

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 4 -

Sastaviti program na programskom jeziku Python, kojim se vrši određena vrsta obrade nad tekstualnim podacima (stringovima). Glavni program treba da:

- učitava jednu ili više linija teksta sa kojim će raditi;
- poziva odgovarajuću funkciju koja izvrši zahtevanu obradu nad zadatim tekstom;
- ispisuje sve dobijene rezultate;

Zavisno od rednog broja problema, svaki student treba da sastavi jedan od programa koji su dati u prilogu ovog dokumenta. Voditi računa o dekompoziciji programa na potprograme prema gore navedenoj raspodeli i zahtevima pojedinačnih zadataka. Po potrebi, dozvoljeno je uvoditi i dodatne potprograme u rešenje.

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:

- Rok za predaju četvrtog domaćeg zadatka je **sreda, 23.12.2020.** 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.
- Domaći zadaci će biti testirani i ocenjivani korišćenjem javnih i tajnih testova.
- 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.
- 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.
- Domaći zadaci se rešavaju **samostalno**. Predmetni nastavnici zadržavaju pravo da nakon predaje domaćih zadataka izvrše proveru sličnosti i preduzmu odgovarajuće disciplinske mere.
- Svi drugi detalji oko predaje, ocenjivanja i odbrane domaćeg zadatka će biti blagovremeno objavljeni.
- Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):

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

- Kao rešenje domaćeg zadatka potrebno je predati sadržaj sledeće datoteke:

- dz4.py**, koja sadrži izvorni tekst programa na programskom jeziku Python;

16.12.2020. godine

sa predmeta

0. Napisati program koji računa vrednost jednostavnog aritmetičkog izraza zadatog u obliku stringa. Format stringa je dat u primeru. Smatrati da su operandi pozitivni celi brojevi, a da se kao aritmetički operatori mogu pojaviti samo sabiranje (+), množenje (*) i stepenovanje (**exp**). Najveći prioritet ima stepenovanje, pa množenje, pa sabiranje. Pretpostaviti da se u izrazu neće pojaviti dva ili više uzastopnih stepenovanja. U stringu koji predstavlja izraz se može pojaviti proizvoljan broj blanko znakova. Ukoliko učitani string ne sadrži dozvoljene znakove, korektno prekinuti program.

Program treba da:

- 1) Učita aritmetički izraz sa standardnog ulaza.
- 2) Izračuna vrednost aritmetičkog izraza.
- 3) Ispiše izračunatu vrednost na standardni izlaz.

Primeri

ULAZ	IZLAZ
1 + 2 exp 3 * 4 * 5	161
14+17*2+3exp2	57
9	9

1. Napisati program koji pronalazi najveći brojeve rečima zadatog stringa. Reči u stringu su međusobno odvojene zarezima. Svaka reč se sastoji od slova (velikih i malih) i cifara. Pronađene najveće brojeve zatim ukloniti iz reči. Ukoliko reč na sadrži nijednu cifru, ne menjati je. Ukoliko se reč sastoji samo od cifara, ukloniti celu reč. Ukoliko reč sadrži više najvećih brojeva, ukloniti sva pojavljivanja. Nove reči ispisati na standardnom izlazu odvojene razmakom. Ukoliko učitani string ne sadrži dozvoljene znakove, korektno prekinuti program.

Program treba da:

- 1) Učita zadati string.
- 2) Pronađe najveći broj u svakoj reči stringa.
- 3) Izbaci najveći broj iz svake reči na način definisan tekstem zadatka.
- 4) Ispiše nove reči odvojene razmakom.

Primeri

ULAZ	IZLAZ
abcd2ef3g0,q312wer67ty	abcd2efg0 qwer67ty
abc,8086,d89e89f89g	abc defg
I0001K	I000K

2. Napisati program koji računa prosečno vreme trkača u trkama na 800 metara na osnovu zadatog teksta. U tekstu su različite trke odvojene znakom |. Za svaku trku su navedena prolazna vremena svakog kruga, pri čemu su krugovi odvojeni znakom /. Vreme kruga je izraženo u sekundama. Ukupno vreme trke se dobija sabiranjem vremena svih krugova trke. Prosečno vreme trkača se formira na osnovu ukupnih vremena svih zadatih trka podeljenog sa brojem trka. Broj krugova u trci može biti različit za različite trke (u zavisnosti od dužine staze). Program treba da ispiše prosečno vreme trkača i broj trka prema formatu u primeru. Obratiti pažnju na promenu oblika reči „trka“ u zavisnosti od poslednje cifre u broju trka („trci“, „trke“, „trka“). Ukoliko učitani string ne sadrži dozvoljene znakove, korektno prekinuti program.

Program treba da:

- 1) Učita zadati tekst sa standardnog ulaza.
- 2) Izračuna prosečno vreme trkača.
- 3) Ispiše izračunato prosečno vreme i broj trka na standardni izlaz prema formatu u primeru. Vreme zaokružiti na dve decimale.

