

Programiranje 2 – drugi kolokvij, 15. 6. 2015.

Rezultati i uvidi u kolokvije: Rezultati u ponedjeljak, 22.6., navečer na webu, a uvidi u utorak, 23.6., u 14 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 (10 + 10 = 20 bodova) Definirana je struktura `karta` za pohranu podataka o igračkoj karti, oblika

```
typedef struct {
    int vrijednost;
    char boja;
} karta;
```

Vrijednost karte je cijeli broj između 1 (as) i 13 (kralj), a boja karte znak 't' (tref), 'p' (pik), 'k' (karo) ili 'h' (herc).

- Napišite funkciju `karta promijesaj(...)` koja prima niz od 52 različite karte, promiješa ih na slučajan način te vraća prvu kartu u špil nakon miješanja.
- Napišite funkciju `int ima_spil(...)` koja prima pokazivač na datoteku karata, već otvorenu za binarno čitanje (možete pretpostaviti da se pokazivač nalazi na početku datoteke). Funkcija treba vratiti 1 ako se iz svih karata u toj datoteci može sastaviti barem jedan potpuni špil karata (sadrži 52 različite karte). U suprotnom, funkcija vraća 0.

Za potrebe ovog zadatka možete/morate koristiti dodatne nizove te funkciju `rand` iz biblioteke `stdlib.h`.

Zadatak 2 ($2 + 3 + 7 + 8 = 20$ bodova) Hrvatske željeznice podatke o linijama imaju spremljene u vezanim listama tako da svaki element vezane liste predstavlja jednu postaju i sadrži naziv (niz znakova od najviše 20 znakova) i cjelobrojnu vrijednost vrijeme koja označava vrijeme u sekundama potrebno vlaku da prijeđe put između prethodne i te postaje u određenoj liniji. U prvoj (odnosno, polaznoj) postaji, varijabla vrijeme ima vrijednost 0. Nazivi postaja se razlikuju.

- Definirajte tip podatka postaja koji će služiti za pohranu jednog elementa liste, na način da bude moguće deklarirati varijablu postaja x.
- Napišite funkciju rednibr() koja prima pokazivač na početak vezane liste prvi i naziv postaje x, te vraća redni broj postaje u pripadnoj liniji (prva postaja ima redni broj 1). Ukoliko ne postoji postaja naziva x, funkcija treba vratiti 0.
- Napišite funkciju povratak() koja, po adresi, prima pokazivač na početak vezane liste i invertira vezanu listu tako da odgovara povratnoj liniji (ako je ulazna vezana lista predstavljala liniju $A \rightarrow B$, funkcija je treba preoblikovati da predstavlja liniju $B \rightarrow A$). Funkcija ne vraća ništa kao funkcijsku vrijednost. Pazite da napravite i odgovarajuću promjenu u varijablama vrijeme.
- Napišite funkciju prototipa

`postaja* put(postaja** prvi, const char* x, const char* y, int* t)`

koja (ako je potrebno) invertira vezanu listu tako da postaja x bude manjeg rednog broja od postaje y, vraća pokazivač na postaju x i preko varijabilnog parametra t vraća vrijeme potrebno vlaku da iz postaje x dođe u postaju y. Ukoliko jedna od postaja ne postoji u navedenoj liniji, funkcija vraća NULL pokazivač i vrijeme postavlja na -1 .

Dozvoljeno je korištenje funkcija iz biblioteke `string.h`. Podzadatak a) je nužan za rješavanje preostalih podzadataka, dok funkcije iz podzadataka b) i c) možete koristiti u podzadatku d) i ako ih niste napisali.

Zadatak 3 (3 + 6 + 6 = 15 bodova) Unos u rječniku hrvatskih riječi sastoji se od tri informacije: same riječi, informacije o vrsti riječi (imenica, glagol, ...) te opisa njezinog značenja.

a) Definirajte tip podatka `unos` koji će služiti za čuvanje jednog unosa iz rječnika hrvatskih riječi. (Sve tri potrebne informacije možete reprezentirati stringovima, no uočite da opis riječi može sadržavati i razmake). Možete pretpostaviti da riječi nisu dulje od 30 znakova, te da informacija o vrsti riječi nije dulja od 10 znakova. S druge strane, nema ograničenja na duljinu opisa pojedine riječi.

b) Napišite funkciju `void novi(unos *x)` koja će omogućiti korisniku da s tipkovnice učitava novi unos u formatu
`riječ (vrsta) - opis`

primjerice:

`skakač (imenica) - jedna od figura u šahu`

te spremi odgovarajući unos u varijablu koju smo funkciji prenijeli po adresi.

c) Napišite funkciju `void objasni(unos *rjecnik, int r, char** tekst, int d, int n)` koja prima rječnik (ostvaren kao niz podataka tipa `unos`, duljine r), tekst (ostvaren kao niz od d stringova) i prirodni broj n . Funkcija treba pronaći n -tu riječ u tekstu te u rječniku naći odgovarajući unos. Ukoliko je unos pronađen, treba ga ispisati (u formatu opisanom u b) zadatku); u suprotnom, treba ispisati odgovarajuću poruku. Možete pretpostaviti da je $n \leq d$ te da je rječnik sortiran leksikografski uzlazno.

