

Laboratorijska vežba 1 - Python

Napomena: Svaki zadatak realizovati u okviru jedne funkcije.

1. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **parni**, koja određuje broj parnih elemenata zadate liste.

Primer: `parni([1, 7, 2, 4, 5]) = 2`

- b. **poredak**, koja 2 liste brojeva objedinjuje u jednu listu koja se sastoji od tuple objekata koji imaju tri elementa. Prvi element rezultujuće kolekcije je element prve liste, drugi je element druge liste a treći ma vredost 'Jeste' ukoliko je element druge liste dva puta veći od elementa prve liste, odnosno 'Nije' ukoliko nije. Dužina liste je dimenzija duže od dve ulazne liste. N-ti tuple podatak rezultujuće kolekcije čine n-ti brojevi iz obe ulazne liste, gde na prvoj poziciji treba da se nađe manji, a na drugoj veći broj iz obe liste.

Kraću ulaznu listu dopuniti sa kraja brojem 0, dok dužine obe liste ne budu iste. Zabranjeno je korišćenje petlji.

Primer: `poredak([1, 7, 2, 4], [2, 5, 2]) = [(1, 2, 'Jeste'), (7, 5, 'Nije'), (2, 2, 'Nije'), (4, 0, 'Nije')]`

2. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **numlista**, koja iz liste koja može da sadrži elemente različitog tipa izdvaja samo numeričke vrednosti

Primer: `numlista(["Prvi", "Drugi", 2, 4, [3, 5]]) = [2, 4]`

spojidict, koja 2 liste obejkata objedinjuje u listu sa elementima tipa dictionary. Dužina liste je dimenzija duže od dve ulazne liste. N-ti dictionary element rezultujuće kolekcije čine n-ti objekti iz obe ulazne liste, gde se na prvoj poziciji nalazi element prve liste uparen sa ključem 'prvi', a na drugoj poziciji element druge liste uparen sa ključem 'drugi'. Kraću ulaznu listu dopuniti sa ' - ', dok dužine obe liste ne budu iste. Zabranjeno je korišćenje petlji.

Primer: `spojidict([1, 7, 2, 4], [2, 5, 2]) = [{'prvi': 1, 'drugi': 2}, {'prvi': 7, 'drugi': 5}, {'prvi': 2, 'drugi': 2}, {'prvi': 4, 'drugi': '-'}]`

3. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **uredi**, koja svaki od prvih n elemenata uvećava za definisanu vrednost a preostale umanjuje za definisanu vrednost. Funkciji se prosleđuje lista koja sadrži samo numeričke vrednosti i vrednost koja treba da se uvećava, odnosno umanjuje. Zabranjeno je korišćenje petlji.

Primer: `uredi([1, 2, 3, 4, 5], 3, 1) = [2, 3, 4, 3, 4]`

- b. **spoji**, koja 2 liste brojeva objedinjuje u jednu listu koja se sastoji od tuple objekata koji imaju tri elementa. Prvi element rezultujuće kolekcije je element prve liste, drugi je element druge liste a treći je zbir elemenata. Dužina liste je dimenzija duže od dve ulazne liste. N-ti tuple podatak rezultujuće kolekcije čine n-ti brojevi iz obe ulazne liste, gde na prvoj poziciji treba da se nađe manji, a na drugoj veći broj iz obe liste.

Kraću ulaznu listu dopuniti sa kraja brojem 0, dok dužine obe liste ne budu iste. Zabranjeno je korišćenje petlji.

Primer: `spoji([1, 7, 2, 4], [2, 5, 2]) = [(1, 2, 3), (5, 7, 12), (2, 2, 4), (4, 0, 4)]`

4. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

a. **zbir**, koja kreira novu listu čiji su elementi zbirovi susednih elementa liste.

Primer: `zbir([1, 2, 3, 4, 5]) = [3, 5, 7, 9]`

b. **suma**, koja vraća sumu svih elemenata u listi brojeva i svim njenim podlistama. Zabranjeno je korišćenje petlji i funkcije `sum`.

