Tree pitanje Ana: '))
        t1=int(input('Prvo pitanje Tin: '))
        t2=int(input('Drugo pitanje Tin: '))
        t3=int(input('Trece pitanje Tin: '))
        au=a1+a2+a3
        tu=t1+t2+t3
        print ('Ana je ukupno osvojila %s bodova.'%(au))
        print ('Tin je ukupno osvojio %s bodova.%(tu))
        if au>tu:
            print ('Ana je pobjednik!')
        else:
            print ('Tin je pobjednik!')
```

```
Prvo pitanje Ana: 5
Drugo pitanje Ana: 4
Tree pitanje Ana: 3
Prvo pitanje Tin: 5
Drugo pitanje Tin: 3
Tree pitanje Tin: 2
Ana je ukupno osvojila 12 bodova.
Tin je ukupno osvojio 10 bodova.
Ana je pobjednik!
```

12. Napisi program koji prevodi boje (barem tri) s hrvatskog na engleski jezik. Ukoliko upisemo nesto drugo osim boje, pojaviti e se poruka: Ta boja ne postoji u programu.

```
In [6]: b=input('Upisi boju: ')
        if b=='plava':
            print ('plava - blue')
        elif b=='zelena':
            print ('zelena - green')
```

```

elif b=='crvena':
    print ('crvena - red')
else:
    print ('Ta boja ne postoji u programu!')

```

Upisi boju: zelena
zelena - green

13.Napisi program koji unosi korisnicko ime te provjerava je li korisnicko ime upisano ili ne. Ukoliko je ispisat ce pozdravnu poruku, a ako nije ispisat ce poruku: "Niste upisali korisnicko ime."

```

In [8]: a=input('Upii korisnicko ime: ')
        if a=='':
            print ('Niste upisali korisnicko ime!')
        else:
            print ('Bok, %s!'%(a))

```

Upii korisnicko ime: Luka
Bok, Luka!

14.Napisi program *PigLatin prevoditelj* koji neku ucitanu rijec s tipkovnice mijenja tako da prvo slovo premjesti na kraj i potom na novu rijec doda nastavak -py. Program treba ispisati novu rijec.

```

In [9]: r=input('Upisi rijec: ')
        nastavak = 'py'

        if len(r) > 0 and r.isalpha():
            r1=r.lower()
            print ('Vaa rijec: ',r1)
            prvo=r1[0]
            nova=r1+prvo+nastavak
            nova=nova[1:]
            print ('Nova rijec: ',nova)

        else:
            print ('Niste upisali rijec!')

```

Upisi rijec: krava
Vaa rijec: krava
Nova rijec: ravakpy

15.Napisi program koji 5 puta ispisuje tvoje ime pomocu petlje for.

```

In [6]: for i in range(5):
        print ('Marko')

```

Marko
Marko
Marko
Marko
Marko

16. Napisi program koji 4 puta ispisiuje tvoje ime pomocu petlje for i zatim jednom tvoje prez-ime ispod svih imena.

```
In [10]: for i in range(4):  
         print ('Marko')  
         print ('Markovic')
```

Marko
Marko
Marko
Marko
Markovic

17. Napisi program koji pomocu petlje for ispisiuje brojeve od 1-10 jedan ispod drugog.

```
In [12]: for i in range(1,11):  
         print (i)
```

1
2
3
4
5
6
7
8
9
10

18. Napisi program koji pomocu petlje for ispisiuje brojeve od 1-10 jedan pored drugog.

```
In [14]: for i in range(1,11):  
         print(i, end = ' ')
```

1 2 3 4 5 6 7 8 9 10

19. Napisi program koji pomocu petlje for ispisiuje sve parne brojeve od 40 do 60 jedan pored drugoga.

```
In [16]: for i in range(40,61):  
         if i%2==0:  
             print(i, end = ' ')
```

40 42 44 46 48 50 52 54 56 58 60

20. Napisi program koji pomocu petlje for ispisuje brojeve djeljive s pet od 100-150 jedan ispod drugoga.

```
In [17]: for i in range(100,151):
         if i%5==0:
             print(i, end = ' ')
```

100 105 110 115 120 125 130 135 140 145 150

21. Napisi program koji ispisuje slovo po slovo, jedno ispod drugoga, pomocu naredbe for rijec 'sunce'.

