

## Programiranje 2 – popravni kolokvij, 1. 9. 2015.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 6.9., navečer (na webu), a uvidi u ponedjeljak, 7.9., u 10 sati.

**Napomene:** Sva rješenja napišite isključivo na papir sa zadatkom jer jedino njega predajete. Pomoćne račune smijete raditi na drugim papirima koje će vam dati dežurni asistent. U svim zadacima nije dopušteno korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje te službenog podsjetnika. Kalkulatori i razne tablice, papiri i sl. nisu dozvoljeni. **Mobitele ugasite i spremite;** nisu dozvoljeni kao zamjena za sat!

**Zadatak 1** (30 bodova) Dijete se penje po ljestvama i može se u jednom koraku pomaknuti za jednu, dvije ili tri prečke naviše. No, neke su prečke slomljene i dijete na njih ne može stati. Napišite rekurzivnu funkciju koja, kao argumente, prima prirodan broj  $n$  (broj prečki) i niz `precke[]`, koji ima jedinice na onim mjestima na kojima su prečke slomljene, a nule na ostalim mjestima, i vraća broj načina na koji se dijete može popesti po ljestvama. Na početku, dijete se nalazi ispred prve prečke, a mora stići iza zadnje prečke.

**Zadatak 2** (25 bodova) Posada trajekta Lošinj–Premuda–Silba–Olib–Zadar odlučila je modernizirati način na koji se vodi evidencija o automobilima koje trajekt prevozi. Informacije o svakom automobilu bit će spremljene u odgovarajuću strukturu, a strukture će biti povezane u vezanu listu.

- (a) U svakom elementu liste želimo čuvati registraciju automobila (string od najviše 10 znakova) i oznaku njegovog odredišta (`char`). Deklarirajte tip podataka `automobil` koji omogućuje kreiranje vezane liste automobila.
- (b) Napišite funkciju `automobil* dodaj(automobil* first)`; koja učitava registraciju automobila (string od najviše 10 znakova) i jedan znak ('P', 'S', 'O' ili 'Z') kao oznaku odredišta. Nakon učitavanja, treba ubaciti novi automobil u vezanu listu, i to neposredno ispred prvog automobila u listi koji ima istu oznaku odredišta. Ukoliko takav automobil ne postoji, novi automobil možete dodati na početak liste.
- (c) Napišite funkciju `automobil* izbaci(automobil* first)`; koja iz vezane liste izbacuje sve automobile čija registracija nije zapisana u ispravnom formatu: XX-yyyy-XX (svaki X predstavlja po jedno veliko slovo, a svaki y po jednu znamenku 0-9).
- (d) Napišite funkciju `automobil* iskrcavanje(automobil* first, char lu)`; Ovdje je `first` pokazivač na početak vezane liste, a `lu` oznaka luke u koju je trajekt upravo pristao. Funkcija treba iskrcati (maknuti iz liste) sve automobile čije je odredište luka u koju je trajekt upravo pristao.  
Napomena: Pretpostavljamo da je lista nastajala kao u (b) zadatku, stoga svi auti s istim odredištem u listi čine jedan "blok".

**Zadatak 3** (30 bodova) Dvojica dječaka igraju igru *potapanje podmornica*, tako da u datotekama `igrac1.txt`, odnosno, `igrac2.txt` postavljaju svoje podmornice na cjelobrojnom kvadratnom polju, na način da u prvom retku napišu dimenziju kvadratne ploče, a potom, u sljedećim retcima, redak po redak kvadratnog polja. Točka (znak `'.'`) označava prazno polje, dok zvjezdica (znak `'*'`) označava polje koje pripada podmornici. Podmornice su pravokutnog oblika, postavljaju se horizontalno i/ili vertikalno, te između svake dvije podmornice u horizontalnim i vertikalnim smjerovima treba biti barem jedno prazno polje, ali se smiju dodirivati u jednoj točki (nekom vrhu polja, kao na ploči drugog igrača u donjem primjeru).

Problem je što se dječaci nisu unaprijed dogovorili kolika je dimenzija polja, koje su moguće dimenzije podmornica ( $1 \times 1$ ,  $1 \times 2$ ,  $2 \times 2$ , ...), te koji je broj podmornica određenog tipa, pa je prije početka same igre potrebno provjeriti jesu li ploče dječaka *kompatibilne*, odnosno, jesu li na pločama zastupljeni isti tipovi podmornica, te je li jednak broj podmornica za svaki zastupljeni tip. Podmornice su invarijantne na rotacije (npr. podmornice dimenzija  $1 \times 2$  i  $2 \times 1$  su istog tipa).

