

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 4 -

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

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

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

- Čita ulaznu datoteku, koja predstavlja listu telefonskih poziva, gde svaki red po formatu **idPoziva brojTelefona vremePocetka vremeKraja** (identifikacija poziva – ceo broj; broj telefona – string do 30 znakova, a vreme početka i kraja poziva su u formatu **hh:mm:ss**, gde **hh** predstavlja sate, **mm** minute, a **ss** sekunde) i koja može u susednim redovima imati podatke o istom broju telefona; program u izlaznu datoteku treba da upiše broj telefona, ukupno trajanje svih poziva i ukupan račun po formatu **brojTelefona, hh:mm:ss, ukupanRacun DIN** (ukupan račun – realan broj); cena poziva od 7:00 do 21:00 iznosi 8 din/min, a cena poziva od 21:00 do 7:00 iznosi 5 din/min; smatrati da se započeti poziv uvek plaća prema tarifi po kojoj je započeo;
- 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 treba da upiše 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 *******;
- 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);
- 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, datumPocetka, 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;
- Čita ulaznu datoteku, koja sadrži vremensku prognozu za gradove u evropskim državama, gde je svaki red po formatu **država grad minTemp maxTemp** (država i grad – stringovi do 20 znakova, minimalna i maksimalna prognozirana temperatura – realan broj); podaci su uređeni rastuće po leksikografskom poretku imena država; za svaku državu koja postoji u ulaznoj datoteci, program treba da upiše u izlaznu datoteku grad sa minimalnom prognoziranom temperaturom i grad sa maksimalnom prognoziranom temperaturom, po formatu **država, grad : minTemp, grad : maxTemp**;

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 listu). 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.

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:

Odbrana četvrtog domaćeg zadatka je u ponedeljak 17.12.2012. i utorak 18.12.2012. godine. Tačni termini za sve studente će biti blagovremeno objavljeni na sajtu predmeta.

1. Formula za redni broj problema i koji treba rešavati je sledeća

(R – redni broj indeksa, G – poslednje dve cifre godine upisa):

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

2. 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.

10.12.2012. godine

sa predmeta