Primeri

ULAZ	IZLAZ
51/52 25/25/26/28	Prosecno vreme 103.50s na 2 trke.
50/56 49/53 55/54 50/50	Prosecno vreme 104.25s na 4 trke.
25/24/26/27	Prosecno vreme 102.00s na 1 trci.

3. Napisati program koji pronalazi sekvencu sa najviše uzastopnih malih slova u svakoj reči zadatog stringa. Reči su međusobno odvojene znakom -. Svaka reč se sastoji od slova (velikih i malih) i cifara. Pronađene sekvence (koji se sastoje od najviše uzastopnih malih slova) iz svake reči spojiti u novu reč. Spajanje izvršiti u istom redosledu u kojem su navedene reči. Ukoliko se u reči nalazi veći broj sekvenci sa najviše uzastopnih malih slova, odabrati poslednju sekvencu u reči. Novodobijenu reč ispisati na standardnom izlazu. Ukoliko učitani string ne sadrži dozvoljene znakove, korektno prekinuti program.

Program treba da:

- 1) Učita zadati string.
- 2) Pronađe sekvence sa najviše uzastopnih malih slova u svakoj reči.
- 3) Pronađene sekvence spoji u novu reč.
- 4) Ispiše novu reč.

Primeri

ULAZ	IZLAZ
abc1d2ef3g0-q312wer67ty	abcwer
abc-1-d89e89f89g	abcg
AbrAkadaBra	kada

4. Napisati program koji pronalazi sve podstringove zadatog stringa (koji predstavlja veliki broj) za koje važi da su deljivi sa 12. Broj je deljiv sa 12 ako je deljiv sa brojevima 3 i 4. Podstringove ispisati sortirano po indeksu (ako se u nekom redu ispisuje podstring od i -te do j -te pozicije, a u narednom redu podstring od k -te do l -te pozicije, treba da važi da je $i < k$ ili $i = k, j < l$). Ispis formatirati na način prikazan u primerima. Ukoliko učitani string ne sadrži cifre, korektno prekinuti program.

Program treba da:

- 1) Učita veliki broj.
- 2) Pronadje i ispiše sve podstringove koji zadovoljavaju opisani uslov. Podstringove ispisati na gore opisan način.

Primeri

ULAZ	IZLAZ
3144189	Broj 3144 na poziciji 0 Broj 144 na poziciji 1
13579111113511757759	
194212	Broj 94212 na poziciji 1 Broj 4212 na poziciji 2 Broj 12 na poziciji 4

5. Napisati program koji iz zadate rečenice uklanja sve reči koje počinju i završavaju se istim slovom, a na njihovo mesto dodaje istu reč bez prvog i poslednjeg slova. Prilikom provere zanemariti razliku između malih i velikih slova i ignorisati znake interpunkcije (.,!?:;:). Smatrati da se reči razdvajaju blanko znacima ili znacima interpunkcije. Zamenu raditi samo ukoliko reč sadrži barem tri slova. Obradu ponavljati sve dok u rečenici postoje reči koje zadovoljavaju zadati uslov za zamenu.

Program treba da:

- 1) Učita rečenicu sa standardnog ulaza.
- 2) Vršiti opisanu obradu.
- 3) Ispisati novu rečenicu na standardni izlaz.

Primeri

ULAZ	IZLAZ
Madam Anna sings.	d nn ing.
Emineme, ti si reper.	ine, ti si p.
Aa!!! ee! Uu! Oho?	Aa!!! ee! Uu! h?

6. Napisati program koji uprošćava zadati izraz uklanjanjem nepotrebnih zagrada. Izraz koji se unosi sadrži prirodne brojeve, osnovne računске operacije (+, -, *, /) i male zagrade ("(", ")", "{", "}") sa standardnim prioritetima. Smatrati da izraz ne sadrži ugnježdene zagrade. Ukoliko učitani string ne sadrži dozvoljene znakove, korektno prekinuti program.

Program treba da:

- 1) Učita izraz.
- 2) Ukloni suvišne zagrade.
- 3) Ispiše uprošćeni izraz.

Primeri

ULAZ	IZLAZ
(2+3)+3+2*2	2+3+3+2*2
(1*2)*(1+3)+2	1*2*(1+3)+2
(1+3)/4*2	(1+3)/4*2

7. Napisati program koji vrši sabiranje dva velika cela broja. Brojevi se u memoriji čuvaju u okviru promenljivih tipa stringa. Broj može da počne znakom -, znakom + ili bez znaka i nakon toga sadržati samo decimalne cifre. Broj ne sme početi vodećom nulom. Ukoliko string sadrži neke druge znake, prekinuti izvršavanje programa i ne ispisivati ništa. Nakon računanja, rezultat ispisati prema formatu u primeru.

Program treba da:

- 1) Učita dva velika broja.
- 2) Proveri da li uneti stringovi predstavljaju cele brojeve. Ukoliko uslov nije ispunjen prekinuti izvršavanje programa.
- 3) Ispiše zbir unetih brojeva.

Primeri

ULAZ	IZLAZ
11111111111111111111 22222222222222222222	Rezultat je 33333333333333333333
1234 012	
+85461 -25489	Rezultat je 59972