

Datoteke

Korištenjem fajla `studenti.txt` riješiti sledeće zadatke:

1. Napisati program koji iščitava ove podatke i zapisuje ih u konzolu.
2. Programski odrediti koliko linija ima u fajlu `studenti.txt`.
3. Napisati program koji računa koliko osoba pod imenom Marko se nalazi u fajlu.
4. Napisati program koji iz zadatog fajla ispisuje samo prezimena studenata.
5. Programski odrediti koliko bodova je imao student koji se preziva „Jovanovic“.
6. Napisati program koji čita fajl `studenti.txt`, iz njega izvlači samo poene i upisuje ih u drugi fajl `bodovi.txt` u proizvoljnom formatu.
7. Napisati program koji iščitava fajl `bodovi.txt` i provjerava da li je iko od studenata imao 100 poena.
8. Napisati program koji iščitava fajl `bodovi.txt` i računa koliko su bodova studenti imali u prosjeku.
9. Odrediti koliko studenata će dobiti ocjenu 10.
NAPOMENA: za ocjenu 10 potrebno je da student ostvari 91 poen ili više.
10. Odrediti ime i prezime studenta koji ima najviše bodova.

Try/except

1. Napisati funkciju `korjenuj` koja od korisnika traži da unese pozitivan broj, a vraća vrijednost njegovog kvadratnog korijena. Ako korisnik unese nevalidnu vrijednost, potrebno je baciti izuzetak i ispisati poruku: “Uneseni broj nije validan.”
2. Napisati funkciju `saberi vrijednosti` koja kao parametar prima rječnik, a vraća zbir vrijednosti tog rječnika. Ako dođe do greške prilikom sabiranja, vratiti `None`.
3. Napisati funkciju `ispis` u kojoj se od korisnika traži da unese prirodan broj `n`. Zadatak funkcije je da ispiše sve prirodne brojeve manje od `n`. Ako korisnik unese nevalidnu vrijednost za `n`, ispisati mu poruku.

Na primjer, ako korisnik unese broj 4, potrebno je ispisati brojeve:

- 1
- 2
- 3

4. Napisati funkciju koja od korisnika traži da unese decimalni broj, a zatim ispisuje vrijednost tog broja zaokruženog na prvi veći (nemanji) cijeli broj. Ako korisnik unese nevalidnu vrijednost, ispisati mu odgovarajuću poruku.
5. Napisati funkciju **dnevnik** koja kao parametar prima listu imena đaka u dnevniku. Unutar funkcije se traži od korisnika da unese redni broj učenika kojeg želi da pogleda. Ako korisnik pokuša da pristupi učeniku van opsega, ispisati mu poruku o grešci. Na primjer, ako je **dnevnik** = [„Marko“, „Ana“, „Maja“, „Iva“], a korisnik unese 1 jer želi da pristupi prvom učeniku, funkcija treba da ispiše „Marko“. Ako korisnik unese 10 jer želi 10. učenika, funkcija treba da ispiše poruku o grešci.

Kolekcije

1. Napisati funkciju **uduplaj** koja kao parametar prima torku brojeva, a vraća listu brojeva koji su jednaki originalnim elementima torke pomnoženim sa 2. Na primjer, ako su elementi ulazne torke bili 4, 5, 6, 2, onda elementi izlazne liste treba da budu 8, 10, 12, 4.
2. Napisati funkciju **prebroj** koja kao parametar prima torku brojeva. Unutar funkcije se od korisnika traži da unese broj. Povratna vrijednost funkcije je broj pojavljivanja unijetog broja u ulaznoj listi. Na primjer, ako je ulazna torka (6, 4, 6, 3), a korisnik unese 6, funkcija treba da vrati 2. Ako korisnik unese 10, funkcija treba da vrati 0.
3. Napisati funkciju **objedini** koja kao parametre prima tri vrijednosti, a vraća rječnik kod koga su ključevi riječi jedan, dva i tri, a vrijednosti prvi, drugi i treći proslijeđeni parametar.
4. Napisati funkciju **zaokruži na veći** koja kao parametar prima listu brojeva, a vraća listu kod koje su elementi jednaki ulaznim elementima, ali su zaokruženi na prvi manji (neveći) cijeli broj. Na primjer, za ulaz [5.6, 0.6, -2.2], funkcija treba da vrati [5, 0, -3]