Na primjer, sljedeće ploče

<code>igrac1.txt</code>	<code>igrac2.txt</code>
<code>4</code>	<code>4</code>
<code>...*</code>	<code>.**.</code>
<code>*...</code>	<code>*...</code>
<code>*.**</code>	<code>*..*</code>
<code>..**</code>	<code>*...</code>

nisu kompatibilne, iako su na obje ploče 3 podmornice, jer na prvoj ploči postoji podmornica tipa  $2 \times 2$ , koja se ne pojavljuje na drugoj ploči.

- Spremite igraće ploče u dinamički alocirano polje redaka ploče (dinamičko dvodimenzionalno polje charova).
- Definirajte tip podatka `podmornica` koji će sadržavati varijable `a` i `b`, tipa `int`, koje označavaju dimenziju podmornice (kako je tip podmornice invarijantan na rotacije, uzmite da je  $a \leq b$ ), te varijablu `kolicina`, tipa `int`, koja sadrži broj podmornica tipa  $a \times b$  na odgovarajućoj ploči.
- Definirajte dinamičke nizove `podmornice1` i `podmornice2`, tipa `podmornica`, koji će sadržavati podatke o podmornicama na prvoj, odnosno, drugoj igraćoj ploči. Pripazite da alocirate točno onoliko memorije koliko je potrebno.
- Koristeći polja iz prethodnog dijela, ispitate jesu li igraće ploče kompatibilne.

**Zadatak 4** (30 bodova) Putnička agencija drži ponude zapisane u tekstualnoj datoteci. Svaki redak odgovara jednoj ponudi i oblika je **\*\* destinacija \*\* opis ponude | cijena |**, pri čemu je cijena zapisana kao realan broj s dva decimalna mjesta. Na primjer

**\*\* Pariz \*\* Let avionom, razgledavanje Elizejskih poljana, pet dana. | 2999.99 |**

Svaka linija sadrži najviše 1000 znakova. Napišite program koji učitava, kao argumente komandne linije, imena datoteke s ponudama i izlazne datoteke, te destinaciju. Program treba prepisati, iz datoteke ponude u izlaznu datoteku, sve retke koji sadrže učitavanu destinaciju i to tako da budu sortirani po cijeni uzlazno. U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.

## Programiranje 2 – popravni kolokvij, 1. 9. 2015.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 6.9., navečer (na webu), a uvidi u ponedjeljak, 7.9., u 10 sati.

**Napomene:** Sva rješenja napišite isključivo na papir sa zadatkom jer jedino njega predajete. Pomoćne račune smijete raditi na drugim papirima koje će vam dati dežurni asistent. U svim zadacima nije dopušteno korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje te službenog podsjetnika. Kalkulatori i razne tablice, papiri i sl. nisu dozvoljeni. **Mobitele ugasite i spremite;** nisu dozvoljeni kao zamjena za sat!

**Zadatak 1** (30 bodova) Čovjek prelazi rijeku preko visećeg mosta i može se u jednom koraku pomaknuti za jednu, dvije ili tri daske unaprijed. No, neke su daske slomljene i čovjek na njima ne može stati. Napišite rekurzivnu funkciju koja, kao argumente, prima prirodan broj  $n$  (broj dasaka) i niz `daske[]`, koji ima jedinice na onim mjestima na kojima su daske slomljene, a nule na ostalim mjestima, i vraća broj načina na koji čovjek može prijeći preko mosta. Na početku, čovjek se nalazi ispred prve daske, a mora stići iza zadnje daske.

**Zadatak 2** (25 bodova) Konduktar u autobusu Zagreb–Karlovac–Delnice–Rijeka odlučio je modernizirati način na koji se vodi evidencija o putnicima u njegovom autobusu. Informacije o svakom putniku bit će spremljene u odgovarajuću strukturu, a strukture će biti povezane u vezanu listu.

