

Uvod u programiranje

Ispit – grupa 2

1. (13p) Profesor Dragan je studentu Mihajlu zadao zadatak da napiše program za igru sudoku. Međutim, Mihajlo "nema mnogo vremena" da se bavi time, pa je odlučio da deo vezan za završnu proveru "prepiše" od studenata sa predmeta Uvod u programiranje. Program koji je Mihajlu potreban treba da proveri da li je uneta 9x9 matrica sa konzole (ne treba unositi dimenzije) rešena sudoku igra i ispiše "da" ili "ne" u zavisnosti od rezultata te provere. Prepostaviti da su uneti samo brojevi od 1 do 9. Sudoku igra je rešena ako su ispunjeni sledeći uslovi:

1. ni u jednom redu matrice se ne može naći dva ili više istih elemenata
2. ni u jednoj koloni matrice se ne može naći dva ili više istih elemenata
3. ni u jednoj 3x3 podmatrici sa početnim indeksom reda i kolone deljivim sa 3 (gornji levi element treba da bude na indeksu 0/0 ili 0/3 ili 0/6 ili 3/3 ili 3/6 itd) se ne može naći dva ili više istih elemenata

Napomena: Nije dozvoljeno korišćenje pomoćnih nizova i matrica.

Primer kada treba ispisati "da":	Primer kada treba ispisati "ne":
5 3 4 6 7 8 9 1 2	5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8	6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7	1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3	8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1	4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6	7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4	9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5	2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9	3 4 5 2 8 6 1 7 5
	(greške su označene boldom)

Dodatno pitanje za razbijanje tenzije na ispitu: Koju ocenu je profesor Dragan 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 hteo da smišlja zadatke). On je odlučio da motivacione poruke budu u heksadecimalnoj formi, te da studenti mogu da ih pročitaju samo ako napišu program koji učitava stringove sastavljene samo od cifara i malih slova a, b, c, d, e i f i pretvara ih u čitljiv tekst. Pretvaranje ovakvog stringa u čitljiv tekst se radi na sledeći način:

1. u svakoj iteraciji se uzimaju po 2 karaktera iz stringa
 2. traži se broj u dekadnom sistemu koji odgovara formiranim heksadecimalnim broju
 3. broj u dekadnom sistemu se pretvara u odgovarajući karakter po ASCII kodu
- Omogućiti unos stringova kroz petlju sve dok korisnik ne unese string koji sadrži još nešto osim validnih karaktera. Taj string na kraju ne pretvarati u čitljiv tekst.

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

497370697420555550

Ispit UUP

537265636e6f6f

Srecnoo

31323320616263

123 abc

cg

(string sadrži karaktere koji nisu validni, pa se ovde program završava)

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 ime[30];  char grupa[15];  struct Element *levo, *desno;  
} Element;
```

U tekstualnoj datoteci "ulaz.txt" nalazi se lista učenika u školi Hogvorts (ime učenika i grupa su 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 grupi rastuće, a ako dva učenika pripadaju istoj grupi onda po njihovim imenima opadajuće. Grupa je string koji počinje velikim slovom, dok su svi ostali karakteri mala slova. Ukoliko ovaj uslov nije zadovoljen, u polje grupa u strukturi upisati prazan string i omogućiti da on bude dodat na krajnji desni deo stabla. Iz funkcije je potrebno vratiti stablo nakon dodavanja svih elemenata.

b) (2p) **void ispisiDeoStabla(Element* stablo, char* grupa)**. Ispisati imena svih učenika iz stabla koji pripadaju grupi. Imena ispisivati rastuće.

c) (3p) **char* nadjiGrupuZaUcenika(Element* stablo, char* ucenik)**. Ispisati u kojoj se grupi nalazi učenik koristeći podatke iz stabla.

d) (8p) **char* promeniGrupu(Element* stablo, char* ucenik, char* novaGrupa)**. Promeniti grupu u kojoj se nalazi učenik koristeći podatke iz stabla. Grupa u kojoj se učenik treba nalaziti nakon izvršavanja funkcije je novaGrupa, a povratna vrednost funkcije treba da bude grupa u kojoj se do tada nalazio.

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 ucenike iz grupe Gryffindor:

Aberforth Dumbledore

Albus Dumbledore

Alice Longbottom

Alicia Spinnet

Andrew Kirke

Angelina Johnson

Charity Burbage

Colin Creevey

Cormac McLaggen

Cuthbert Binns

(još redova)

Ispisujemo sve ucenike koji nisu ni iz jedne grupe:

Alphard Black

Borgin

Burke

Elsbeth McGillicuddy

Flora Carrow

Hestia Jones

Ignatius Prewett

Ignotus Peverell

(još redova)

Harry Potter se nalazi u grupi Gryffindor

Heidi Macavoy se nalazio u grupi Gryffindor

Heidi Macavoy se sada nalazi u grupi Slytherin

Ispisujemo deo stabla nakon brisanja memorije:

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