

## 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 navedenom rasporedu.**

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

0. raspoređuje novinske članke iz ulančane liste po stranicama novina; članak se sastoji od imena i procenta prostora na stranici koji zauzima; svaka stranica može sadržati maksimalno N članka; program treba da pročita broj N sa standardnog ulaza i formira novinske stranice tako da se na stranicu uvek dodaje najveći članak koji može da stane, sve dok ima članaka u listi; novu stranicu formirati kada postojeća ne može da se dopuni preostalim člancima iz liste; rezultat obrade smestiti u ulančanu listu koja sadrži stranice sa imenima članaka koji se nalaze na svakoj od njih;
1. obrađuje ulančanu listu koja sadrži zakazane sastanke u toku jednog dana; program treba da ispiše sve termine sastanaka koji se preklapaju i da formira i ispiše ulančanu listu sastanaka na koji korisnik treba da ode u okviru zadatog dana; sastanak je određen mestom sastanka, celobrojnim prioritetom (0-10) i vremenom početka i kraja; ako se dva ili više sastanaka preklapaju, prednost dati onome sa većim prioritetom, a u slučaju jednakih prioriteta, prioritet dati onom koji počinje najranije;
2. simulira rad agencije za prodaju karata za neki događaj; agencija sadrži spisak ljudi u vidu ulančane liste koji žele da kupe kartu; podatak o jednom čoveku su ime, e-mail adresa, cena koju je spreman da plati i oznaku da li se radi o poznatoj ličnosti; agencija prodaje karte u koracima; u svakom koraku sa standardnog ulaza se unese broj karata i cena pojedinačne karte koje će biti raspodeljene i to na sledeći način: karte se prvo dodeljuju poznatim ličnostima koji imaju dovoljno novca da plate kartu, a nakon toga svim ostalima koji imaju dovoljno novca da plate kartu; u svakom koraku ispisati ko je dobio karte; raspodela karata se završava kada svi zainteresovani dobiju karte ili kada agencija nema više karata za prodaju;
3. 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; rezultat raspoređivanja smestiti u ulančanu listu koja sadrži timove sa podacima o igračima koji se nalaze u svakom od njih;
4. raspoređuje glasače po biračkim mestima u nekoj opštini; opština se sastoji od više naseljenih mesta, a svako naseljeno mesto ima najmanje jedno zasebno biračko mesto; 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; program treba da spoji sve susedne, slobodne memorijske blokove predstavljene putem ulančane liste u jedan veći memorijski blok; jedan element 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; smatrati da je razlika adresa dva uzastopna bloka jednaka veličini prvog bloka u bajtovima; ulančanu listu slobodnih memorijskih blokova održavati sortiranom po adresi početka bloka;

Sve funkcije smestiti u odgovarajuće **.c** datoteke (prema donjem spisku), a prototipove svih funkcija smestiti u zajedničku **.h** datoteku. Učitavanje liste/lista realizovati pomoću funkcije kojoj će kao argument biti dostavljena adresa pokazivača na početak liste i broj elemenata koji treba učitati, a koja preko povratne vrednosti vraća podatak o uspešnosti učitavanja. Funkcijama koje vrše 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 (varijante 0, 1, 3, 4) obezbediti posebne stavke menija (koje su potrebne) za svaku listu.** 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 komentarisano, tako da pri pregledu programa lako može biti uočeno šta radi bilo koja programska celina. Takođe, treba poštovati pravila nazublivanja (identacije) određenih celina prilikom pisanja koda.

#### **Napomene:**

1. Odbrana četvrtog domaćeg zadatka je u sredu, 20.05.2015., četvrtak, 21.05.2015. i petak, 22.05.2015. 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 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.

12.05.2015. godine

*sa predmeta*