```
In [19]: a='sunce'
         for i in a:
             print(i)
```

s
u
n
c
e

22. Napisi program koji za upisanu rijec ispisuje samo samoglasnike.

```
In [11]: a=input('Upii rijec: ')
         for i in a:
             if i in 'aeiouAEIOU':
                 print (i)
```

Upii rijec: informatika
i
o
a
i
a

23. Prosiri prethodni program tako da umjesto samoglasnika program prebroji i ispise broj samoglasnika u rijeci.

```
In [12]: a=input('Upii rijec: ')
         sam=0
         for i in a:
             if i in 'aeiouAEIOU':
                 sam+=1
         print ('Broj samoglasnika: ', sam)
```

```
Upii rijec: informatika
Broj samoglasnika: 5
```

24. Prosiri prethodni program tako da ispisuje broj samoglasnika, broj suglasnika i broj svih ostalih znakova.

```
In [13]: a=input('Upii rijec: ')
        sam=0
        sug=0
        ost=0
        for i in a:
            if i in 'aeiouAEIOU':
                sam+=1
            elif i.isalpha():
                sug+=1
            else:
                ost+=1
        print ('Broj samoglasnika: ', sam)
        print ('Broj suglasnika: ', sug)
        print ('Broj ostalih znakova: ', ost)
```

```
Upii rijec: #informatika24
Broj samoglasnika: 5
Broj suglasnika: 6
Broj ostalih znakova: 3
```

25. Napisi program za popularni *Ljubavni kalkulator*. Za dva ucitana imena (musko i zensko) program treba samostalno predvidjeti postotak sretne veze.

```
In [18]: import random
        a=input('Zensko ime: ')
        b=input('Musko ime: ')
        postotak = random.randint(1,100)
        print (postotak,'%')
```

```
Zensko ime: Petra
Musko ime: Petar
97 %
```

26. Napisi program koji slucajnim odabirom odabire jednog ucenika s popisa. Popis od nekoliko ucenika stavi u listu [].

```
In [28]: import random
        ucenici=['Pero', 'Ivo', 'Ana', 'Iva', 'Luka', 'Josipa']
        print (random.choice(ucenici))
```

Luka

27. Poznati Hobit Bilbo Baggins jednoga je dana na putu u *Shire* u sumi pronasao veliki lonac neobicnog napitka. Bio je to carobni napitak. **Kada god bi popio jednu casu napitka, Bilbo bi narastao za 16.46 cm.** Sto ga je cinilo vrlo sretnim, posto je poznato da je visok samo 112 cm. No morao je biti pazljiv: U jednom danu nije smio popiti vise od 7 casa napitka. Ako bi popio vise od 7 casa, smanjio bi se na velicinu komarca: 0.43 cm.

Ako znamo koliko je Bilbo casa napitaka popio na dan, napisi program koji ce izracunati i ispisati koliko je cm Bilbo bio visok na kraju dana. Visinu zaokruzi na dvije decimale.

Uz rezultat ispiši poruku koja opisuje njegovu visinu:

Ako je visina 0.43 ispiši se poruka KOMARAC.

Ako je visina >0.43 i visina <112 cm ispiši se poruka HOBIT

Ako je visina >112 i visina <190 ispiši se poruka COVJEK

Ako je visina >190 ispiši se poruka VILENJAK

28. Vinko i njegov prijatelj Janko smislili su novu igru. Prvi je na redu Vinko i n puta baca kockicu. Ako dobije paran broj, Janko mu mora dati 2 slicice nogometasa, ako dobije neparan broj, on Janku mora dati tri slicice nogometasa. Ako se dogodi da Vinko ili Janko ostaju bez slicica igra prestaje.

Napisi program koji ce ispisati koliko je ukupno slicica osvojio Vinko, koliko Janko te u redak ispod toga poruku tko je pobjednik Vinko ili Janko.

Ulazni podaci: Broj slicica koje posjeduje Vinko, broj slicica koje posjeduje Janko, broj bacanja kocke i u sljedecih n redaka b broj koji je Vinko dobio.

