

Termin XXIV

13. april 2019.

1. sumiraj_prvih_n.py

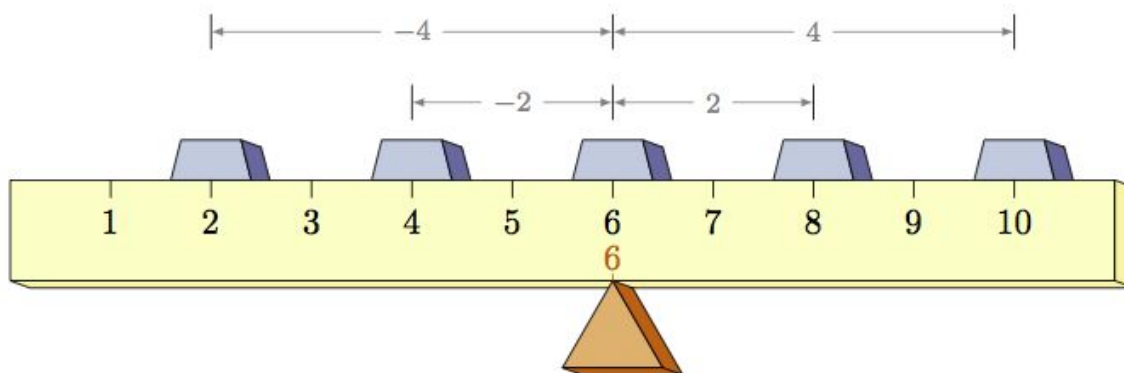
Napisati funkciju *sumiraj_prvih_n()* koja za prosleđen broj n vraća zbir svih brojeva od 1 do n . U glavnom delu programa uneti broj n i štampati vrednost funkcije *sumiraj_prvih_n()* kojoj se prosleđuje broj n kao argument.

2. stepenuj.py

Napisati funkciju *stepenuj()* koja za prosleđena dva broja a i b ispisuje vrednost b -tog stepen broja a . U glavnom delu programa uneti dva broja sa standardnog ulaza i pozvati funkciju *stepenuj()* kojoj se prosleđuju ta dva broja kao argumenti.

3. aritmeticka_sredina.py

Napisati funkciju *aritmeticka_sredina()* koja za prosleđen pozitivan n učitanih celih brojeva vraća aritmetičku sredinu tih unetih brojeva. U glavnom delu programa štampati vrednost funkcije *aritmeticka_sredina()*.



4. faktorisi.py

Napisati funkciju *faktorisi()* koja ispisuje sve faktore koji su prosti brojevi prosleđenog celog broja u istoj liniji. U glavnom delu programa pozvati funkciju *faktorisi()* za argumente 72, 36, 128 i 515.

$$72 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3$$
$$= 2^3 \cdot 3^2$$

↑ Prime number 2 occurs three times as factors

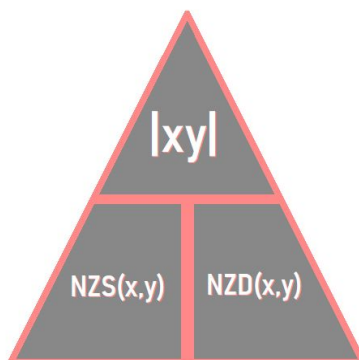
← Prime number 3 occurs two times as factors

5. nzd_nzs.py

Napisati funkcije:

- *NZD()* koja za dva prosleđena broja Euklidovim algoritmom (efikasniji način) određuje najveći zajednički delilac ta dva broja.
- *NZS()* koja za dva prosleđena broja određuje najmanji zajednički sadržalac traženjem najmanjeg proizvoda svih faktora ta dva broja.
- *invertuj()* koja vraća odgovarajući broj na osnovu datog trougala sa slike. Ona ima 3 argumenta, redom:
 - string promenljivu koja može biti ili "nzd" ili "nzs",
 - celobrojnu promenljivu koja predstavlja proizvod brojeva x i y,
 - celobrojnu promenljivu koja može biti ili NZD ili NZS.

U glavnom delu programa ispisati NZD i NZS za unete cele brojeve x i y, a zatim, koristeći funkciju *invertuj()* odrediti i ispisati traženu vrednost koja zavisi od toga da li se funkciji prosleđuje tekst "nzd" ili "nzs". *Primer: invertuj("nzd", 500, 25) -> određuje NZD za brojeve čiji je NZS 25.*



48, 28	2
24, 14	2
12, 7	2
6, 7	2
3, 7	3
1, 7	7
1, 1	

$$\text{NZS} = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \cdot 7 = 336$$

6. funkcije_provera.py

Šta će Python okruženje ispisati izvršavanjem sledećih komandi?

```
def vrati_uvecan(x=int):
    return x + 1

def vrati_umanjen(x=int):
    print(x - 1)

def funkcija3(x=int, y=int, z=int):
    return z, x, y

def funkcija4(x=int):
    x = x - 4
    print(x * 2)
    return x + 2








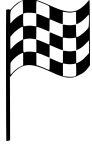
def funkcija5():
    n = int(input())
    for i in range(n):
        print(int(input()))
    return 10

print(vrati_uvecan(5))
print(vrati_umanjen(7))
print(funkcija3(4, 7, 12))
print(funkcija4(2))
funkcija5()
```



7. veverica_nije_tica.py

Napisati program koji učitava poziciju veverice Danice p koja predstavlja redni broj drveta na kojoj se nalazi ($1 \leq p \leq 100$). Računanje koliko je lešnika veverica sakupila se vrši u pomoćnoj funkciji koja treba da vrati broj sakupljenih lešnika. Veverica sakuplja ako prvo skoči napred za k koraka drveta, a zatim unazad za j koraka. *Napomena:* $p-j \neq 0$. Kako je Veverica Danica je fina i ona uzima samo polovinu. Ako je broj neparan, ta polovina je prvi manji ceo broj. Na standardnom izlazu ispisati koliko je lešnika sakupila.

										
p			$k1-j1$	$k1$		$k2-j2$	$k2$			end