

## 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.

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

0. iz ulančane liste koja sadrži vremenske termine zauzetosti sala za kolokvijume izbacuje sve one termine koji se preklapaju; jedan element liste sadrži broj sale i početak i kraj termina koji se zadaju u satima i minutima; izbačene termine smestiti u novu listu;
1. spaja susedne, slobodne memorijske blokove predstavljene putem ulančane liste; 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; razlika adresa dva uzastopna bloka je jednaka veličini prvog bloka u bajtovima;
2. iz liste čiji su elementi celi brojevi, vrši izbacivanje prvih  $n$  elemenata koji imaju najveće zbrove cifara od svih elemenata u listi; nenegativan, ceo broj  $n$  se unosi putem standardnog ulaza;
3. simulira razmenu sličica između dve osobe; svaka osoba poseduje dva spiska sličica; jedan je spisak sličica koje osoba poseduje za razmenu (duplikati), a drugi je spisak sličica koje joj nedostaju; pretpostaviti da jedna osoba poseduje samo jedan duplikat; spisak sličica se predstavlja u vidu ulančane liste koja sadrži redni broj sličice u albumu i ime i prezime igrača koji se nalazi na sličici; na osnovu zadatih spiskova svake osobe, potrebno je formirati spisak sličica koje nedostaju prvoj osobi, a poseduje druga i obrnuto; na kraju, omogućiti razmenu  $n$  sličica, pri čemu se  $n$  unosi sa standardnog ulaza;
4. simulira igru „aračkinje baračkinje“; u igri učestvuju dva tima igrača poređanih u dve vrste (ulančane liste) pri čemu se igrači jednog tima drže za ruke; za svakog igrača se unosi ime i snaga u napadu i odbrani (realan broj); jedan potez igre se sastoji iz sledećeg: igrač tima koji napada se zaleće iz svoje vrste i trči ka suparničkoj pokušavajući da probije lanac; ukoliko je jačina njegove snage napada veća od zbira jačina snage odbrane igrača koje on hoće da rastavi, on uspeva da ih rastavi i može da izabere igrača suparničkog tima koga će kao pobednik tog poteza dovesti nazad u svoj tim; ukoliko mu je napad slabiji od odbrane on ostaje u suparničkom timu na mestu gde je hteo da probije lanac; za svaki napad se putem standardnog ulaza unosi redni broj igrača koji napada i redni broj levog igrača od para igrača koje će pokušati da rastavi, kao i redni broj igrača koga bi doveo u svoj tim ako bi bio jači; potrebno je ažurirati i prikazati stanje lista nakon svakog poteza; nakon prvog tima na potezu je uvek drugi; igra se završava kada u nekom timu ostane jedan igrač;
5. predstavlja simulaciju rešavanja „[Josephus problem](#)“-a; određeni broj osoba (predstavljenih svojim imenima i prezimenima u vidu ulančane liste) je raspoređeno u krug i svaki korak rešavanja problema se sastoji od izbacivanja jedne osobe iz kruga, a zatim preskakanja  $n$  narednih osoba; pozicija u krugu osobe koja se izbacuje na samom početku i broj  $n$  se unose sa standardnog ulaza; potrebno je prikazati stanje liste osoba nakon svakog kruga sve dok u listi ne preostane jedna osoba;

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, 3 i 4) obezbediti posebne stavke menija za svaku listu. Ukoliko se obrada sastoji iz više koraka (varijante 3 i 4) obezbediti posebne stavke menija za svaki korak. Voditi računa o pravilnom alociraju 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.

**Važno: po uspešnom rešavanju osnovnog zadatka, potrebno je omogućiti ispisivanje bitova binarne predstave podatka tipa float. U zavisnosti od dodeljenog problema, treba omogućiti ispis bitova proizvoljnog realnog broja koji već postoji u programu (npr. rezultat neke obrade ili neki od unetih podataka) ili učitati broj sa standardnog ulaza. Opciju za ispis bitova binarne predstave podatka tipa float dodati u meni kao jednu od mogućnosti. Za funkciju koja ostvaruje opisanu obradu dodati posebnu .h i posebnu .c datoteku.**

#### Napomene:

1. Odbrana četvrtog domaćeg zadatka je u laboratoriji 60, u četvrtak, 27.05. i sredi 28.05.2010., prema rasporedu koji je dostupan putem sistema WebLab.
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;
  - dz4.vcproj, koja sadrži informacije o projektu koji se odnosi na osnovni program;
  - dz4\_dodatak.h, koja sadrži prototip funkcije za ispis bitova realnog broja,
  - dz4\_dodatak.c, koja sadrži telo funkcije za ispis bitova realnog broja,
  - dz4\_dodatak.vcproj, koja sadrži informacije o projektu koji se odnosi na dodatni program;
  - dz4.sln, koja sadrži informacije o svim projektima relevantnim za ovaj zadatak.

18.05.2010. godine

sa predmeta