Za dodatnih 5 bodova (ne ulaze u 80%, tj. 80% = 12 bodova) pronađite riječ u rječniku u logaritamskoj složenosti.

Zadatak 4 (20 bodova) Zračna luka koristi tekstualnu datoteku u kojoj vodi evidenciju o prometu na sljedeći način. U svakoj liniji zapisana su tri podatka, odvojena razmacima. Prvi je oznaka leta, koja je niz od najviše 10 znakova. Zatim, vrijeme polijetanja, u formatu hh:mm, te je na kraju zapisano odredište, koje je niz od najviše 30 znakova i može sadržavati razmake. Primjer jedne linije je

52YP238 09:00 ZAGREB

Napišite program koji učitava, kao argumente komandne linije, imena datoteke evidencije i izlazne datoteke. Program treba prepisati linije iz datoteke evidencije u izlaznu datoteku, tako da ih sortira leksikografski uzlazno obzirom na odredište.

U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.

Programiranje 2 – drugi kolokvij, 15. 6. 2015.

Rezultati i uvidi u kolokvije: Rezultati u ponedjeljak, 22.6., navečer na webu, a uvidi u utorak, 23.6., u 14 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 (zaglavljje `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 (10 + 10 = 20 bodova) Definirana je struktura `slicica` za pohranu podataka o sličicama za album, oblika

```
typedef struct {
    int broj;
    char dio;
} slicica;
```

Broj sličice je cijeli broj između 1 i 20, a dio je znak 'p' (plavi), 'c' (crveni), 'z' (zeleni) ili 'n' (narančasti), koji označava u kojem dijelu albuma se nalazi sličica.

- Napišite funkciju `slicica promijesaj(...)` koja prima niz od 80 različitih sličica, promiješa ih na slučajan način te vraća prvu kartu u špilju nakon miješanja.
- Napišite funkciju `int ima_album(...)` koja prima pokazivač na datoteku sličica, već otvorenu za binarno čitanje (možete pretpostaviti da se pokazivač nalazi na početku datoteke). Funkcija treba vratiti 1 ako se iz svih sličica u toj datoteci može kompletirati barem jedan potpuni album (sadrži 80 različitih sličica). U suprotnom, funkcija vraća 0.

Za potrebe ovog zadatka možete/morate koristiti dodatne nizove te funkciju `rand` iz biblioteke `stdlib.h`.

Zadatak 2 ($2 + 3 + 7 + 8 = 20$ bodova) Jedna kurirska kompanija podatke o dostavnim linijama ima spremljene u vezanim listama tako da svaki element vezane liste predstavlja jedno dostavno mjesto i sadrži šifru (niz znakova od najviše 15 znakova) i cjelobrojnu vrijednost `vrijeme` koja označava vrijeme u sekundama potrebno dostavljaču da s prethodnog dostavnog mjesta dođe na to dostavno mjesto u određenoj dostavnoj liniji. Za prvo dostavno mjesto, varijabla `vrijeme` ima vrijednost 0. Šifre dostavnih mjesta se razlikuju.

- Definirajte tip podatka `mjesto` koji će služiti za pohranu jednog elementa liste, na način da bude moguće deklarirati varijablu `mjesto a`.
- Napišite funkciju `rednibr()` koja prima pokazivač na početak vezane liste `prvi` i šifru dostavnog mjesta `s`, te vraća redni broj dostavnog mjesta u pripadnoj dostavnoj liniji (prvo dostavno mjesto ima redni broj 1). Ukoliko ne postoji dostavno mjesto naziva `s`, funkcija treba vratiti 0.
- Napišite funkciju `suprotan()` koja, po adresi, prima pokazivač na početak vezane liste i invertira vezanu listu tako da odgovara dostavnoj liniji suprotne orijentacije (ako je ulazna vezana lista predstavljala dostavnu liniju $A \rightarrow B$, funkcija je treba preoblikovati da predstavlja dostavnu liniju $B \rightarrow A$). Funkcija ne vraća ništa kao funkcijsku vrijednost. Pazite da napravite i odgovarajuću promjenu u varijablama `vrijeme`.
- Napišite funkciju prototipa

```
mjesto* dostava(mjesto** prvi, const char* a, const char* b, int* t)
```

koja (ako je potrebno) invertira vezanu listu tako da dostavno mjesto `a` bude manjeg rednog broja od dostavnog mjesta `b`, vraća pokazivač na dostavno mjesto `a` i preko varijabilnog parametra `t` vraća vrijeme potrebno dostavljaču da od dostavnog mjesta `a` dođe do dostavnog mjesta `b`. Ukoliko jedno od dostavnih mjesta ne postoji u navedenoj dostavnoj liniji, funkcija vraća NULL pokazivač i vrijeme postavlja na -1 .

