

---

## Matematička radionica mladih - Informatika

# Termin VI

10. oktobar 2019.

### Redovni zadaci

#### 1. liste\_provera1.py

Koje od navedenih linija Pajton koda neće prouzrokovati grešku? Obrazložiti.

- 1) lista = [1, 2, "Pera", 4, 3.45]
- 2) lista = {0, 1, 2, 3, 2, 3}  
lista[0] = 5
- 3) lista = [0, 1, 2, 3]  
lista[1.0] = 4
- 4) lista = [0, 1, 2, 3]  
print(lista)
- 5) lista = [1, 2, "Pera", 4, 3.45]  
append.lista(4)
- 6) lista = [1, 2, 3, 4]  
print( lista.len() )
- 7) lista = [0, 1, 2, 3, 4]  
print( lista[ len(lista) ] )
- 8) list = lista()
- 9) my\_list = list()  
my\_list.append(4)  
my\_list.append(3)  
my\_list[ len(my\_list) - 1 ] = 10
- 10) my\_list = [ 1, 2, "Gile", 4, "Pera", 8]  
print( my\_list[ (len(my\_list) - 1) // my\_list[3] ] )
- 11) lista = [3, 8, 1, 6, 0, 8, 4]  
lista.index()
- 12) lista = [0, 1, 10, 11]  
print( 01 in lista )
- 13) lista = ["Plava", "Crvena", "Zelena"]  
print( lista[-2] )

## 2. liste\_provera2.py

Data je sledeća predefinisana lista: `lista = [1, 5, 3, 6, 8, 3, 5, 7]`. Šta će Pajton okruženje ispisati pri izvršavanju sledećeg koda? Obrazložiti.

- 1) `print( lista.index(6) )`
- 2) `print( lista[-1] - lista[1] )`
- 3) `for x in lista:`  
    `x += 1`  
    `print(lista)`
- 4) `for i in range(len(lista)):`  
    `lista[i] += 1`  
    `print(lista)`
- 5) `for i in range(2, len(lista) - 2, 3):`  
    `print(i)`
- 6) `print( lista[2: 4] )`
- 7) `for x in lista[::]:`  
    `print(x)`
- 8) `for x in lista[::3]:`  
    `print(x)`
- 9) `for x in lista[len(lista) : len(lista) + 1: 3]:`  
    `print(x)`
- 10) `i = 2`  
    `pom_lista = list()`  
    `duzina = len(lista)`  
    `while i < len:`  
        `pom_lista.append( lista[i - 1] )`  
        `i += 2`  
    `print(pom_lista)`

## 3. python\_funkcije.py

Napisati Pajton funkciju kojom se proverava da li je lista prosleđena kao argument funkciji prazna. Ako nije prazna ispisati njene elemente.

### PRIMER

#### ULAZ

```
glavna_lista = []
```

#### IZLAZ

```
Data lista je prazna.
```

## 4. razdvajanje\_reci.py

Napisati program koji učitava nekoliko reči razdvojenih donjim crtama, a zatim od takvog stringa uzeti samo karaktere sa parnim indeksima i od takvih napraviti nov string. Zatim formirati niz reči deljenjem tog string-a na osnovu donje crte.

### PRIMER

ULAZ	IZLAZ
zdravo_sвете_radi_tarana	['zrv', 'vt', 'aitrn']

## 5. prebrajanje\_slova.py

Napisati program koji učitava  $n$  reči sa standardnog ulaza i zatim broji koliko ukupno samoglasnika sadrže te reči. Podrazumevano je da su reči unošene malim slovima.

### PRIMER

ULAZ	IZLAZ
4	11
bicikla	
voda	
zemlja	
cokolada	

## 6. duzine\_reci.py

Napisati program koji generiše listu reči učitanih sa standardnog ulaza dok se ne unese "STOP", tako da nijedna reč te nove liste nema dužinu veću ili jednaku  $n$ .

### PRIMER

ULAZ	IZLAZ
cao	['cao', 'zmaj']
zdravo	
bravo	
dobar	
zmaj	
STOP	
5	

## 7. ubacivanje\_i\_izbacivanje\_elementa.py

Napisati program koji učitava 5 celih brojeva sa standardnog ulaza i pakuje ih u listu.

- 1) Ubaciti novi element, unet sa standardnog ulaza, posle prvog parnog elementa. Ukoliko lista nema parne elemente, dodati na kraj liste.
- 2) Izbaciti prvi neparni element iz liste.
- 3) Proširiti trenutnu listu novom listom sa elementima [1, 2, 3, 4].
- 4) Prebrojati broj ponavljanja elemenata sa vrednošću 4.
- 5) Štampati sortiranu listu.

## 8. deoba\_stringa.py

Napisati program koji rastavlja string koji se unosi sa standardnog ulaza i kreira listu elemenata od pojedinačnih reči tog string-a, gde se za rastavljanje slova u rečima koristi prvo slovo prve reči.

### PRIMER

#### ULAZ

Ovo je jedan veoma divan dan.

#### IZLAZ

['v', ' je jedan ve', 'ma divan dan']

**str = "HELLO"**

<b>H</b>	<b>E</b>	<b>L</b>	<b>L</b>	<b>O</b>
0	1	2	3	4

str[0] = 'H'

str[1] = 'E'

str[2] = 'L'

str[3] = 'L'

str[4] = 'O'

## Dodatni zadaci

### 9. kupovina.py

Poznate su sve cene predmeta koji se prodaju u jednoj prodavnici. Kupac ima na raspolaganju određeni iznos dinara i želi da kupi što skuplje predmete. Redom uzima predmete počev od najskupljeg, dok ima novca. Ako nema novca za najskuplji, uzima najskuplji za koji ima novca. Napomena: ova strategija ne garantuje da će predmeti koje kupi biti ukupno najveće moguće vrednosti (npr. Ako ima 5 dinara i ako su cene predmeta 4, 3 i 2 dinara, on će kupiti samo predmet od 4 dinara, a mogao bi da kupi predmete od 3 i 2 dinara). Napisati program koji određuje koliko novca kupcu preostaje nakon kupovine na opisan način. U prvoj liniji standardnog ulaza nalazi se iznos novca (ceo broj) koji ima kupac, u drugoj broj predmeta,  $N$  (prirodni broj manji od 50000), a zatim se u narednoj liniji unose redom cene predmeta, razdvojene razmacima. Na standardni izlaz ispisati preostali iznos novca, nakon kupovine na opisani način.

#### PRIMER

**ULAZ**

1250

5

1010 357 725 1125 115

**IZLAZ**

10

**ULAZ**

10000

6

3010 3005 5725 1265 2075 385

**IZLAZ**

0



## 10. dijagonale.py

Na jednoj neobičnoj šahovskoj tabli crna polja su raspoređena po dijagonalama, kao što je prikazano na slici (na slici je prikazano prvih 8 vrsta i 8 kolona, iako je tabla veća). Napisati program koji određuje boju polja na osnovu unetog rednog broja vrste i kolone. Sa standardnog ulaza se učitavaju redni brojevi vrste i kolone tri polja  $v$  ( $1 \leq v \leq 100$ ) i  $k$  ( $1 \leq k \leq 100$ ), razdvojeni jednim razmakom, za svako polje u posebnom redu. Na standardni izlaz ispisati boju polja (crno ili belo), za svako polje u posebnom redu.

	1	2	3	4	5	6	7	8
1			■			■		
2		■			■			■
3	■			■			■	
4			■			■		
5		■			■			■
6	■			■			■	
7			■			■		
8		■			■			■

### PRIMER

ULAZ	IZLAZ
1 5	belo
2 4	belo
3 4	crno