

## Crni biseri - Python (najčešće greške u kodovima)

1. Šta je rezultat rada sledećeg programa:

```
def transformacija(x):  
    return list(dict.fromkeys(x))  
  
lista = transformacija(["m", "a", "t", "e", "m", "a", "t", "i", "k", "a" ])  
  
print(lista)
```

2. Šta je rezultat rada sledećeg programa:

```
def funkcija(x):  
    return x[::-1]  
  
niska = funkcija("Stairway to Heaven 888")  
  
print(niska)
```

3. Šta je rezultat rada sledećeg programa:

```
niska = "Zdravo, 1, svete, 45!"  
print(niska.split(","))
```

4. Šta je rezultat rada sledećeg programa:

```
voce = ("jabuka", "banana", "lubenica")  
voce[1] = "dinja"  
print(voce)
```

5. Šta je rezultat rada sledećeg programa:

```
voce = ("jabuka", "banana", "lubenica")  
voce[3] = "dinja"  
print(voce)
```

6. Šta je rezultat rada sledećeg programa:

```
voce = {"jabuka", "banana", "lubenica"}  
for i in range(3):  
    print(voce[i])
```

7. Šta je rezultat rada sledećeg programa:

```
voce = ["jabuka", "banana", "lubenica"]  
for i in range(3):  
    print(voce[i])
```

8. EFIKASNA UPOTREBA += tj. **append** (efekat Schlemiel the Painter's algorithm)

<http://poincare.matf.bg.ac.rs/~jelenagr/1d/STL%20strikes%20again.pdf>

Uporedite operacije za nadovezivanje, tj. pokrenite naredni Python program

```
import time
```

```
n= 100000
```

```
start_time = time.time()
l = []
for i in range(n):
    l = l + [i * 2]
print(time.time() - start_time)
```

```
start_time = time.time()
l = []
for i in range(n):
    l += [i * 2]
print(time.time() - start_time)
```

```
start_time = time.time()
l = []
for i in range(n):
    l.append(i * 2)
print(time.time() - start_time)
```

PRIMER IZVRŠAVANJA  
94.03179216384888  
0.09375524520874023  
0.09375929832458496

Zaključak:

Operator "+" je i 1000 puta sporiji od metoda append.

Operator tzv. augmented assignment ("+=") je malo sporiji od metoda append.

Pojašnjenje: Schlemiel the Painter's algorithm

9. Šta je rezultat rada sledećeg programa:

```
f = open("readme.txt", "r")
print(f.read(15))
```

10. Šta je rezultat rada sledećeg programa:

```
f = open("readme.txt", "r")
for x in f:
    print(x)
```

11. Šta je rezultat rada sledećeg programa:

```
a = [100, 200, 3, 4]
b = [-1, -2, 11, 10]
c = [-1, -4, -5, -8]
rez1=list(map(lambda x, y : x+y, a, b))
```

```
print(rez1)
```

12. Šta je rezultat rada sledećeg programa:

```
a = [100, 200, 3, 4]
b = [-1, -2, 11, 10]
c = [-1, -4, -5, -8]
rez2=list(map(lambda x, y, z : x+y+z, a, b, c))
print(rez2)
```

13. Šta je rezultat rada sledećeg programa:

```
a = [100, 200, 3, 4]
b = [-1, -2, 11, 10]
c = [-1, -4, -5, -8]
rez3= list(map(lambda x, y, z : 1.5*x + 2.5*y - z, a, b, c))
print(rez3)
```

14. Šta je rezultat rada sledećeg programa:

```
voce = ("jabuka", "banana", "lubenica")
iteratorski = iter(voce)
```

```
print(next(iteratorski))
print(next(iteratorski))
print(next(iteratorski))
```

15. Šta je rezultat rada sledećeg programa:

```
voce = ("jabuka", "banana", "lubenica")
iteratorski = iter(voce)
```

```
print(next(iteratorski))
print(next(iteratorski))
print(next(iteratorski))
print(next(iteratorski))
```

16. Šta je rezultat rada sledećeg programa:

```
voce = ("jabuka", "banana", "lubenica")
iteratorski = iter(voce)
```

```
print(next(next(iteratorski)))
print(next(iteratorski))
print(next(iteratorski))
```

17. Šta je rezultat rada sledećeg programa:

```
import datetime
```

```
x = datetime.datetime.now()
```

```
print(x.year)  
print(x.strftime("%A"))  
print(x.strftime("%a"))  
print(x.strftime("%x"))
```