Dozvoljeno je korištenje funkcija iz biblioteke `string.h`. Podzadatak a) je nužan za rješavanje preostalih podzadataka, dok funkcije iz podzadataka b) i c) možete koristiti u podzadatku d) i ako ih niste napisali.

Zadatak 3 ($3 + 6 + 6 = 15$ bodova) Recenzija filma sastoji se od četiri podatka: imena filma, imena redatelja, numeričke ocjene (prirodni broj od 1 do 10) i kratkog teksta u kojem je dan osvrt na film.

- a) Definirajte tip podatka `film` koji će služiti za čuvanje recenzije jednog filma. Možete pretpostaviti da ime filma nije dulje od 50 znakova (uočite da naslov može sadržavati razmake) i da ime redatelja nije dulje od 30 znakova. Duljina recenzije pojedinog filma nije unaprijed ograničena.
- b) Napišite funkciju `void novi(film *x)` koja će omogućiti korisniku da s tipkovnice učitava recenziju novog filma te spremi odgovarajući unos u varijablu tipa `film` koju smo funkciji prenijeli po adresi. Nove filmove unosimo u sljedećem formatu:

```
Naslov filma: <naslov>
Redatelj: <ime redatelja>
Ocjena: k
Recenzija: <tekst recenzije>
```

Znakovi `< i >` nisu dio unosa, primjerice: `Naslov filma: Mad Max - Fury road`

- c) Napišite funkciju `void recenziraj(film *popis, int v, char* naslov)` koja prima popis recenzija (ostvaren kao niz podataka tipa `film`, duljine v) i naslov nekog filma. Funkcija treba ispisati recenziju filma s odgovarajućim naslovom (u formatu opisanom u b) zadatku) te naslove svih filmova istog redatelja. U slučaju da film nije pronađen, potrebno je ispisati prikladnu poruku. Možete pretpostaviti da je niz filmova sortirani (leksikografski uzlazno) prema naslovu filma. Za dodatnih 5 bodova (ne ulaze u 80%, tj. $80\% = 12$ bodova) pronađite traženi film u logaritamskoj složenosti.

Zadatak 4 (20 bodova) Trajektna luka koristi tekstualnu datoteku u kojoj vodi evidenciju o prometu na sljedeći način. U svakoj liniji zapisana su tri podatka, odvojena razmacima. Prvi je oznaka broda, koja je niz od najviše 10 znakova. Zatim, vrijeme polaska, u formatu hh:mm, te je na kraju zapisano odredište, koje je niz od najviše 30 znakova i može sadržavati razmake. Primjer jedne linije je

AB640 12:00 SPLIT

Napišite program koji uzme, kao argumente komandne linije, imena datoteke evidencije i izlazne datoteke. Program treba prepisati linije iz datoteke evidencije u izlaznu datoteku, tako da ih sortira leksikografski silazno obzirom na odredište.

U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.

Programiranje 2 – drugi kolokvij, 15. 6. 2015.

Rezultati i uvidi u kolokvije: Rezultati u ponedjeljak, 22.6., navečer na webu, a uvidi u utorak, 23.6., u 14 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 (10 + 10 = 20 bodova) Definirana je struktura `listic` za pohranu podataka o listićima za tombolu, oblika

```
typedef struct {
    int broj;
    char slovo;
} listic;
```

Broj listića je cijeli broj između 1 i 100, a slovo je znak 'a' (auto), 'm' (mobitel), 'v' (vino) ili 't' (torta), koji označava na koju nagradu se odnosi listić.

- Napišite funkciju `listic promijesaj(...)` koja prima niz od 400 različitih listića, promiješa ih na slučajan način te vraća prvi listić u nizu nakon miješanja.
- Napišite funkciju `int ima_izvlačenje(...)` koja prima pokazivač na datoteku listića, već otvorenu za binarno čitanje (možete pretpostaviti da se pokazivač nalazi na početku datoteke). Funkcija treba vratiti 1 ako se od listića u datoteci može sakupiti barem jedan skup od 400 različitih listića da bi se moglo održati izvlačenje nagrada. U suprotnom, funkcija vraća 0.

Za potrebe ovog zadatka možete/morate koristiti dodatne nizove te funkciju `rand` iz biblioteke `stdlib.h`.

