

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 telefonski imenik, gde je svaki red po formatu `idOsobe ime prezime brojTelefona` (identifikacija osobe – ceo broj; ime, prezime, broj telefona – string do 30 znakova) i koja može u susednim redovima imati podatke o istoj osobi (isti su identifikacija osobe, ime i prezime); program u izlaznu datoteku štampa ime i prezime osobe i sve njene brojeve telefona po formatu `ime, prezime, brojTelefona1, ..., brojTelefonaN` (za svaku osobu u zasebnom redu); poslednje tri cifre svakog broja telefona u izlaznoj datoteci zameniti sa `***`;
1. 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 štampa podatke o osobama koje imaju više od jedne e-mail adrese po formatu `ime, srednjeSlovo, prezime, brojAdresa` (za svaku osobu u zasebnom redu);
2. čita ulaznu datoteku sa podacima o dial-up pristupima, gde je svaki red po formatu `idKorisnika username hh:mm:ss` (identifikacija korisnika – ceo broj, korisničko ime – string do 20 znakova, dok je poslednji podatak o vremenu provedenom na vezi u satima, minutima i sekundama) i koja u može u susednim redovima imati podatke o istom korisniku (isti su identifikacija korisnika i korisničko ime); program treba da napravi izlaznu datoteku čiji su redovi formata kao kod ulazne datoteke; u izlaznoj datoteci se nalaze zbirni podaci o dužini trajanja konekcija za svakog korisnika pojedinačno (za svakog korisnika u zasebnom redu);
3. čita ulaznu datoteku, koja sadrži podatke o turneji muzičkog 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 uzlazno 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 kojoj su samo države u kojima će sastav nastupati u više od 2 grada, svaka država u zasebnom redu;
4. čita ulaznu datoteku, koja sadrži podatke o turneji muzičkog sastava, gde je svaki red po formatu `država grad brojGledalaca datum` (država i grad – stringovi do 20 znakova, broj gledalaca – ceo broj, datum je po formatu `dd/mm/gggg`); podaci su uređeni uzlazno po datumu; sastav se ne vraća u već posećenu državu; program treba da napravi izlaznu datoteku po formatu `država, brojGradova, datumPočetka, datumKraja, brojNastupa` (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`, poslednji podatak je broj nastupa u državi) i u kojoj se nalaze samo one države u kojima će svaki nastup sastava biti pred više od 15000 ljudi;

Program treba da čita ulaznu datoteku uz vođenje računa o tipu podataka koji se čita. **Za sve podatke koje ima smisla grupisati, osmisлити 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). Smatrati da ulazna datoteka ima 0 ili više redova. Radi lakšeg čitanja, smatrati da svi stringovi sadrže tačno jednu reč. 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.**

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 utorak, 22.12.2009. godine. Tačni termini za sve studente su objavljeni u okviru sistema WebLab.
2. Formula za redni broj problema i koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):

$$i = (R + G) \bmod 5$$

3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
 - **dz4.PAS**, koja sadrži izvorni tekst programa na programskom jeziku Pascal;
 - **dz4_mod.PAS**, koji sadrži modifikovani program koji radi sa skupovima;
 - **dz4_u1.txt, ..., dz4_uN.txt**, koje sadrže nekoliko odabranih test primera kojima će program biti testiran.

10.12.2009. godine

sa predmeta