

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 2 -

Sastaviti program na programskom jeziku Python, kojim se vrši određena vrsta obrade nad nizom celih brojeva. Glavni program treba da:

- a) učitava sve potrebne podatke (dužinu niza/nizova, elemente niza/nizova, itd.);
- b) izvrši zahtevanu obradu;
- c) ispisuje sve dobijene rezultate;

Zavisno od rednog broja problema, svaki student treba da sastavi **jedan** od programa koji su dati u prilogu ovog dokumenta.

Radi boljeg testiranja programa, odabrati nekoliko dodatnih skupova podataka sa kojima će program biti testiran. Svaki primer treba da sadrži ulazne podatke i očekivani izlaz za te podatke.

Napomene:

1. Rok za predaju drugog domaćeg zadatka je **sreda, 02.12.2020.** putem kursa predmeta na **Moodle** platformi za elektronsko učenje. Tačan termin za predaju će biti naknadno definisan za sve studente. Termin će biti ograničenog vremenskog trajanja.
2. Domaći zadaci će biti testirani i ocenjivani korišćenjem javnih i tajnih testova.
3. Studentima će nekoliko dana pre roka za predaju biti dostupno okruženje za testiranje rešenja domaćeg zadatka na Moodle platformi za elektronsko učenje korišćenjem javnih testova.
4. Prilikom predaje domaćeg zadatka studenti će rešavati i kratak test znanja u vezi rešenja domaćeg zadatka i relevantnog gradiva iz oblasti programiranja na programskom jeziku Python koje obuhvata temu domaćeg zadatka.
5. Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preduzmu odgovarajuće disciplinske mere.
6. Svi drugi detalji oko predaje, ocenjivanja i odbrane domaćeg zadatka će biti blagovremeno objavljeni.
7. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):

$$i = (R + G) \bmod 8$$

8. Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:

- `dz2.py`, koja sadrži izvorni tekst programa na programskom jeziku Python;

0. Napisati program koji proverava da li je od zadate liste cifara moguće napraviti palindrom. Palindrom je lista brojeva za koju vazi da je $a[i] = a[n - i - 1]$ za svaki element liste. Ukoliko je moguće napraviti program treba da ispiše vrednost **True**, u suprotnom treba da ispiše **False**.

Program treba da:

- 1) Učita listu cifara.
- 2) Proveri da li svaki element liste predstavlja jednocifren broj. Ukoliko ne predstavlja, prekinuti izvršavanje programa i ne ispisivati ništa.
- 3) Vrš obradu po opisanom postupku.
- 4) Ispiše dobijeni rezultat nakon završetka obrade na prethodno opisani način.

Primeri

ULAZ	IZLAZ
1 2 1 4 2 0 4	True
1 2 3 4 5	False
1 3 5 3 5 1	True

1. Napisati program koji za zadatu listu brojeva računa i ispisuje aritmetičku sredinu svih brojeva u listi. Nakon toga, program treba da pronade i ispiše najmanji element koji je veći ili jednak aritmetičkoj sredini i najveći element koji je manji ili jednak aritmetičkoj sredini svih brojeva.

Program treba da:

- 1) Učita listu brojeva.
- 2) Računa aritmetičku sredinu brojeva u listi.
- 3) Pronade i ispiše najmanji element koji je veći ili jednak izračunatoj aritmetičkoj sredini
- 4) Pronade i ispiše najveći element koji je manji ili jednak izračunatoj aritmetičkoj sredini

Primeri

ULAZ	IZLAZ
1 5 15 4 2 7 9	6.14 7 5
3 3 3	3.00 3 3
1 2 3	2.00 2 2

2. Napisati program koji učitava listu brojeva, a zatim napravi i ispiše novu listu brojeva koji sadrži samo brojeve koji su se u početnoj listi nalazili tačno jedanput. Redosled brojeva treba biti isti kao u ulaznoj listi. Nakon toga, program treba da ispiše sve parove brojeva, u novoj listi, za koje važi da je zbir brojeva veći od proizvoda brojeva. Parove brojeva ispisati sortirano po indeksu (ako se u nekom redu ispisuju brojevi $a[i]$ i $a[j]$, a u narednom redu brojevi $a[k]$ i $a[l]$, treba da važi da je $i < k$ ili $i = k, j < l$).

Program treba da:

- 1) Učita listu brojeva.
- 2) Napravi listu brojeva koji se u početnoj listi pojavljuju tačno jedanput uz zadržavanje redosleda kao u ulaznoj listi.
- 3) Ispiše sve parove brojeva iz nove liste, za koje važi da je zbir brojeva veći od proizvoda brojeva. Parove brojeva ispisati na gore opisani način.