5. Napisati funkciju `ispis` koja kao parametar prima torku brojeva, i njen zadatak je da ispiše elemente te torke u jednom redu, bez zagrada. Na primjer, ako je ulazni parametar torka (5, 6, 7, 2), fukcija treba da ispiše 5, 6, 7, 2.
6. Napisati funkciju `saberi` koja kao parametar prima torku brojeva, a kao rezultat vraća zbir elemenata te torke.
7. Napisati funkciju `nađi dva najveća` koja kao parametar prima listu prirodnih brojeva, a vraća torku koja se sastoji iz drugog i prvog najvećeg broja te liste. Ako lista ima manje od 2 elementa, ispisati poruku: "Lista nema dovoljno elemenata". Na primjer, za parametar `[5, 3, 7, 0, 6]` potrebno je vratiti `(6, 7)`.
8. Napisati funkciju `spoj rječnike` koja kao parametre prima dva `rječnika` proizvoda i njihovih kolicina, a kao rezultat vraća novi rječnik koji se sastoji od svih proizvoda i ukupne količine.
Na primjer, ako su ulazni rječnici bili

```
rječnik1 = {
    "kafa": 5,
    "cokolada": 2
}
rječnik2 = {
    "voda": 4,
    "kafa": 3
},
```

Rezultujući rječnik treba da bude

```
rječnik = {
    "kafa": 8,
    "cokolada": 2,
    "voda": 4
}
```
9. Napisati funkciju `provjeri` koja kao parametar prima rječnik čiji su elementi oblika riječ: broj slova. Zadatak funkcije je da provjeri da li je rječnik pravilno napravljen, odnosno da li su sve vrijednosti rječnika duzine odgovarajućih ključeva. Ako jesu, funkcija treba da vrati `True`, a ako nisu, `False`.
Na primjer, ako se funkciji proslijedi `rječnik = {"Ana":3, "Pera": 4, "Iva": 3}`, ona treba da vrati `True`

Ako se proslijedi `rječnik = {"Ana":3, "Pera": 7, "Iva": 3}`, funkcija treba da vrati `False` jer riječ "Pera" ima 4, a ne 7 slova.

10. Napisati funkciju `ima li duplikata` koja kao parametar prima listu stringova i vraća `True` ako u njoj postoje duplikati, `False` u suprotnom.
Na primjer, u listi 5, 3, 5, 6 ima duplikata, dok u listi 5, 7, 9 nema.

Moduli

1. Napisati funkciju `generiši nasumičan` broj koja je bez parametara. U funkciji se od korisnika traži da unese dva broja, `a` i `b`. Zadatak funkcije je da generiše i ispiše nasumičan broj iz intervala `[a, b]`. Za generisanje nasumičnog broja koristiti funkciju `randint` iz modula `random`.
2. Korištenjem funkcije `randint` iz modula `random` napisati funkciju `odaberi slučajan broj` koja kao parametar prima listu brojeva, a ispisuje jedan nasumično odabran element te liste.
3. Napisati funkciju `baci novčić` koja nema parametara. Zadatak funkcije je da na proizvoljan nasumičan način ispiše „pismo“ ili „glava“ svaki put kada se funkcija pokrene.
4. Napisati funkciju `tangens` koja kao parametar prima veličinu ugla `n`. Zadatak funkcije je da ispiše sledeću poruku: „Tangens ugla od `[n]` radijana je `[tg(n)]`“.
5. Napraviti skriptu `rotiranje.py` i u sklopu nje:
 - a. Napisati funkciju `rotiraj lijevo` koja kao parametar prima listu. Rezultat funkcije je lista kod koje su svi elementi rotirani jedno mjesto ulijevo. Prvi element treba da dođe na posljednju poziciju u listi.
Na primjer, ako je ulaz `[1, 2, 3, 4]`, izlaz treba da bude `[2, 3, 4, 1]`

- b. Napisati funkciju `rotiraj desno` koja kao parametar prima listu. Rezultat funkcije je lista kod koje su svi elementi rotirani jedno mjesto udesno. Posljednji element treba da dođe na prvu poziciju u listi.

Na primjer, ako je ulaz `[1, 2, 3, 4]`, izlaz treba da bude `[4, 1, 2, 3]`

Napraviti skriptu `provjera.py` i u sklopu nje:

- a. Kreirati proizvoljnu listu. Importovati funkciju za lijevo rotiranje i rotirati listu za jedno mjesto ulijevo. Ispisati dobijenu listu.
- b. Importovati funkciju za desno rotiranje i rotirati listu za dva mjesta udesno. Ispisati dobijenu listu.