

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se potpisati na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Rezultati i uvid u kolokvije:** ponedjeljak, 17. lipnja 2013., u 12 sati.

### 1. zadatak

(30 bodova) Pončo i Toro sjede na prvom u nizu od  $n$  lopoča u bari, i žele doći na posljednji. Uvijek se kreću u istom smjeru: tromiji Toro se jednim skokom pomiče za 1 ili 2 lopoča, dok se okretniji Pončo pomiče za do 4 lopoča u jednom skoku. No, Pončo se boji da ne propadne u vodu, i staje samo na lopoče na kojima je Toro već bio. U glavnom programu učitajte  $n$  i ispišite broj načina na koje oni to mogu učiniti, modulo broj vrijednosti prikazivih u tipu `long`. Načine razlikujemo samo po tome koja žaba je skočila na koji lopoč, ne i po redosljedu skakanja.

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 2. zadatak

(25 bodova) U svakom elementu vezane liste želimo čuvati osobno ime (jedan string od najviše 12 znakova) i broj cipela (jedan cijeli broj).

- (a) (5 bodova) Napišite definiciju tipa podataka za pojedini element liste tako da bude moguće deklarirati varijable naredbom `element x`;
- (b) (5 bodova) Napišite funkciju `void ispisi (element *first, int a)` koja ispisuje imena onih elemenata liste (zadane pokazivačem `first` na prvi element) kojima je broj cipela strogo veći od `a`.
- (c) (10 bodova) Napišite funkciju `brisi()` koja iz liste briše sve elemente kojima ime završava samoglasnikom. Argumente i povratni tip funkcije `brisi()` sami odredite.
- (d) (5 bodova) Napišite funkciju `dodaj()` koja na početak liste dodaje element koji sadrži zadano osobno ime i zadani broj cipela (zadaju se kao argumenti funkcije). Argumente i povratni tip funkcije `dodaj()` sami odredite.

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(30 bodova) Matricu dimenzije  $r \times s$  popunjenu znakovima '.', 'o', '<', '>', '^', 'v' nazivamo *bombonijera*. Od sljedećih konfiguracija unutar takve matrice,

1. >o<            2. v  
                      o  
                      ^
3. v  
                      >o<  
                      ^

prve dvije nazivamo *cijelim slatkišima*, a smijete pretpostaviti da se treća ne pojavljuje. U matrici se mogu nalaziti i komadi slatkiša koji ne pripadaju ni jednom cijelom slatkišu.

Niz bombonijera zapisan je u tekstualnoj datoteci na sljedeći način. U prvoj liniji je broj  $t$  bombonijera u datoteci (cijeli broj). Zapis svake bombonijere počinje praznom linijom. Iza nje je linija s dva cijela broja  $r$  i  $s$  (dimenzije), nakon čega slijedi  $r$  redaka od po  $s$  znakova koji tvore bombonijeru.

Napišite program koji prima dva argumenta komandne linije. Oba argumenta su nazivi tekstualnih datoteka. U prvoj (ulaznoj) je već zapisan niz bombonijera, a u drugu (izlaznu) treba za svaku bombonijeru, u zasebnom retku napisati broj cijelih slatkiša u njoj (v. primjer).

Napomene: U ovom zadatku smijete koristiti pomoćne nizove. Smijete pretpostaviti da dimenzije  $r$  i  $s$  ne prelaze 400. Nemojte uvoditi ograničenja na broj  $t$  (broj bombonijera u ulaznoj datoteci).

### Primjer:

ulaz	izlaz
2	3
	2
5 4	
.>o<	
v.^.	
ooo.	
^.^.	
>o<<	
3 4	
...v	
>o<o	
...^	

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(30 bodova) Za obradu rezultata golf turnira, podatke o jednom turniru pamtimo u obliku strukture koja sadrži ime turnira (string od 30 znakova) i polje podataka za 72 igrača, a za svakog igrača pamtimo ime (string od 30 znakova) i njegovu zaradu na turniru (tipa `double`).

- (a) (4 boda) Definirajte tip `turnir` za takvu strukturu koja sadrži samo navedene podatke, tako da bude moguća deklaracija varijable `turnir t`;
- (b) (8 bodova) Napišite funkciju `poredaj` koja prima jedan turnir kao argument, sortira sve igrače silazno po zaradi i vraća tako sortirani turnir.
- (c) (8 bodova) Napišite funkciju `zarada` koja prima ime igrača i ime datoteke turnira. Funkcija treba vratiti ukupnu zaradu tog igrača na svim turnirima u binarnoj datoteci zadanog imena.
- (d) (10 bodova) Napišite funkciju `uredi` koja prima pokazivač na već otvorenu binarnu datoteku turnira. Funkcija treba sortirati igrače na svim turnirima u toj datoteci, koristeći direktno binarno pisanje, te funkciju iz (b).

Uputa: Funkcija za “direktno” pozicioniranje u datoteci je

```
int fseek(FILE *fp, long int offset, int origin);
```

gdje je `offset` pomak u byteovima, a `origin` “ishodište” za pomak = jedna od vrijednosti: `SEEK_SET` (početak), `SEEK_CUR` (trenutna pozicija) ili `SEEK_END` (kraj). Izlaz različit od nule je signal greške.

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, mobiteli, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se potpisati na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Rezultati i uvid u kolokvije:** ponedjeljak, 17. lipnja 2013., u 12 sati.