Primeri

ULAZ	IZLAZ
5 1 -3 4 2 1 1 4	[5, -3, 2] 5 -3 -3 2
1 1 2 2 3 3	[]
1 1 2	[2]

3. Napisati program koji od zadate liste brojeva napravi novu listu u kome se prvo nalaze svi brojevi deljivi sa 3, zatim svi parni brojevi, a na kraju svi preostali neparni brojevi. Redosled brojeva iz iste grupe ne treba menjati.

Program treba da:

- 1) Učita listu brojeva.
- 2) Napravi novu listu brojeva prema gore opisanom kriterijumu.
- 3) Ispiše rezultujuću listu.
- 4) Ispiše broj pojavljivanja elemenata prema gore definisanim grupama.

Primeri

ULAZ	IZLAZ
5 8 15 23 8 27 2 3 13 30	[15, 27, 3, 30, 8, 8, 2, 5, 23, 13] 4 3 3
1 2 3	[3, 2, 1] 1 1 1
6 12 8 7 9 17 11	[6, 12, 9, 8, 7, 17, 11] 3 1 3

4. Napisati program koji iz liste pozitivnih celih brojeva izbacuje sve brojeve koji zadovoljavaju opisani uslov. Iz liste treba izbaciti sve brojeve za koje važi da je suma cifara datog broja podignutih na stepen jednak broju cifara datog broja jednaka istom tom broju (npr. $153 = 1^3 + 5^3 + 3^3$ ili $8208 = 8^4 + 2^2 + 0^4 + 8^4$).

Program treba da:

- 1) Učita listu brojeva.
- 2) Ukloni iz liste brojeve prema gore opisanom kriterijumu.
- 3) Ispiše rezultujuću listu.

Primeri

ULAZ	IZLAZ
8208 5 156 153	[156]
1634 5 6	[]
548834 11 725	[11, 725]

5. Napisati program koji vrši obradu liste decimalnih cifara zadatog broja. Iz liste je potrebno ukloniti jednu cifru zadatom broju, tako da rezultat bude najveći broj deljiv brojem 2^n , gde se n unosi sa standardnog ulaza i važi $n \geq 1$. Broj je deljiv sa 2^n ukoliko je broj koji čine poslednjih n cifara deljiv brojem 2^n , što je dozvoljeno proveriti operacijom ostatka pri deljenju. Broj n se unosi pre samih cifara.

Program treba da:

- 1) Učita listu brojeva.
- 2) Proveri da li svaki element liste predstavlja jednocifren broj. Ukoliko ne predstavlja, prekinuti izvršavanje programa i ne ispisivati ništa.
- 3) Ukloni iz liste jednu cifru prema gore opisanom kriterijumu.
- 4) Ispiše rezultujuću listu.

Primeri

ULAZ	IZLAZ
2 1 2 3 4	[1, 2, 4]
1 1 4 8 6	[4, 8, 6]
2	[]

6. Napisati program koji iz liste celih pozitivnih brojeva izbacuje sve brojeve koji nisu k -hipersavršeni. Broj n je k -hipersavršen ako za njega važi jednakost $n = 1 + k * (\sigma(n) - n - 1)$, gde je $\sigma(n)$ predstavlja sumu svih celobrojnih delioca broja n . Parameter k se unosi sa standardnog ulaza pre samog niza.

Program treba da:

- 1) Učita listu brojeva.
- 2) Proveri da li svaki element liste predstavlja pozitivan broj. Ukoliko ne predstavlja, prekinuti izvršavanje programa i ne ispisivati ništa.
- 3) Ukloni iz liste sve brojeve koji ne zadovoljavaju opisani kriterijum.
- 4) Ispiše rezultujuću listu.

Primeri

ULAZ	IZLAZ
2 1 3 5 6 21 28	[21]
1 1 3 5 6 21 28	[6, 28]
3 103 214 325	[325]

7. Napisati program koji na osnovu liste celih brojeva formira novu listu koja će za svaki element polazne liste sadržati vrednost njegove aditivne perzistencije. Aditivna perzistencija broja je jednaka broju puta koji treba ponoviti operaciju sabiranja cifara na dobijeni rezultat sabiranja da bi se dobio jednocifren broj (npr. aditivna perzistencija broja 2718 je jednaka 2: $(2 + 7 + 1 + 8) = 18$, $(1 + 8) = 9$).

Program treba da:

- 1) Učita listu brojeva.
- 2) Formira novu listu koja sadrži vrednost aditivne perzistencije za svaki element ulazne liste.
- 3) Ispiše rezultujuću listu.

Primeri

ULAZ	IZLAZ
1 3 5 6 21 28	[0, 0, 0, 0, 1, 2]
99993 21 28	[3, 1, 2]
103 214 325	[1, 1, 2]