

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 sa podacima o partijama jedne kartaške igre, gde je svaki red po formatu **idPartija idIgrač karta1 karta2 karta3 karta4** (identifikacija partije i identifikacija igrača – celi brojevi; karte – dvoslovna oznaka, prvi znak boja K, P, H ili T, drugi znak vrednost 2, 3, ... 9, X, A, J, Q ili K) i koja može u susednim redovima imati podatke o istoj partiji; program u izlaznu datoteku treba da upiše za svaku partiju da li je ispravna ili nije, po formatu **idPartija ispravna** (ispravnost – string „Ispavan“ ili „Neispravan“); partija se igra sa jednim špilom od 52 karte; partija je ispravna ako se svaka karta u partiji javlja najviše jedanput;
1. čita ulaznu datoteku, koja sadrži informacije o fudbalerima, gde je svaki red po formatu **idFudbaler ime prezime pozicija** (identifikacija fudbalera – ceo broj; ime, prezime – string do 25 znakova; pozicija – jednoslovna oznaka) i koja može u susednim redovima imati podatke o istom fudbaleru (isti su identifikacija fudbalera, ime i prezime); prvo slovo u poziciji označava da li se radi o odbrani (slovo D), o veznom redu (slovo M), ili o napadu (slovo A); program u izlaznu datoteku treba da upiše podatke o svim fudbalerima koji mogu da igraju na više od jedne pozicije u timu, po formatu **ime, prezime, pozicije** (treći podatak je niz svih pozicija na kojima fudbaler može da igra, odvojeni znakom razmaka), za svakog fudbalera u zasebnom redu;
2. čita ulaznu datoteku, koja sadrži informacije o internet člancima objavljenim na raznim portalima u toku jednog dana, gde je svaki red po formatu **naslov portal hh:mm:ss brojKomentara** (naslov – string do 50 znakova koji se nalazi između znakova navoda, portal – string od 20 znakova; trenutak objave u formatu **hh:mm:ss** - gde **hh** predstavlja sate, **mm** minute, a **ss** sekunde, broj komentara – ceo broj); podaci su uređeni rastuće po leksikografskom poretku naslova; za svaki članak sa istim naslovom koji postoji u ulaznoj datoteci, program treba da upiše u izlaznu datoteku internet portale na kojima je članak dobio najmanje i najviše komentara, po formatu **naslov, portalMax, portalMin**;
3. čita ulaznu datoteku, koja sadrži podatke o sušnim periodima po godini, gde je svaki red po formatu **godina mesto količina datumPočetka datumKraja** (godina – ceo broj; mesto – string do 20 znakova; količina kiše u datom periodu – realan broj; datum početka i kraja perioda je po formatu **dd-mm-gggg**); informacije o jednom mestu nalaze se u susednim redovima; samo jedan sušni period postoji za jednu godinu; program treba da napravi izlaznu datoteku sa sušnim godinama za svako mesto po formatu **mesto, brojGodina, nizGodina**; drugi podatak je broj godina za vreme kojih je u sušnom periodu palo manje od 10ml kiše prosečno po danu; treći podatak su sve godine koje su bile sušne odvojene znakovima razmaka;

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;
5. čita ulaznu datoteku koja sadrži podatke o knjigama u biblioteci u formatu `idKnjige, nazivKnjige, imePisca, prezimePisca, brojPrimeraka` (identifikator knjige – ceo broj, naziv knjige – string od najviše 100 znakova koji se nalazi između znakova navoda, ime i prezime pisca – stringovi do 30 znakova, broj primeraka knjige – ceo broj); ukoliko ih ima više, dela istog pisca će biti u susednim redovima; treba formirati izlaznu datoteku koja će imati format `PrezimePisca, ImePisca, brojKnjiga`, gde je poslednja stavka ukupan broj primeraka knjiga tog pisca koje biblioteka ima;
6. čita ulaznu datoteku koja predstavlja evidenciju o angažovanju demonstratora i koji ima sledeći format: `šifraPredmeta, ime, prezime, početakTermina, krajTermina` (ime, prezime, šifra predmeta – stringovi do 20 znakova, početak i kraj termina su u formatu `hh:mm` - gde `hh` predstavlja sate, a `mm` minute); različita angažovanja istog demonstratora će biti u susednim redovima; potrebno je napraviti datoteku po formatu `ime, prezime, zarada`, gde je zarada ukupna količina novca koju treba isplatiti demonstratoru; cenu radnog sata korisnik unosi sa glavnog ulaza

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, 28.12.2016. i četvrtak, 29.12.2016. 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.