

PRAKTIKUM IZ PROGRAMIRANJA 1**- domaći zadatak broj 4 -**

Sastaviti program na programskom jeziku Pascal, koji radi sa **tekst** datotekama po sledećem redosledu izvršavanja:

- a) poziva potprogram koji učitava imena datoteka sa kojima će raditi,
- b) poziva odgovarajući potprogram koji izvrši traženu obradu nad datotekama,
- c) ponavlja prethodne korake sve dok korisnik za ime datoteke ne zada prazan string.

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

0. čita ulaznu datoteku, koja predstavlja promet na kreditnim karticama, gde je svaki red po formatu **brojKartice ime prezime promena** (broj kartice – string od tačno 7 znakova, ime i prezime – stringovi do 30 znakova, promena – realan broj) i koja može u susednim redovima imati podatke o istoj osobi (isti su ime i prezime); program u izlaznu datoteku prepisuje ime i prezime osobe, sve njene brojeve kartica i ukupnu promenu po formatu **ime, prezime, brojKartice1, ..., brojKarticeN, ukupnaPromena** (za svaku osobu u zasebnom redu); prva četiri znaka svakog broja kartice u izlaznoj datoteci zameniti sa *;
1. čita ulaznu datoteku sa podacima o pristupima internet servisu, gde je svaki red po formatu **internetServis ipAdresa pocetak kraj količina** (internet servis – string do 20 znakova, ip adresa korisnika – string do 16 znakova, pocetak i kraj – vremena u formatu **hh:mm:ss**; količina – ostvaren saobraćaj izražen u kilobajtima kao ceo broj) i koja može u susednim redovima imati podatke o istom servisu i korisniku (isti su internet servis i ip adresa korisnika); program treba da napravi izlaznu datoteku čiji redovi sadrže informacije o prosečnoj brzini prenosa podataka korisnika internet servisa po formatu **ipAdresa internetServis prosek**;
2. čita ulaznu datoteku, koja sadrži podatke o turneji muškičkog sastava, gde je svaki red po formatu **država grad datum** (država i grad – stringovi do 30 znakova, datum je u obliku **gggg-mm-dd**); podaci su uređeni rastuće po datumu; sastav se ne vraća u već posećenu državu; program treba da napravi izlaznu datoteku po formatu **država, brojGradova, datumPočetka, datumKraja** (drugi podatak je broj gradova koji će biti posećen u datoj državi, datum početka i datum kraja su prvi i poslednji datum nastupa u datoj državi po formatu **dd.mm.gg**); u izlaznoj datoteci treba da se nalaze samo države u kojima će sastav nastupati u više od 2 grada, svaka država u zasebnom redu;
3. čita ulaznu datoteku, koja predstavlja adresar, gde je svaki red po formatu **idOsobe ime imeOca prezime eMailAdresa** (identifikacija osobe – ceo broj; ime, ime oca, prezime, e-mail adresa – string do 25 znakova) i koja može u susednim redovima imati podatke o istoj osobi (isti su identifikacija osobe, ime, ime oca i prezime); program u izlaznu datoteku treba da upiše podatke o osobama koje imaju više od jedne e-mail adrese po formatu **ime, srednjeSlovo, prezime, brojAdresa** (za svaku osobu u zasebnom redu);
4. čita ulaznu datoteku, koja predstavlja bazu podataka o knjigama iznajmljivanim iz biblioteke, gde je svaki red po formatu **idOsobe ime prezime idKnjige datumUzimanja datumVraćanja** (identifikacija osobe i identifikacija knjige – ceo broj; ime i prezime – string do 25 znakova; datum – string po formatu **dd-mm-gggg**) i koja može u susednim redovima imati podatke o istoj osobi (isti su identifikacija osobe, ime i prezime); za svaku osobu, program u izlaznu datoteku treba da upiše podatke o knjigama za koje je prekoračeno dozvoljeno vreme iznajmljivanja knjige od 15 dana; podatke upisati po formatu **ime, prezime, idKnjige1 - ... - idKnjigeN ukupanBrojDanaPrekoračenja**;

5. čita ulaznu datoteku, koja predstavlja red letenja, gde je svaki red po formatu `idLeta polazište destinacija hh:mm` (identifikacija leta – ceo broj, polazište i destinacija – stringovi do 25 znakova, dok je poslednji podatak o vremenu trajanja leta u satima i minutima) i koja može u susednim redovima imati podatke o povezanim letovima (isti su destinacija prvog i polazište drugog); program u izlaznu datoteku prepisuje podatke o letovima koji mogu da se povežu po formatu `polazište destinacija ukupnoTrajanje međutačka1 - ... - međutačkaN` (za svaki let u zasebnom redu);
6. čita ulaznu datoteku, koja predstavlja muzičku plejlistu, gde je svaki red po formatu `imeBenda - Naslov numere - trajanje` (ime benda i naslov numere - string do 25 znakova koji može sadržati razmake; trajanje numere u formatu `hh:mm:ss`, gde `hh` predstavlja sate, `mm` minute, a `ss` sekunde) i koja može u susednim redovima imati podatke o istom bendu (isto je ime benda); program u izlaznu datoteku treba da upiše podatke o ukupnom trajanju svih numera za svaki bend, naslov najduže numere i njeno trajanje po formatu `imeBenda : ukupnoTrajanje : naslovNajdužeNumere : trajanjeNajdužeNumere`;

Program treba da čita ulaznu datoteku 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. Podatke iz datoteka nije dozvoljeno smeštati u bilo kakvu zbirku podataka u memoriji (na primer, niz ili ulančanu listu).** Smatrati da ulazna datoteka ima 0 ili više redova. Radi lakšeg čitanja, smatrati da svi stringovi sadrže tačno jednu reč, osim gde je to eksplicitno drugačije naznačeno. 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.**

Važno: Po uspešnom rešavanju osnovnog zadatka, napraviti modifikaciju programa tako da program obrađuje samo one linije iz ulaznih datoteka čiji je redni broj u skupu vrednosti iz opsega [1,255], koje korisnik unosi sa tastature.

Napomene:

1. Odbrana četvrtog domaćeg zadatka je u sredu, 16.12.2015. i četvrtak, 17.12.2015. godine. 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.

09.12.2015. godine

sa predmeta