

**PRAKTIKUM IZ PROGRAMIRANJA 1**

- domaći zadatak broj 4 –

- Python grupa -

Sastaviti program na programskom jeziku Python, koji radi sa **tekstom (stringom)** po sledećem redosledu izvršavanja:

- učitava jednu ili više linija teksta sa kojim će se raditi,
- poziva odgovarajuću funkciju koja izvrši traženu obradu nad zadatim tekstom,
- ponavlja prethodne korake sve dok korisnik ne zada prazan tekst.

Zavisno od rednog broja problema, sastaviti program, koji obavlja sledeću obradu:

- čita tekstualnu liniju koja predstavlja aritmetički izraz koji se sastoji od pozitivnih celih brojeva u osnovi 2, 8 ili 16 i aritmetičkih operatora sabiranja (+) i oduzimanja (-). Binarni brojevi počinju prefiksom **b**, oktalni počinju nulom **0**, dok heksadecimalni brojevi počinju prefiksom **0x**. Potrebno je izračunati zadati izraz i rezultat ispisati u osnovi 10. Primer ulaza: **b1001+ 0xAf2 - b11**, izlaz je **2808**. Smatrati da između operanada i operacija može postojati proizvoljan broj belina.
- čita po dve tekstualne linije od kojih prva predstavlja spisak namirnica koje se kupuju i njihove pojedinačne količine u gramima (**g**) ili kilogramima (**kg**), dok druga linija predstavlja spisak namirnica i njihove cene po kilogramu. Pojedinačne namirnice i u jednoj i u drugoj liniji su odvojene zarezom. Potrebno je pronaći ukupnu cenu kupovine. Primer ulaza: prva linija: **so 500g, brašno 4kg, bademi 400g, brasno 2500g**, druga linija: **so 50, brasno 187, bademi 3811**, izlaz: **2764.9**. Smatrati da su količine i cene celi brojevi, kao i da su slova **š, č, ć, ž** i **đ** ekvivalentna slovima **s, c, c, z** i **dj** respektivno.
- čita tekstualnu liniju koja predstavlja logički izraz, gde se mogu pojaviti operatori za poređenje vrednosti (**>**, **<**, **==**), kao i logički operatori I (**&&**) ili ILI (**||**). Smatrati da su operandi pozitivni celi brojevi. Potrebno je odrediti logičku vrednost zadatog izraza. Operacije I i ILI imaju manji prioritet od operacija poređenja. Smatrati da su operacije I i ILI istog prioriteta i da primenjuju s leva na desno. Primer ulaza: **12>2 &&3 < 14||56==56**, izlaz: **true**. Smatrati da između operanada i operacija može postojati proizvoljan broj belina.
- čita tekstualnu liniju koja predstavlja logički izraz, gde se mogu pojaviti operatori za poređenje vrednosti (**>**, **<**, **<=**, **>=**, **==**), kao i logički operatori I (**&&**) ili ILI (**||**). Smatrati da su operandi imenovane promenljive čija se imena sastoje isključivo od slova engleskog alfabeta. Operatori za poređenje (sem **==**) u zadatom izrazu mogu da se lančaju zajedno, na primer **5<=a<10**. Potrebno je izmeniti zadati izraz tako da se operatori **&&** i **||** zamene operatorima **and** i **or**, kao i da se ulančana poređenja (**5<=a<10**) zamene posebnim poređenjima (**5<=a and a<10**). Primer ulaza: **a==b &&a<c<=b || aa<=d<=b**, izlaz: **a==b and a<c and c<=b or aa<=d and d<=b**. Smatrati da između operanada i operacija može postojati proizvoljan broj belina.
- čita tekstualnu liniju koja predstavlja izraz sa obeleženim formatiranjem. Format može da bude podebljan (**bold**), podvučen (**underline**) ili da deklariše veliko početno slovo (**upper**). Formatirani deo teksta počinje nazivom formata unutar oštih zagrada (**<bold>**), a završava se kosom crtom i nazivom formata unutar oštih zagrada (**</bold>**). Potrebno je zadatu tekstualnu liniju formatirati na sledeći način: **bold** format treba da označenom tekstu doda **\*\*\*** na početak i kraj, **underline** treba da doda **\_\_** na početak i na kraj, dok **upper** treba da svako početno slovo reči učini velikim. Primer ulaza: **<underline><bold>Ovo</bold></underline> je <underline>sve podvučeno</underline>**, dok je ostalo **<upper>napisano velikim početnim slovima</upper>**, izlaz: **\*\*\*Ovo\*\*\* je sve podvučeno, dok je ostalo Napisano Velikim Pocetnim Slovima.**

Smatrati da je format ispravno deklarisan i da se u slučaju ugneždavanja formati zatvaraju u obrnutom redosledu od otvaranja.