- (a) U svakom elementu liste želimo čuvati šifru putnika (string od najviše 14 znakova) i oznaku njegovog odredišta (`char`). Deklarirajte tip podataka `putnik` koji omogućuje kreiranje vezane liste putnika.
- (b) Napišite funkciju `putnik* dodaj(putnik* first)`; koja učitava šifru putnika (string od najviše 14 znakova) i jedan znak ('K', 'D' ili 'R') kao oznaku odredišta. Nakon učitavanja, novog putnika treba ubaciti u vezanu listu, i to neposredno ispred prvog putnika u listi koji ima istu oznaku odredišta. Ukoliko takav putnik ne postoji, novog putnika možete dodati na početak liste.
- (c) Napišite funkciju `putnik* izbaci(putnik* first)`; koja iz vezane liste izbacuje sve putnike čija šifra nije zapisana u ispravnom formatu. Ispravna šifra se sastoji od dva velika slova (inicijali putnika) i 11-znamenkastog broja (OIB), odvojenih crticom ('-'), primjerice, AB-01234567890.
- (d) Napišite funkciju `putnik* iskrcavanje(putnik* first, char dest)`; Ovdje je `first` pokazivač na početak vezane liste, a `dest` oznaka odredišta na koje je autobus upravo stigao. Funkcija treba iskrcati (maknuti iz liste) sve putnike čije je odredište grad u koji je autobus upravo stigao.  
Napomena: Pretpostavljamo da je lista nastajala kao u (b) zadatku, stoga svi putnici s istim odredištem u listi čine jedan "blok".

**Zadatak 3** (30 bodova) Dvojica prijatelja igraju igru *potapanje brodova*, tako da u datotekama `ploca1.txt`, odnosno `ploca2.txt` postave svoje brodove na cjelobrojnom kvadratnom polju, na način da u prvom retku napišu dimenziju kvadratne ploče, a potom, u sljedećim retcima, redak po redak kvadratnog polja. Crtica (znak '-') označava prazno polje, dok nula (znak '0') označava polje koje pripada brodu. Brodovi su pravokutnog oblika, postavljaju se horizontalno i/ili vertikalno, te između svaka dva broda u horizontalnim i vertikalnim smjerovima treba biti barem jedno prazno polje, ali se smiju dodirivati u jednoj točki (nekom vrhu polja, kao na ploči prvog igrača u donjem primjeru).

Problem je što se prijatelji nisu unaprijed dogovorili kolika je dimenzija polja, koje su moguće dimenzije brodova ( $1 \times 1$ ,  $1 \times 2$ ,  $2 \times 2$ , ...), te koji je broj brodova određenog tipa, pa je prije početka same igre potrebno provjeriti jesu li ploče prijatelja *kompatibilne*, odnosno, jesu li na pločama zastupljeni isti tipovi brodova, te je li jednak broj brodova za svaki zastupljeni tip. Brodovi su invarijantni na rotacije (npr. brodovi dimenzija  $1 \times 2$  i  $2 \times 1$  su istog tipa).

Na primjer, sljedeće ploče

<code>ploca1.txt</code>	<code>ploca2.txt</code>
4	4
-0-0	00--
-0--	00-0
-0--	----
--00	000-

nisu kompatibilne, iako su na obje ploče 3 broda, jer na prvoj ploči postoji brod tipa  $1 \times 2$ , koji se ne pojavljuje na drugoj ploči.

- Spremite igraće ploče u dinamički alocirano polje redaka ploče (dinamičko dvodimenzionalno polje charova).
- Definirajte tip podatka `brod` koji će sadržavati varijable `k` i `l`, tipa `int`, koje označavaju dimenziju broda (kako je tip broda invarijantan na rotacije, uzmite da je  $k \leq l$ ), te varijablu `broj`, tipa `int`, koja sadrži broj brodova tipa  $k \times l$  na odgovarajućoj ploči.
- Definirajte dinamičke nizove `brodovi1` i `brodovi2`, tipa `brod`, koji će sadržavati podatke o brodovima na prvoj, odnosno, drugoj igraćoj ploči. Pripazite da alocirate točno onoliko memorije koliko je potrebno.
- Koristeći polja iz prethodnog dijela, ispitajte jesu li igraće ploče kompatibilne.

**Zadatak 4** (30 bodova) Putnička agencija drži ponude zapisane u tekstualnoj datoteci. Svaki redak odgovara jednoj ponudi i oblika je -> destinacija -> opis ponude \$ cijena \$ , pri čemu je cijena zapisana kao realan broj s dva decimalna mjesta. Na primjer

-> Atena -> Let avionom, razgledavanje Akropole, pet dana. \$ 3100.99 \$

Svaka linija sadrži najviše 1000 znakova. Napišite program koji učitava, kao argumente komandne linije, imena datoteke s ponudama i izlazne datoteke, te destinaciju. Program treba prepisati, iz datoteke ponude u izlaznu datoteku, sve retke koji sadrže učitavanu destinaciju i to tako da budu sortirani po cijeni silazno. U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.