

## PRAKTIKUM IZ PROGRAMIRANJA 2

### - domaći zadatak broj 4 -

Sastaviti program na programskom jeziku C koji vrši određenu vrstu obrade nad jednom ili više jednostruko ulančanih lista. Program treba da omogući učitavanje svih potrebnih ulaznih podataka, njihovo ispisivanje, obradu, ispis dobijenih rezultata, i ponavljanje navedenih koraka sve dok korisnik ne unese vrednost koja označava kraj programa. **Navedene korake u izvršavanju programa realizovati kao zasebne funkcije prema dole navedom rasporedu.**

Zavisno od rednog broja problema, sastaviti **jedan** od sledećih programa, koji:

0. obrađuje ulančanu listu video klipova sa Youtube servisa i formira odgovarajuće plejliste; podaci o video klipu se sastoje od naslova, kratkog opisa, trajanja, kategorije i niza tagova (jednosložne reči koje opisuju klip); za potrebe zadatka smatrati da je broj tagova ograničen; korisnik sa standardnog ulaza unosi maksimalan broj klipova koji želi u plejlisti, maksimalno trajanje pojedinačnog klipa i određeni broj tagova; potrebno je formirati posebnu plejlistu za svaku kategoriju klipova koja se pojavljuje u ulaznoj ulančanoj listi; svaku plejlistu treba popunjavati svim onim klipovima odgovarajuće kategorije i trajanja koji su obeleženi barem jednim od tagova koje je uneo korisnik; plejlistu popunjavati do maksimalnog broja klipova definisanog od strane korisnika;
1. raspoređuje posetioce koncerta po kategorijama: fan pit, parter 1, parter 2, tribine, vip loža; za svaku kategoriju se unosi broj mesta i cena ulaznice; podaci o posetiocima koncerta su smešteni u okviru ulančane liste i se sastoje od imena i prezimena i dve željene kategorije po prioritetima; program treba da rasporedi posetioce po kategorijama i dodeli im karte prema željama; ukoliko nije moguće dodeliti kartu prema željama, dodeliti kartu iz neke od preostalih kategorija koja nije skuplja od prve želje, ali tek nakon što se svim posetiocima dodeli karta po prvoj ili drugoj želji; u suprotnom dodeliti oznaku da posetilac nije dobio kartu; na kraju izračunati ukupnu sumu od ulaznica koju je zaradio organizator koncerta;
2. raspoređuje igrače u fudbalske timove od 11 igrača; podaci o igračima se sastoje od imena, prezimena, pozicije (D – odbrana, M – sredina, F – napad) i ocene (realan broj od 0 do 10); korisnik unosi željenu formaciju tima (broj igrača u odbrani, sredini i napadu), a program treba da od raspoloživih igrača formira što je moguće veći broj timova; tim se uvek formira u zadatoj formaciji od najboljih raspoloživih igrača i tako redom; rezultat raspoređivanja smestiti u ulančanu listu koja sadrži timove sa podacima o igračima koji se nalaze u svakom od njih;
3. vrši obradu ulančane liste koja sadrži vremensku prognozu za određeni period u godini; podaci o vremenskoj prognozi za određeni dan su datum, prognozirana minimalna i maksimalna temperatura i vremenske prilike (sunčano, oblačno, kišovito, sa snežnim padavinama); podaci u listi su sortirani po datumu; korisnik sa standardnog ulaza unosi vremenski period od interesa, dozvoljene vremenske prilike i temperaturni opseg i broj uzastopnih dana sa takvim vremenskim prilikama; program treba da iz liste izdvoji sve disjunktne periode uzastopnih dana koji zadovoljavaju kriterijume koje je uneo korisnik; rezultat treba smestiti u zasebne ulančane liste za svaki period vremena koji odgovara kriterijumima;
4. formira biračka mesta u nekoj opštini; opština se sastoji od više naseljenih mesta, a svako naseljeno mesto treba da ima najmanje jedno zasebno biračko mesto; jedno biračko mesto može da primi najviše N glasača; podaci o glasačima se sastoje od imena, prezimena, JMBG-a i mesta stanovanja; korisnik unosi maksimalan kapacitet pojedinačnog biračkog mesta, a program treba da rasporedi glasače po biračkim mestima u leksikografskom poretku vodeći računa o naseljenom mestu u kojem žive; svakom biračkom mestu dodeliti naziv koji se sastoji od imena naseljenog mesta i rednog broja biračkog mesta (npr. „Beograd 1“, „Beograd 2“ i sl.); rezultat raspoređivanja smestiti u ulančanu listu koja sadrži podatke o glasačima i dodeljnim biračkim mestima;