### 1. zadatak

(30 bodova) Dva žapca, Fatso i Banjo, preskaču s jedne obale bare na drugu, preko  $n$  lopoča koji se u njoj nalaze. Uvijek se kreću u istom smjeru: potpuniji Fatso se jednim skokom pomiče na sljedeći lopoč ili preskače jedan, dok agilniji Banjo može preskočiti i do 3 lopoča u jednom skoku. Ipak, Banjo ide na sigurno, i ne želi stati na lopoče koje je Fatso preskočio. U glavnom programu učitajte  $n$  i ispišite broj načina na koje oni to mogu učiniti, modulo broj vrijednosti prikazivih u tipu `int`. Načine razlikujemo samo po tome koja žaba je skočila na koji lopoč, ne i po redoslijedu skakanja.

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_ JMBAG: \_\_\_\_\_

### 2. zadatak

(25 bodova) U svakom elementu vezane liste želimo čuvati kućni broj (jedan cijeli broj) i ime ulice (jedan string od najviše 23 znaka).

- (a) (5 bodova) Napišite definiciju tipa podataka za pojedini element liste tako da bude moguće deklarirati varijable naredbom `element x;`
- (b) (5 bodova) Napišite funkciju `void ispisi (element *first, const char s[])` koja ispisuje kućne brojeve onih elemenata liste (zadane pokazivačem `first` na prvi element) kojima je ime ulice jednako stringu `s`.
- (c) (10 bodova) Napišite funkciju `brisi()` koja iz liste briše prvi element kojemu je kućni broj strogo veći od aritmetičke sredine kućnih brojeva svih elemenata liste. Argumente i povratni tip funkcije `brisi()` sami odredite.
- (d) (5 bodova) Napišite funkciju `dodaj()` koja na kraj liste dodaje element koji sadrži zadani kućni broj i zadano ime ulice (zadaju se kao argumenti funkcije). Argumente i povratni tip funkcije `dodaj()` sami odredite.

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 3. zadatak

(30 bodova) Matricu dimenzije  $r \times s$  popunjenu znakovima '.', '0', '[', ']', 'u', 'n' nazivamo *bombonijera*. Od sljedećih konfiguracija unutar takve matrice,

2. u	1. ]0[	3. u
0		]0[
n		n

prve dvije nazivamo *cijelim slatkišima*, a smijete pretpostaviti da se treća ne pojavljuje. U matrici se mogu nalaziti i komadi slatkiša koji ne pripadaju ni jednom cijelom slatkišu.

Niz bombonijera zapisan je u tekstualnoj datoteci na sljedeći način. U prvoj liniji je broj  $t$  bombonijera u datoteci (cijeli broj). Zapis svake bombonijere počinje praznom linijom. Iza nje je linija s dva cijela broja  $r$  i  $s$  (dimenzije), nakon čega slijedi  $r$  redaka od po  $s$  znakova koji tvore bombonijeru.

Napišite program koji prima dva argumenta komandne linije. Oba argumenta su nazivi tekstualnih datoteka. U prvoj (ulaznoj) je već zapisan niz bombonijera, a u drugu (izlaznu) treba za svaku bombonijeru, u zasebnom retku napisati broj cijelih slatkiša u njoj (v. primjer).

Napomene: U ovom zadatku smijete koristiti pomoćne nizove. Smijete pretpostaviti da dimenzije  $r$  i  $s$  ne prelaze 200. Nemojte uvoditi ograničenja na broj  $t$  (broj bombonijera u ulaznoj datoteci).

### Primjer:

ulaz	izlaz
2	2
	3
3 4	
u...	
0]0[	
n...	
5 4	
]0[[	
u.n.	
000.	
n.n.	
.]0[	

## Programiranje 2 – popravni kolokvij, 14. lipnja 2013.

Ime i prezime: \_\_\_\_\_

JMBAG: \_\_\_\_\_

### 4. zadatak

(30 bodova) Za obradu rezultata teniskih turnira, podatke o jednom turniru pamtimo u obliku strukture koja sadrži ime turnira (string od 20 znakova) i polje podataka za 128 igrača, a za svakog igrača pamtimo ime (string od 20 znakova) i njegov broj bodova na turniru (tipa `int`).

- (a) (4 boda) Definirajte tip `turnir` za takvu strukturu koja sadrži samo navedene podatke, tako da bude moguća deklaracija varijable `turnir t;`.
- (b) (8 bodova) Napišite funkciju `poredaj` koja prima jedan turnir kao argument, sortira sve igrače uzlazno po broju bodova i vraća tako sortirani turnir.
- (c) (8 bodova) Napišite funkciju `bodovi` koja prima ime igrača i ime datoteke turnira. Funkcija treba vratiti ukupan broj bodova tog igrača na svim turnirima u binarnoj datoteci zadanog imena.
- (d) (10 bodova) Napišite funkciju `uredi` koja prima pokazivač na već otvorenu binarnu datoteku turnira. Funkcija treba sortirati igrače na svim turnirima u toj datoteci, koristeći direktno binarno pisanje, te funkciju iz (b).

Uputa: Funkcija za “direktno” pozicioniranje u datoteci je

```
int fseek(FILE *fp, long int offset, int origin);
```

gdje je `offset` pomak u byteovima, a `origin` “ishodište” za pomak = jedna od vrijednosti: `SEEK_SET` (početak), `SEEK_CUR` (trenutna pozicija) ili `SEEK_END` (kraj). Izlaz različit od nule je signal greške.