

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 sa podacima o datotekama, gde je svaki red po formatu **vlasnik gggg-mm-dd veličina ime** (vlasnik datoteke – jedna reč do 30 znakova; datum poslednje izmene datoteke po formatu **dd-mm-gggg**; veličina datoteke u bajtovima – ceo broj; ime datoteke – string do 255 karaktera) i koja može u susednim redovima imati podatke o datotekama istog vlasnika; program u izlaznu datoteku treba da upiše zbirne podatke za svakog vlasnika po sledećem formatu: **vlasnik brojDatoteka ukupnaVelicina**; ukupnu veličinu svih datoteka treba izraziti u kilobajtima (i dodati sufiks **K**) ili megabajtima (i dodati sufiks **M**) u zavisnosti da li ukupna veličina svih datoteka prevazilazi 2^{10} ili 2^{20} bajtova;
- čita ulaznu datoteku sa podacima o korisnicima i programima koje su oni pokrenuli na nekom operativnom sistemu, gde je svaki red po formatu **korisnik dd-mm-gggg hh:mm:ss imePrograma** (korisnik - jedna reč do 30 znakova; datum pokretanja programa po formatu **dd-mm-gggg**; vreme pokretanja programa po formatu **hh:mm**, gde **hh** predstavlja sate, a **mm** minute; ime programa – string do 255 karaktera) i koja može u susednim redovima imati podatke o pokrenutim programima istog korisnika; program u izlaznu datoteku treba da upiše za svakog korisnika podatke o pokrenutom programu koji se najduže izvršavao do trenutka obrade po sledećem formatu: **korisnik imePrograma ukupnoVremeIzvršavanja**;
- čita ulaznu datoteku sa podacima o internet sajtovima i njihovim pristupima u okviru 24 časa, gde je svaki red po formatu **internetSajt ipAdresa vremePristupa** (internet domen – jedna reč do 20 znakova; ip adresa sajta – string do 16 znakova, vremena pristupa u formatu **hh:mm:ss**, gde **hh** predstavlja sate, **mm** minute, a **ss** sekunde) i koja može u susednim redovima imati podatke o istom sajtu; program treba da napravi izlaznu datoteku čiji redovi sadrže podatke o broju pristupa sajtu u zadatom vremenskom periodu po sledećem formatu: **internetSajt ipAdresa brojPristupa**; vremenski period korisnik zadaje preko glavnog ulaza;
- čita ulaznu datoteku, koja sadrži informacije o vestima objavljenim od strane različitih novinskih agencija u toku jednog dana, gde je svaki red po formatu **naslov; agencija; hh:mm:ss** (naslov – string do 50 znakova; agencija – jedna reč do 20 znakova; trenutak objave u formatu **hh:mm:ss** - gde **hh** predstavlja sate, **mm** minute, a **ss** sekunde); 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 novinsku agenciju koja je prva objavila članak i novinsku agenciju koja je poslednja objavila članak, po formatu **naslov, portalPrvi, portalPoslednji**; za naslov članka treba predvideti tačno 20 znakova; ukoliko je naslov kraći, dopuniti ga blanko znacima sa leve strane, a ukoliko je duži skratiti ga sa desne strane na tačno 20 znakova;

4. čita ulaznu datoteku, koja sadrži vremensku prognozu za gradove u svetu, gde je svaki red po formatu `država grad minTemp maxTemp nadmorskaVisina` (država i grad – stringovi do 20 znakova; minimalna i maksimalna prognozirana temperatura – realan broj; nadmorska visina – ceo 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`; u izlaznu datoteku prepisati samo gradove koji se nalaze na nadmorskoj visini koja je niža od one koja se zadaje putem glavnog ulaza;
5. čita ulaznu datoteku, koja sadrži podatke o salama za polaganje ispita, njihovim kapacitetima i raspoloživosti, gde je svaki red po formatu `sala kapacitet vremePocetka vremeKraja` (sala – string do 20 znakova; kapacitet – ceo broj; vreme početka i kraja raspoloživosti u formatu `hh:mm`, gde `hh` predstavlja sate, `mm` minute) i koja može u susednim redovima imati podatke o istoj sali; program treba da napravi izlaznu datoteku čiji redovi sadrže informacije o salama po istom formatu kao što je ulazni, a koje su raspoložive u vremenskom periodu koji korisnik zadaje preko glavnog ulaza; na kraju datoteke ispisati ukupan kapacitet svih takvih sala u posebnom redu;
6. čita ulaznu datoteku, koja sadrži podatke o ispitima u jednom ispitnom roku, gde je svaki red po formatu `datum vreme sifraIspita brojStudenata` (šifra ispita – string do 10 znakova; datum – string po formatu `dd/mm/gggg`; vreme – string po formatu `hh:mm`; brojStudenata – ceo broj) i koja može u susednim redovima imati podatke o ispitima koji se održavaju istog datuma; podaci su uređeni rastuće po datumu; program treba da napravi izlaznu datoteku čiji svaki red sadrži informacije o najmasovnijem ispitu koji se održava tog datuma po formatu `datum vreme sifraIspita brojStudenata`;

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. 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, 20.12.2017. i četvrtak, 21.12.2017. 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.