

Uvod u programiranje

Ispit – grupa 1

1. (13p) Profesor Marko je studentu Mihajlu zadao zadatak da napiše program koji proverava da li je zadati kvadrat magičan. Međutim, Mihajlo "nema mnogo vremena" da se bavi time, pa je odlučio da zadatak "prepiše" od studenata sa predmeta Uvod u programiranje. Program koji je Mihajlu potreban treba da učita sa tastature broj N i matricu M dimenzija NxN, te da proveri sledeće uslove:

1. u svakom polju matrice je broj između 1 i N^2 (uključujući i brojeve 1 i N^2)
2. ne postoji dva ili više polja matrice u kojima se nalazi isti broj
3. zbroovi brojeva u svim redovima, kolonama i na svim dijagonalama matrice su jednaki

Program treba da ispiše poruku "da" ili "ne" u zavisnosti od rezultata provere navedenih uslova.

Napomena: Nije dozvoljeno korišćenje pomoćnih nizova i matrica.

| Ulaz: | Ulaz: | Ulaz: | Ulaz: |
|---------------|-------------------|--------------------|----------------|
| 3 | 3 | 3 | 3 |
| 8 1 6 | 8 10 6 | 8 1 6 | 8 1 7 |
| 3 5 7 | 3 5 7 | 3 1 7 | 3 5 6 |
| 4 9 2 | 4 9 2 | 4 9 2 | 4 9 2 |
| Izlaz: da | Izlaz: ne | Izlaz: ne | Izlaz: ne |
| Zbroovi su 15 | 10 je izvan ranga | 1 se javlja 2 puta | Nije isti zbir |

Dodatno pitanje za razbijanje tenzije na ispitu: Koju ocenu je profesor Marko upisao Mihajlu? (prihvataju se samo netačni odgovori)

2. (12p) Profesor Dalibor je asistentu Mihajlu zadao zadatak da studentima na ispitu iz predmeta Uvod u programiranje ispiše zanimljive motivacione poruke (kad već nije htelo da smišlja zadatke). On je odlučio da motivacione poruke budu u binarnoj formi, te da studenti mogu da ih pročitaju samo ako napišu program koji učitava binarne stringove (stringove sastavljene samo od karaktera 0 i 1) i pretvara ih u čitljiv tekst. Pretvaranje binarnog stringa u čitljiv tekst se radi na sledeći način:

1. u svakoj iteraciji se uzima po 8 karaktera iz stringa
2. traži se broj u dekadnom sistemu koji odgovara formiranim binarnom broju
3. broj u dekadnom sistemu se pretvara u odgovarajući karakter po ASCII kodu

Postoji mogućnost da korisnik unese nevalidan binarni string. U tim slučajevima ispisati odgovarajuću poruku o grešci umesto pretvaranja u čitljiv tekst. Moguće greške su:

- dužina stringa nije deljiva sa 8, poruka o grešci treba da sadrži podatak o tome kolika je dužina unetog stringa
- prilikom pronalaženja karaktera sa datim ASCII kodom je pronađen nevalidan karakter (validni karakteri su mala i velika slova, cifre i razmak – space), poruka o grešci treba da sadrži ASCII kod nevalidnog karaktera

Test primeri (korisnički unos je **bold**, ispis iz programa je *italic*):

0100100101110011011100000110100101110100001000000101010101010

10101010000

Ispit UUP

01010011011100100110010101100011011011100110111101101111

Srecnootoo

00110001001100100011001100100000011000010110001001100011

123 abc

0011000100110010001100110010000001100001011000100110001100

Greska: Broj karaktera u unetoj liniji 53 nije deljiv sa 8

0100001001101111011110100110010100100000011011010111011101101

11101101000100001

Greska: Karakter sa ASCII kodom 33 nije validan

Napomena: Nije dozvoljeno korišćenje dodatnih nizova (uključujući stringove), kao i strtok funkcije u ovom zadatku.