5. simulira upravljanje memorijom od strane operativnog sistema; jedan element ulančane liste sadrži adresu početka bloka (ceo broj), veličinu memorije koju blok zauzima u bajtovima i polje koje označava da li je blok slobodan ili ne; memorijski blokovi su uređeni po početnoj adresi bloka; korisnik repetitivno unosi zahteve za alokacijom ili dealokacijom memorije unošenjem veličine potrebnog bloka prilikom alokacije ili unošenjem adrese bloka koji treba osloboditi prilikom dealokacije; prilikom alokacije memorije, korisniku vratiti adresu prvog slobodnog bloka, a ukoliko je slobodni blok veći od potrebne veličine memorije, podeliti ga na dva nova bloka (zauzeti i preostali slobodni); prilikom dealokacije memorije, program treba da spoji dva susedna memorijska bloka ukoliko su slobodna i razlika početnih adresa im je jednaka veličini prvog bloka;

Sve funkcije smestiti u odgovarajuće **.c** datoteke (prema donjem spisku), a prototipove svih funkcija smestiti u zajedničku **.h** datoteku. Funkcijama koje vrše učitavanje i obradu treba dostaviti samo neophodne podatke (pokazivač na početak liste i, po potrebi, podatke od kojih zavisi obrada). Potrebno je napisati funkcije koje vrše ispisivanje liste, brisanje liste, ubacivanje elementa na početak i na kraj liste i izbacivanje proizvoljnog elementa iz liste.

Napraviti interaktivni meni kojim se omogućava učitavanje liste, brisanje liste, izbacivanje elementa iz liste, ubacivanje elemenata na početak i kraj liste, ispisivanje liste, obrada liste i prekidanje programa. **Ukoliko program radi sa više listi obezbediti posebne stavke menija (koje su potrebne) za svaku listu. Ukoliko se obrada sastoji iz više koraka obezbediti posebne stavke menija za svaki korak.** Voditi računa o pravilnom alociranju i dealociranju dinamičke memorije. **Potprogrami ne smeju komunicirati pomoću globalnih promenljivih, već samo preko liste argumenata i povratne vrednosti.**

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. Odabrati nekoliko skupova podataka sa kojima će program biti testiran. Odabrane test primere priložiti na listu papira pre odbrane. Kandidati koji na odbrani nemaju spremna makar tri suštinski različita test primera ne mogu dobiti maksimalan broj poena. Programski kod rešenja zadatka treba da bude uredno komentaran, tako da pri pregledu programa lako može biti uočeno šta radi bilo koja programska celina. Takođe, treba poštovati pravila nazubljanja (identacije) određenih celina prilikom pisanja koda.

### **Napomene:**

1. Odbrana četvrtog domaćeg zadatka je u četvrtak, 24.05.2018. po rasporedu dostupnom 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):  **$i = (R + G) \bmod 6$**
3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
  - **dz4.c**, koja sadrži izvorni tekst osnovnog, glavnog programa na programskom jeziku C;
  - **dz4.h**, koja sadrži prototipove svih funkcija opisanih u postavci zadatka;
  - **dz4\_unos.c**, **dz4\_ispis.c**, **dz4\_obrada.c**, koje sadrže izvorne tekstove potrebnih funkcija.

15.05.2018. godine

*sa predmeta*