Primer: `suma([[1, 2, 3],[4, 5, 6],[7, 8, 9]]) = 45`

5. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

a. **brojel**, koja broji koliko elemenata ima svaka podlista liste koja joj je prosleđena. Ukoliko element liste nije podlista funkcija vraća -1.

Primer: `brojel([1, 2], [3, 4, 5], 'el', ['1', 1]) = [2, 3, -1, 2]`

b. **proizvod**, koja računa proizvod liste podlisti (A) i liste brojeva (B). Smatrati da je broj podlisti u listi A jednak dužini liste B. Funkcija vraća listu koja ima onoliko elemenata koliko je dužina ulaznih listi. N-ti element izlata liste predstavlja sumu svih elemenata N-te podliste liste A koji u prethodno pomnoženi N-tim elementom u liste B. Zabranjeno je korišćenje petlji i funkcije `sum`.

Primer: `proizvod([[1,2,3],[4,5,6],[7,8,9]], [1,2,3]) = [6, 30, 72]`

6. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

a. **razlika**, koja prihvata dve liste (bilo kog tipa podataka), a ima povratnu vrednost koja je lista sastavljena od svih elemenata iz prve liste, koji se ne nalaze u drugoj listi.

Primer: `razlika([1, 4, 6, "2", "6"], [4, 5, "2"]) = [1, 6, "6"]`

b. **objedini**, koja 2 liste brojeva objedinjuje u jednu listu koja se sastoji od parova brojeva (tuple). Dužina liste treba da je dimenzija duže od dve ulazne liste. N-ti tuple podatak rezultujuće kolekcije čine n-ti brojevi iz obe ulazne liste, gde na prvoj poziciji treba da se nađe manji, a na drugoj veći broj iz obe liste.

Kraću ulaznu listu dopuniti sa kraja brojem 0, dok dužine obe liste ne budu iste.

Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `objedini([1, 7, 2, 4, 5], [2, 5, 2]) = [(1, 2), (5, 7), (2, 2), (0, 4), (0, 5)]`

7. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

a. **operacija**, koja listu tuple vrednosti transformiše u listu brojeva, koji se dobijaju primenom operacije prosleđene putem argumenta.

Primer: `operacija([(1, 4, 6), (2, 4), (4, 1)], lambda x, y: x + y) = [11, 6, 5]`

b. **objedini**, koja od liste tuple objekata kreira dictionary. Prvi element svakog tuple objekta postaje ključ rečnika, dok sve ostale vrednosti postaju vrednost (lista vrednosti).

Ukoliko tuple podatak ima manje od 2 vrednosti, na mesto vrednosti postaviti `None`.

Ukoliko ključ već postoji u rečniku, prepisati njegovu vrednost novom.

Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `objedini([(1), (3, 4, 5), (7), (1, 4, 5), (6, 2, 1, 3)]) = { 1: [4, 5], 3: [4, 5], 7: None, 6: [2, 1, 3] }`

8. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

a. **izmeni**, koja svaki n-ti element liste zamenjuje brojem koji predstavlja sumu svih elemenata originalne liste, od prvog, do n-tog elementa.

Primer: `izmeni([1, 2, 4, 7, 9]) = [1, 3, 7, 14, 23]`

- b. **izracunaj**, koja u listi koja može da se sastoji od brojeva i podlisti, na n-tom mestu u rezultujućem nizu upisuje broj sa n-te pozicije iz ulaznog niza, a ukoliko se radi o podlisti, zamenjuje je proizvodom svih brojeva u podlisti.
Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `izracunaj([1, 5, [1, 5, 3], [4, 2], 2, [6, 3]]) = [1, 5, 15, 8, 2, 18]`

9. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **prosek**, koja za svaki element prosleđene liste, koja se sastoji isključivo od podlisti, vraća aritmetičku sredinu svih njenih vrednosti.

Primer: `prosek([[1, 4, 6, 2], [4, 6, 2, 7], [3, 5], [5, 6, 2, 7]]) = [3.25, 4.75, 4.0, 5.0]`

- b. **zamena**, koja u listi brojeva, brojeve manje od broja x, koji se prosleđuje kao argument, zamenjuje zbirom svih vrednosti ulazne liste, koje imaju indeks veći od indeksa broja koji se obrađuje.

Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `zamena([1, 7, 5, 4, 9, 1, 2, 7], 5) = [35, 7, 5, 19, 9, 9, 7, 7]`

10. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **izbroj**, koja određuje broj elemenata liste, gde svaki element može da bude podlista ili broj.
Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `izbroj([1, [1, 3, [2, 4, 5, [5, 5], 4]], [2, 4], 4, 6]) = 13`

- b. **stepen**, koja svaki par dva broja u ulaznoj listi (x, y), transformiše u x^y .

Primer: `stepen([1, 5, 2, 6, 1, 6, 3, 2, 9]) = [1, 25, 64, 6, 1, 216, 9, 512]`

11. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **razlika**, koja kreira novu listu čiji su elementi razlika susednih elementa liste.

Primer: `razlika([8, 5, 3, 1, 1]) = [3, 2, 2, 0]`

- b. **proizvod**, koja vraća proizvod svih elemenata u listi brojeva i svim njenim podlistama. Zabranjeno je korišćenje petlji i funkcije `prod`.

Primer: `proizvod([[1, 3, 5], [2, 4, 6], [1, 2, 3]]) = 4320`

12. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **presekok**, koja prihvata dve liste (bilo kog tipa podataka), a ima povratnu vrednost koja je lista sastavljena od svih elemenata koji se nalaze u obe liste.

Primer: `razlika([5, 4, "1", "8", 3, 7], [1, 9, "1"]) = [1, "1"]`

- a. **izracunaj**, koja u listi koja se sastoji od brojeva i podlisti, menja svaki broj njegovim kvadratom, dok listu zamenjuje zbirom kvadrata brojeva koji je čine.

Zabranjeno je korišćenje petlji (osim u comprehension sintaksi).

Primer: `izracunaj([2, 4, [1, 2, 3], [4, 2], 2, [9, 5]]) = [4, 16, 14, 20, 4, 106]`

13. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **izmeni**, koja kreira novu listu tako da elemente na parnim pozicijama uvećava za jedan, dok elemente na neparnim pozicijama umanjuje za 1.

Primer: `izmeni([8, 5, 3, 1, 1]) = [9, 4, 4, 0, 0]`

- b. **skupi**, koja kreira novu listu tako da svaka dva susedna elementa liste čiji su elementi isključivo podliste zamenjuje podlistom koja sadrži zbir elemenata na odgovarajućim pozicijama. Zabranjeno je korišćenje petlji.

Primer: `skupi([[1, 3, 5],[2, 4, 6],[1, 2]]) = [[3, 7, 11],[3, 6, 6]]`

14. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **unija**, koja prihvata dve liste (bilo kog tipa podataka), a ima povratnu vrednost koja je lista sastavljena od svih elemenata obe liste bez ponavljanja.

Primer: `unija([5, 4, "1", "8", 3, 7], [1, 9, "1"]) = [5, 4, "1", "8", 3, 7, 9, 1]`

- a. **suma**, koja formira novu listu tako što nalazi sumu svih elemenata u podlistama. Zabranjeno je korišćenje petlji i funkcije `sum`.

Primer: `suma([[1, 2, 3],[4, 5, 6],[7, 8, 9]]) = [6, 15, 24]`

15. Korišćenjem programskog jezika Python, napisati sledeće funkcije:

- a. **izdvoji**, koja iz zadate liste čiji su elementi liste, izdvaja n-ti element i formira rezultujuću listu, pri čemu je $n=0$ za prvu podlistu i uvećava se sukcesivno za 1. Ukoliko ne postoji n-ti element u listi vraća se 0.

Primer: `izdvoji([5, 4, 4], [1, 9, 1], [5, 6], [4, 6, 10, 12]) = [5, 9, 0, 12]`

- b. **promeni**, koja u listi brojeva, brojeve veće ili jednake broju x , koji se prosleđuje kao argument, umanjuje za x , dok brojeve manje od x uvećava za x .

Primer: `promeni([7, 1, 3, 5, 6, 2], 3) = [4, 4, 6, 2, 3, 5]`