

## Vežbe 2 - Zadaci

**Zadatak 1.** Realizovati funkcije za određivanje površine pravougaonika, površine trougla i dijagonale pravougaonika.

```
1 import math
2
3 def calculate_triangle_area(a, b, c):
4     s = 1/2*(a+b+c)
5     return math.sqrt(s*(s-a)*(s-b)*(s-c)) # **0.5
6
7 def calculate_rectangle_area(a, b):
8     return a*b
9
10 def calculate_rectangle_diagonal(a, b):
11     return math.sqrt(a**2 + b**2)
12
13 print(calculate_rectangle_area(5, 9))
14 print(calculate_rectangle_diagonal(10, 16))
15 print(calculate_triangle_area(4, 5, 3))
16 # Primer pogresnog unosa
17 # print(calculate_triangle_area(4,8,2))
```

**Zadatak 2.** Odrediti da li uneti broj sa tastature savršen. Savršeni broj je broj čiji je zbir delilaca jednak tom broju (uzimajući u obzir jedinicu). Npr. broj 28 je savršen broj, jer je  $1+2+4+7+14=28$ .

```
1 n = int(input("Unesi broj"))
2
3 delitelji = []
4
5 for i in range(1, n//2+1):
6     if n % i == 0:
7         delitelji.append(i)
8
9 if sum(deltitelji) == n:
10     print("Broj je savrsen")
11 else:
12     print("Broj nije savrsen")
```

**Zadatak 3.** Realizovati igru u kojoj računar „zamišlja“ broj u opsegu [0,1000]. Korisnik unosom sa tastature pokušava da pogodi zamišljenu vrednost. Na osnovu unete vrednosti, u konzoli se ispisuje povratna informacija da li je uneti broj veći ili manji od zamišljenog. Ako korisnik pogodi zamišljenu vrednost igra se završava uz poruku koji je bio i iz kog pokušaja je pronađen zamišljeni broj.

```
1 from random import randint
2
3 x = randint(0, 1000)
4 number = -1
5 cnt = 0
```

```

6
7 while x != number:
8     number = int(input('Unesite broj u opsegu [0,1000]: '))
9     cnt += 1
10    if number > x:
11        print('Zamisljen je manji broj od {}'.format(number))
12    elif number < x:
13        print('Zamisljen je veci broj od {}'.format(number))
14
15 print('Zamisljeni broj {}, pogodjen je iz {} puta'.format(number,cnt))

```

**Zadatak 4.** Odrediti vrednost broja  $e$  korišćenjem razvoja u red  $e = \sum_{n=0}^{\infty} \frac{1}{n!}$ .

```

1 # I nacin
2 def calculate_factorial(n):
3     if n == 0:
4         return 1
5     else:
6         result = 1
7         for i in range(1, n+1):
8             result *= i
9         return result
10
11 s = 0
12 for i in range(10):
13     s += 1/calculate_factorial(i)
14 print(s)
15
16 # II nacin
17 import math
18 def e(n=10):
19     return sum(1 / float(math.factorial(i)) for i in range(n))
20
21 print(e())

```

**Zadatak 5.** Odrediti korene kvadratne jednačine  $ax^2 + bx + c = 0$ , za zadate koeficijetne  $a$ ,  $b$  i  $c$ .

```

1 def koreni_jednacine(a, b, c):
2     diskriminanta = b**2-4*a*c
3
4     if diskriminanta != 0:
5         print('Koreni kvadratne jednacine su:')
6         print('x1= {:.3f}'.format((-b+diskriminanta**0.5)/(2*a)))
7         print('x2= {:.3f}'.format((-b-diskriminanta**0.5)/(2*a)))
8     else:
9         print('Koren kvadratne jednacine je:')
10        print('x1= {:.3f}'.format((-b)/(2*a)))
11

```

```
12 koreni_jednacine(2, 6, 2)
```

**Zadatak 6.** Odrediti sve brojeve blizance do unetog broja sa tastature, kao i razliku između blizanaca. Brojevi blizanci su svi prosti brojevi čija je razlika unapred definisana. Npr. brojevi blizanci sa razlikom dva su 3 i 5, 5 i 7, 17 i 19 itd.

```
1 def eratostenovo_sito(granica):
2     for i in range(2, granica):
3         if granica % i == 0:
4             return False
5     return True
6
7 def trazenje_blizanca(granica, razmak):
8     for i in range(2, granica):
9         j = i+razmak
10        if eratostenovo_sito(i) and eratostenovo_sito(j):
11            print('Blizanci su: {} i {}'.format(i, j))
12
13 max_broj = int(input('Unesite do kog broja odredjujete brojeve blizance'))
14 korak = int(input('Unesite razmak izmedju prostih brojeva: '))
15
16 trazenje_blizanca(max_broj, korak)
```

**Zadatak 7.** Ispitati monotonost niza.

```
1 number_array = []
2 n = int(input('Unesi broj clanova niza'))
3
4 for i in range(1, n+1):
5     element = int(input('Unesi {}. clan'.format(i)))
6     number_array.append(element)
7
8
9 def ispitivanje_monotonosti(niz):
10    if all(niz[i] <= niz[i+1] for i in range(len(niz)-1)):
11        print('Niz je monotono neopadajuci')
12    elif all(niz[i] >= niz[i+1] for i in range(len(niz)-1)):
13        print('Niz je monotono nerastuci')
14    elif all(niz[i] == niz[i+1] for i in range(len(niz)-1)):
15        print('Niz je stacionaran')
16    else:
17        print('Niz nije monoton')
18
19
20 ispitivanje_monotonosti(number_array)
```