8 Python

For zanka, seznami in counterji

Pogledali si bomo vpeljavo **counter**-ja oz. **števca.** To je število, ki ga definiramo znotraj funkcije, ki nam nekaj prešteva.

Poglejmo si na primeru:
Funkcija **prestej_soda** sprejme seznam števil in nam pove, koliko števil je sodih.

```
> def prestej_soda(sez):
>     counter = 0                    #nastavimo števec na 0
>     for stevilo in seznam:         #sprehodimo se čez seznam
>         if število % 2 == 0:       #preverimo, če je število sodo
>             counter += 1           # če je, povečamo števec za 1
>     return counter                 # na koncu vrnemo števec
```

Preverili smo vsa števila v seznamu in za vsako najdeno sodo število smo povečali števec za 1 (dobesedno smo jih šteli), bo končni števec enak ravno številu sodih števil v seznamu.

```
> seznam = [1, 2, 3, 4, 5, 6, 7, 8, 9]
> prestej_soda(seznam)
4
```

1. Naloga:
Definiraj funkcijo **prestej_enice**, ki sprejme seznam števil in vrne število enic v seznamu.

Definiraj še funkcijo **preštej_stevila**, ki spreme seznam števil in število n ter nam vrne kolikokrat se število n ponovi v seznamu.

2. Naloga:
Definiraj funkcijo **enako_dolga**, ki sprejme dva seznam in nam vrne *True*, če sta seznama enako dolga oz. *False*, če nista.
Namig: Pomagaj si z vgrajeno funkcjio ***len(),*** ki nam pove dolžino niza.

```
> a = "niz"
> len(a)
3
```

3. Naloga:
Definiraj funkcijo **prestej_crke**, ki sprejme seznam nizov in nam vrne število posameznih črk (torej nizov dolžine 1).

Definiraj še funkcijo **prestej_besede**, ki sprejme seznam nizov in nam vrne število besed, to so vsi nizi, katerih dolžina je več kot 1.

4. Naloga:
Definiraj funkcijo **ohrani_liha**, ki sprejme seznam števil in nam vrne nov seznam, ki vsebuje samo liha števila iz prvotnega seznama.

```
> seznam = [1, 2, 3, 4, 5, 6, 7, 8, 9]
> ohrani_lihe(seznam)
[1, 3, 5, 7, 9]
> seznam
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

5. Naloga:
Definiraj funkcijo **nastavi_enako_dolzino**, ki sprejme dva seznama števil ter vrne izboljšanega drugega tako, da bo enako dolg kot prvi. Če je drugi prvotno krajši od prvega, naj funkcija drugega dopolni z ničlami tako, da bosta enako dolga. Če bo prvotni drugi daljši od prvega, pa naj ga odreže toliko, da bosta enako dolga. Če sta enaka že na začetku, vrne drugega.

```
> sez1 =  [1, 2, 3, 4, 5]
> sez2 = [1, 2, 3]
> sez3 = [1, 2, 5]
>  nastavi_enako_dolzino(sez1, sez2)
[1, 2, 3, 0, 0]
> nastavi_enako_dolzino(sez2, sez1)
[1, 2, 3]
> nastavi_enako_dolzino(sez2, sez3)
[1, 2, 5]
```

6. Naloga:
Definiraj funkcijo **vsota**, ki sprejme dva enako dolga seznama števil in izračuna vrne seznam, kjer je prvi element v vrnjenem seznamu vsota prvih dveh elementov prejetih seznamov.

Primer:

> sez1 = [1, 3, 4]
> sez2 = [2, 4, 2]

$$sez1 + sez2 = [1, 3, 4] + [2, 4, 2]$$
$$= [1 + 2, 3 + 4, 4 + 2]$$
$$= [3, 7, 6]$$

> vsota(sez1, sez2)
[3, 7, 6]

7. Naloga:
Definiraj funkcijo **skalarni_produkt**, ki sprejme dva enako dolga seznama števil in nam vrne vsoto produktov istoležečih členov.

Primer vsote produktov istoležečih člemov:

> sez1 = [1, 3, 4]
> sez2 = [2, 4, 2]

$$sez1 \cdot sez2 = [1, 3, 4] \cdot [2, 4, 2]$$
$$= 1 \cdot 2 + 3 \cdot 4 + 4 \cdot 2$$
$$= 2 + 12 + 8$$
$$= 22$$

> skalarni_produkt(sez1, sez2)
22