```
In [1]: v=int(input('Vinko:'))
        j=int(input('Janko:'))
        n=int(input())
        for i in range(n):
            b=int(input())
            if b%2==0:
                v=v+2
                j=j-2
                if j<=1:
                    v=v+1
                    j=0
                    break
            else:
                j=j+3
                v=v-3
                if v<=2:
                    j=j+2
                    v=0
                    break

        print (v,j)
```

```

if v>j:
    print ('VINKO')
else:
    print ('JANKO')

```

```

Vinko:20
Janko:24
3
4
3
7
16 28
JANKO

```

29. Jan se za vrijeme praznika preselio u Zagreb. Novi ljudi, novo susjedstvo i jako neobicna, napustena kuca na kraju ulice. Kuca je jako velika, kao dvorac. Jan je za vrijeme praznika cesto odlazio tamo i istraživao kucu. Kuca je imala jako puno soba oznacenih brojevima od 1 do n. Primjetio je nesto jako cudno - broj soba u kuci bi se svaki dan mijenjao. A svaki dan iz svake sedme sobe cuo bi se zvuk violine.

Kako bi pronasao neki predlozak tj. neko pravilo po kojemu se sobe mijenjaju Jan je odlucio napisati program, no treba mu i tvoja pomoc.

Potrebno je ucitati broj soba u kuci te na temelju toga ispisati: * u koliko soba ce se cuti zvuk violine * koliko soba ima paran broj

Ulazni podatak: ukupan broj soba

```

In [5]: n=int(input())
        p=0
        v=0
        for i in range(1,n+1):
            if i%2==0:
                p+=1
            if i%7==0:
                v+=1
        print ('Parnih soba:', p)
        print ('Zvuk violine:', v)

```

```

20
Parnih soba: 10
Zvuk violine: 2

```

30. Napisi program koji ucitava n brojeva te ispisuje:

- koliko od tih n brojeva je dijeljivo s tri
- koliko od tih n brojeva ima zadnju znamenku broj 6
- zbroj svih n brojeva
- najveći od ucitanih brojeva

```

In [11]: n=int(input())
        sest=0
        tri=0
        lista=[]
        for i in range(n):
            b=int(input())
            lista.append(b)
            if b%3==0:
                tri+=1
            b=str(b)
            if b[-1]=='6':
                sest+=1

        print ('Dijeljivo s tri:',tri)
        print ('Zadnja znamenka 6:',sest)
        print ('Zbroj brojeva:', sum(lista))
        print ('Najveci od ucitanih', max(lista))

```

```

4
36
9
26
1
Dijeljivo s tri: 2
Zadnja znamenka 6: 2
Zbroj brojeva: 72
Najveci od ucitanih 36

```

31. Napisi program koji e ucitani broj ispisati obrnuto.

```

In [19]: broj = int(input())
        obrnuti = 0
        while(broj > 0):
            ostatak = broj %10
            obrnuti = (obrnuti *10) + ostatak
            broj = broj //10

        print (obrnuti)

```

```

376
673

```

32. Napisi program koji za ucitano prvo slovo a i drugo slovo b ispisuje sva slova abecednim redom koja se nalaze izmedju a i b. Slova se trebaju ispisati u jednom retku.

```

In [20]: a = input()
        b = input()

```

```

br = ord(a)
while br <= ord(b):
    print(chr(br), end=" ")
    br += 1

```

d
h
d e f g h

33. Napisi program koji za ucitani broj n mnozi njegove znamenke sve dok ne dodje do jednoznamenkastog broja. Program treba ispisati koliko puta se mnozenje moze ponoviti.

```

In [21]: n = int(input())
         m = 1
         z=0
         if n < 10:
             print (0)
         else:
             while n > 0:
                 m = n%10 * m
                 n = n//10
                 z += 1
             print (z+1)

```

476
4

34. Napisi program u kojem korisnik unosi brojeve, sve dok ne unese 0. Odmah ispisujte samo dvoznamenkaste brojeve.

```

In [23]: while True:
         broj = int(input('Upisi broj: '))
         if broj > 9 and broj < 100:
             print (broj)
         elif broj == 0:
             break

```

Upisi broj: 5
Upisi broj: 34
34
Upisi broj: 6
Upisi broj: 24
24
Upisi broj: 0