

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 4 –

- Pascal grupa -

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

- učitava jednu ili više linija teksta sa kojim će se raditi,
- poziva odgovarajući potprogram koji 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 i 16 i aritmetičkih operatora sabiranja (+) i oduzimanja (-). Binarni brojevi počinju prefiksom **b**, 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 se između operanada i operacija nalazi tačno jedan razmak.
- 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, brasno 4kg, bademi 400g, brasno 2500g**, druga linija: **so 50, brasno 187, bademi 3811**, izlaz: **2764**. 9. Smatrati da su količine i cene celi brojevi.
- 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 postoji tačno jedan razmak i da je izraz ispravno napisan.
- 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 postoji tačno jedan razmak i da je izraz ispravno napisan.
- 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: **<bold>Ovo</bold> je <underline>sve podvuceno</underline>, dok je ostalo <upper>napisano velikim pocetnim slovima</upper>**, izlaz: *****Ovo*** je __sve podvuceno__, dok je ostalo Napisano Velikim Pocetnim Slovima**. Smatrati da je format ispravno napisan i da nema ugneždavanja formata.

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 (bbb/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 -. Primer ulaza: `Sava Savic-Obradovic, 15.3.2000., 064/123-45-67`, izlaz: `so2000299`.

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 $\text{ord}('A') + \text{ord}('Z') - \text{ord}(ch)$, ukoliko je karakter predstavljao veliko slovo, odnosno $\text{ord}('a') + \text{ord}('z') - \text{ord}(ch)$, ukoliko je karakter predstavljao malo slovo. Ukoliko karakter nije bio slovo, onda operacija **I**nvert nema efekta. Smatrati da su podaci korektni i da se karakteri teksta indeksiraju počevši od jedinice.

Primer ulaza:

```
Lvlaje akurlp ken
```

```
R(8:19)S(4:11)I(1)I(L)
```

Izlaz:

```
Ovo je neka poruka
```

Program treba da čita tekstualne linije uz vođenje računa o tipu podataka koji se čita. **Za sve podatke koje ima smisla grupisati, osmisliti sopstveni tip zapisa i koristiti promenljive tog tipa, tamo gde je to moguće.** 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 tekstom programa. **Potprogrami 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.PAS`, koja sadrži izvorni tekst programa na programskom jeziku Pascal;
 - `dz4_u1.txt`, ..., `dz4_uN.txt`, koje sadrže nekoliko odabranih test primera kojima će program biti testiran.

11.12.2019. godine

sa predmeta