5. čita tekstualnu liniju, koja u sebi sadrži podatke o korisniku odvojene zarezom. Podaci su redom ime i prezime (prezime može da sadrži dve reči odvojene crtom), datum rođenja (`dd.mm.gggg.`) i broj telefona (bilo koji od sledećih formata: `bbb/bbb-bb-bb`, `bbb bbb bbbb` ili `+bbb bb bbb bb bb`). Potrebno je kreirati šifru korisnika tako što se uzimaju prva slova imena (veliko slovo) i prezimena (malo slovo; u slučaju dva prezimena, bira se samo drugo), a zatim se nadovežu godina rođenja i broj koji se dobija sabiranjem delova telefonskog broja korisnika koji su razdvojeni znakovima `/` i `-` ili jednim blanko znakom. Primer ulaza: `Sava Savic-Obradovic, 15.3.2000., 064/123-45-67`, izlaz: `so2000299`. Drugi primer ulaza: `Sava Savic-Obradovic, 15.3.2000., +381 64 123 45 67`, izlaz: `so2000680`.
6. čita dve tekstualne linije, koje predstavljaju šifrovanu poruku i ključ za dešifrovanje. Šifrovana poruka može da sadrži slova engleskog alfabeta i razmake (*space*). Ključ za dešifrovanje može da sadrži operacije zamene mesta karakterima u poruci obeležene indeksima (**S**wap), operaciju obrtanja zadatog dela poruke (**R**evert), pri čemu se zadaju početni (inkluzivno) i krajnji indeks opsega (ekskluzivno), kao i invertovanje vrednosti svakog pojavljivanja zadatog karaktera u poruci (**I**nvert), čime se dobija novi karakter čija je ASCII vrednost `ord('A') + ord('z') - ord(ch)`, ukoliko je karakter predstavljao veliko slovo, odnosno `ord('a') + ord('z') - ord(ch)`, ukoliko je karakter predstavljao malo slovo. Ukoliko karakter nije bio slovo, onda operacija **Invert** nema efekta. Smatrati da su podaci korektni i da se karakteri teksta indeksiraju od počevši od nule.

Primer ulaza:

```
Lvlaje akurlp ken  
R(7:18)S(3:10)I(1)I(L)
```

Izlaz:

```
Ovo je neka poruka
```

Program treba da čita tekstualne linije uz vođenje računa o tipu podataka koji se čita. Ako nešto u postavci zadatka nije dovoljno precizno definisano ili ako su neki od zahteva međusobno suprotstavljeni, usvojiti razumnu pretpostavku i rešiti zadatak korišćenjem te pretpostavke. Osmisliti i napisati na papiru nekoliko primera ulaznih podataka sa kojima će se program testirati, kao i očekivani izlaz za odabrane primere. Test primere otkucati u bilo kom tekst editoru i priložiti zajedno sa tekstem programa. **Funkcije ne smeju pristupati promenljivama glavnog programa direktno, već samo putem svojih argumenata.**

### Napomene:

1. Odbrana četvrtog domaćeg zadatka je u sredu, 18.12.2019. Tačni termini za sve studente su objavljeni na sajtu predmeta.
2. Formula za redni broj problema **i** koji treba rešavati je sledeća  
(R – redni broj indeksa, G – poslednje dve cifre godine upisa):  
$$i = (R + G) \bmod 7$$
3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
  - `dz4.py`, koja sadrži izvorni tekst programa na programskom jeziku Python;
  - `dz4_u1.txt`, ..., `dz4_uN.txt`, koje sadrže nekoliko odabranih test primera kojima će program biti testiran.

11.12.2019. godine

sa predmeta