

Ponavljjanje nastavnih sadržaja iz informatike

TJEDAN PONAVLJANJA I PLANIRANJA



Što trebate ponavljati?

Dragi učenici 8.a razreda, ovaj tjedan ćemo ponavljati nastavno gradivo iz informatike koje se odnosi na nastavnu temu **6.B. Programski jezik Python**. Ova nastavna tema započinje na stranici 72 u udžbeniku, a završava na stranici 87 u udžbeniku.

Preporučam da pročitate i proučite tekst koji se odnosi na tu nastavnu temu, ali s obzirom na izvanredne okolnosti i težinu gradiva ja ću vam na sljedećim stranicama ovog dokumenta napisati i objasniti što trebate znati. Ovu nastavnu temu ću ocjenjivati!

Na kraju dokumenta se nalazi 5 zadataka koje trebate riješiti danas. Odgovore mi pošaljite u obliku privatnih poruka na Yammeru.

Što trebate znati?

1. Trebate znati koji će biti rezultati usporedbe dvaju slova ili dvije riječi. Imajte na umu da su za Python velika slova **manja** od malih slova!

```
>>> 'a' < 'b'  
True
```

Slovo *a* prije je po abecedi od slova *b*.

```
>>> 'marko' < 'mladen'  
True  
>>> 'MARKO' < 'MLADEN'  
True
```

Riječ *MARKO* prije je po abecedi od riječi *MLADEN*, pa Python vraća `True`, ali samo ako su riječi pisane velikim ili malim slovima.

```
>>> 'm' < 'M'  
False
```

Python razlikuje mala i velika slova. Prvo dolaze sva velika slova, pa onda mala. Vidi tablicu kodova na kraju udžbenika.

```
>>> 'Mladen' < 'marko'  
True  
>>> 'mARKO' < 'MLADEN'  
False
```

Ako usporedite *Mladen* i *marko*, dobit ćete `True` jer je $M < m$, ali ako usporedite *mAR-KO* i *MLADEN*, dobit ćete `False` jer nije $m < M$.

Što trebate znati?

2. Trebate znati na koji način funkcionira **algoritam razvrstavanja izborom**:

Birali smo najmanje slovo i stavili ga na početak novog znakovnog niza. Zatim se postupak ponavlja sa znakovnim nizom koji je kraći za jedan znak, sve dok zadani znakovni niz nije prazan. Pogledajte primjer razvrstavanja znakovnog niza od pet znakova.

zadani znakovni niz					prvi po abecedi	razvrstani znakovni niz
M	A	T	E	J	A	
M	T	E	J		E	A
M	T	J			J	AE
M	T				M	AEJ
T					T	AEJM
						AEJMT

Što trebate znati

3. Trebate znati što radi koja naredba u programu koji razvrstava znakove na temelju algoritma razvrstavanja izborom:

```
1. riječ = input('Upiši riječ: ')
2. sort = ''
3. while riječ != '':
4.     prvi = min(riječ)
5.     sort = sort + prvi
6.     mjesto = riječ.index(prvi)
7.     riječ = riječ[0:mjesto]+riječ[(mjesto+1):]
8. print(sort)
```

1. Upisuje podatak s tipkovnice i sprema u varijablu *riječ*
2. Definiramo varijablu *sort* koja sadrži prazan znakovni niz
3. Sve dok varijabla *riječ* ima znakova u sebi
4. U varijablu *prvi* stavljamo prvo slovo po abecedi iz riječi
5. Proširujemo varijablu *sort* tako da prvo slovo po abecedi iz riječi stavljamo na kraj te varijable
6. Tražimo redni broj prvog slova po abecedi iz riječi i to spremamo u varijablu *mjesto*
7. Prvo slovo po abecedi iz riječi uklanjamo iz riječi tako da riječ nakon ove naredbe ima jedan znak manje
8. Ispisujemo vrijednost varijable *sort*, a u toj varijabli se nalaze razvrstani znakovi. Ova naredba nije dio while petlje!

Što trebate znati?

4. Trebate znati što su rekurzivni potprogrami. To su potprogrami koji pozivaju sami sebe, odnosno potprogrami koji u sebi sadrže naredbu koja poziva isti taj potprogram.
5. Trebate znati što je dubina rekurzije. To je broj uzastopnih pozivanja istog potprograma (broj uzastopnih rekurzivnih poziva).
6. Trebate znati razliku između procedure i programske funkcije. **Procedura** je potprogram koji izvrši niz naredbi i ne vraća nikakvu vrijednost, dok je **programska funkcija** potprogram koji izvrši niz naredbi i vraća neku vrijednost (*return* je naredba za vraćanje neke vrijednosti).
7. Trebate na primjeru prepoznati je li potprogram procedura ili funkcija:

```
def potprogram(a) :  
    print(a)  
    return (a)
```

Ovo je **funkcija** jer se vraća vrijednost

```
def potprogram(a) :  
    print(a)  
    a = a + 1
```

Ovo je **procedura** jer se ne vraća vrijednost

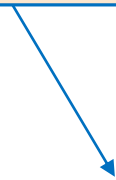
Što trebate znati?

8. Trebate znati prepoznati rekurzivan potprogram na temelju više ponuđenih potprograma:

```
def potprogram(n):  
    if n == 1:  
        print(1)  
    else:  
        print(n)  
        potprogram(n-1)
```

```
def potprogram(n):  
    if n == 1:  
        print(1)  
    else:  
        print(n)  
        n = n - 1
```

```
def potprogram(n):  
    if n == 1:  
        print(1)  
    else:  
        print('Rekurzija')
```



Ovo je rekurzivni potprogram jer sadrži rekurzivni poziv potprograma koji je naznačen plavim pravokutnikom

Što trebate znati?

9. Trebate znati napisati program s rekurzivnim potprogramom koji zbraja prvih n prirodnih brojeva. To je sljedeći program s rekurzivnim potprogramom:

```
def potprogram(n) :  
    if n == 1:  
        return 1  
    else:  
        return n + potprogram(n-1)  
  
n = input('Upišite broj n: ')  
n = int(n)  
potprogram(n)
```

Obratite pažnju na ovaj dio. Pomoću ovog uvjeta ograničavamo broj uzastopnih pozivanja rekurzivnog potprograma. Da nema tog uvjeta, potprogram bi se beskonačno pozivao!

Zadaci

1. Što će ispisati Python kada izvršimo sljedeću naredbu: `>>> 'vesna' < 'Veronika'`
2. Ako primijenimo algoritam razvrstavanja izborom na riječ MARKO, koji će biti rezultat (odnosno kako će izgledati razvrstani znakovni niz)?
3. Što radi naredba `mjesto = riječ.index(prvi)` u programu koji implementira algoritam razvrstavanja izborom?

4. Je li ovo programska funkcija?

```
def potprogram(a):  
    a = a - 1  
    print(a)
```

5. Je li ovo rekurzivan potprogram?

```
def potprogram(n):  
    print(n)  
    potprogram(n-1)
```