3. (25p) Deklarisana je struktura Element na sledeći način:

```
typedef struct Element {  
    char grad[30]; double peron; struct Element *levo, *desno;  
} Element;
```

U tekstualnoj datoteci "ulaz.txt" nalazi se lista gradova iz kojih je moguće putovati do Hogvorts-a, zajedno sa peronima sa kojih polazi voz, razdvojeni zarezom. Damblor je izabrao baš vas da napišete program koji na osnovu tog fajla radi određene funkcionalnosti.

U fajlu zadatka (koji se kompajlira ali ne radi ništa) je dat šablon za funkcije koje treba implementirati (ne menjati im potpis!) i glavni program koji ne treba menjati. Implementirati sledeće funkcije (dozvoljeno je dodavati i pomoćne funkcije):

a) (11p) **Element* ucitajFajl()**. Učitava podatke iz ulaznog fajla u binarno pretraživačko stablo. Podaci se u stablo prilikom dodavanja sortiraju po gradu rastuće, a ako ima više čvorova stabla vezanih za isti grad, potrebno je sortirati ih opadajuće po peronu. Peroni su u fajlu dati kao razlomci, npr. 8 3/4 (osam i tri četvrtine), njih treba u stablo upisati kao realne brojeve, npr. 8.75.

b) (2p) **void ispisiDeoStabla(Element* stablo, char pocetnoSlovo)**. Ispisati podatke iz stabla sortirane po gradu rastuće i peronu opadajuće. U ispis uključiti samo one čvorove takve da imena gradova u njima počinju na zadato pocetnoSlovo.

c) (3p) **double naciNajmanjiPeronZaGrad(Element* stablo, char* grad)**. Ispisati najniži broj perona za zadati grad koristeći podatke iz stabla.

d) **(8p) double promeniPeron(Element* stablo, char* grad, double stariPeron, double noviPeron).** Promeniti peron sa kog polazi voz iz zadatog grada koristeći podatke iz stabla. Treba pronaći stariPeron za taj grad i zameniti ga vrednošću noviPeron. Ukoliko je stariPeron pronađen, vratiti ga kao povratnu vrednost funkcije. Ukoliko ne postoji za zadati grad, vratiti 0.

e) **(1p) void brisanje(Element** stablo).** Obrisati podatke iz stabla i obrisati svu zauzetu memoriju. Koren stabla postaviti na NULL.

Primer ispisa za treći zadatak:

Ispisujemo sve gradove na slovo 'L' i njihove perone:

Lyon 8.166667
Lyon 6.500000
Lyon 4.125000
Luxembourg 4.666667
London 7.400000
London 4.200000
Lodz 3.250000
Lodz 2.250000
Ljubljana 4.571429
Lisbon 7.800000
Lisbon 5.571429
Lisbon 5.333333
Lisbon 4.666667
Lisbon 3.500000
Lisbon 1.800000
Lisbon 1.500000
Leipzig 7.500000
Leipzig 2.166667
Leipzig 1.500000

Najnizi peron za grad Lodz je 2.250000

Menjamo peron za grad Lodz sa 2.25 na 0.1 => 2.250000

Najnizi peron za grad Lodz je sada 0.100000

Hajde sada jedan peron za koji smo sigurni da ne postoji: Belgrade -1 na 0.1 => 0.000000

Ispisujemo sada stablo nakon brisanja:

Zadaci koji se iz bilo kog razloga ne kompajliraju se neće bodovati. Strogo je zabranjeno korišćenje mobilnih telefona, interneta i pametnih satova. Njihova upotreba biće kažnjena sa 0 bodova na kolokvijumu i zabranom izlaska na sledeće provere znanja.

Spisak dozvoljenih bibliotečkih funkcija (u nastavku) važi generalno. Dozvoljeno je implementirati svoju verziju bibliotečkih funkcija koje nisu dozvoljene.

- **stdio.h:** printf, scanf, fgets, puts, fputs
- **stdlib.h:** malloc, calloc, realloc, free
- **string.h:** strlen, strcpy, strcat, strstr, strchr, strcmp (strtok je dozvoljeno koristiti samo u 3. zadatku)
- **math.h:** sqrt, abs, pow, ceil, floor, round