Zadatak 2 ($2 + 3 + 7 + 8 = 20$ bodova) Jedna turistička agencija nudi razna krstarenja Jadranom i podatke o tim brodskim linijama ima spremljene u vezanim listama tako da svaki element vezane liste predstavlja jednu luku i sadrži ime (niz znakova od najviše 20 znakova) i cjelobrojnu vrijednost `vrijeme` koja označava vrijeme u sekundama potrebno brodu da prijeđe put između prethodne i te luke u određenoj liniji. U prvoj (odnosno polaznoj) luci, varijabla `vrijeme` ima vrijednost 0. Nazivi luka se razlikuju.

- Definirajte tip podatka `luka` koji će služiti za pohranu jednog elementa liste, na način da bude moguće deklarirati varijablu `luka a`.
- Napišite funkciju `rednibr()` koja prima pokazivač na početak vezane liste `prvi` i naziv luke `a`, te vraća redni broj luke u pripadnoj liniji (polazna luka ima redni broj 1). Ukoliko ne postoji luka naziva `a`, funkcija treba vratiti 0.
- Napišite funkciju `obratno()` koja, po adresi, prima pokazivač na početak vezane liste i invertira vezanu listu tako da odgovara povratnoj brodskej liniji (ako je ulazna vezana lista predstavljala liniju $A \rightarrow B$, funkcija je treba preoblikovati da predstavlja liniju $B \rightarrow A$). Funkcija ne vraća ništa kao funkcijsku vrijednost. Pazite da napravite i odgovarajuću promjenu u varijablama `vrijeme`.
- Napišite funkciju prototipa

```
luka* krstarenje(luka** prvi, const char* a, const char* b, int* t)
```

koja (ako je potrebno) invertira vezanu listu tako da luka `a` bude manjeg rednog broja od luke `b`, vraća pokazivač na luku `a` i preko varijabilnog parametra `t` vraća vrijeme potrebno brodu da iz luke `a` dođe u luku `b`. Ukoliko jedna od luka ne postoji u navedenoj liniji, funkcija vraća `NULL` pokazivač i vrijeme postavlja na `-1`.

Dozvoljeno je korištenje funkcija iz biblioteke `string.h`. Podzadatak a) je nužan za rješavanje preostalih podzadataka, dok funkcije iz podzadataka b) i c) možete koristiti u podzadatku d) i ako ih niste napisali.

Zadatak 3 ($3 + 6 + 6 = 15$ bodova) Za kataloški zapis knjige u knjižnici potrebna su četiri podatka: naslov, ime autora, izdavač i standardni broj (ISBN).

- a) Definišite tip podatka `knjiga` koji će služiti za čuvanje podataka o jednoj knjizi. Sva četiri podatka možete interpretirati kao stringove (uočite da odgovarajući stringovi moraju uključivati i razmake). Duljina naslova knjige nije ograničena, dok za ostale podatke možete postaviti "globalna" ograničenja — primjerice, ISBN se uvijek sastoji od 10 znamenki.
- b) Napišite funkciju `void nova(knjiga *x)` koja će omogućiti korisniku da s tipkovnice učitava podatke o knjizi te spremi odgovarajući unos u varijablu tipa `knjiga` koju smo funkciji prenijeli po adresi. Nove knjige unosimo u sljedećem formatu:

```
Naslov knjige: <naslov>
Autor: <ime autora>
Izdavač: <ime izdavaca>
ISBN: <10 znamenki>
```

Znakovi `< i >` nisu dio unosa, primjerice: `Naslov knjige: The Adventures of Huckleberry Finn`

- c) Napišite funkciju `void ispisi(knjiga *katalog, int k, char* naslov)` koja prima katalog (ostvaren kao niz podataka tipa `knjiga`, duljine `k`) i naslov neke knjige. Funkcija treba ispisati podatke o knjizi odgovarajućeg naslova (u formatu opisanom u b) zadatku) te naslove svih knjiga istog autora. U slučaju da knjiga nije pronađena, potrebno je ispisati prikladnu poruku. Možete pretpostaviti da je katalog sortiran (leksikografski uzlazno) prema naslovima knjiga. Za dodatnih 5 bodova (ne ulaze u 80%, tj. $80\% = 12$ bodova) pronađite knjigu traženog naslova u logaritamskoj složenosti.

Zadatak 4 (20 bodova) Zračna luka koristi tekstualnu datoteku u kojoj vodi evidenciju o prometu na sljedeći način. U svakoj liniji zapisana su tri podatka, odvojena razmacima. Prvi je oznaka leta, koja je niz od najviše 10 znakova. Zatim, vrijeme polijetanja, u formatu hh:mm, te je na kraju zapisano odredište, koje je niz od najviše 30 znakova i može sadržavati razmake. Primjer jedne linije je

51ZP28 11:00 RIJEKA

Napišite program koji učitava, kao argumente komandne linije, imena datoteke evidencije i izlazne datoteke. Program treba prepisati linije iz datoteke evidencije u izlaznu datoteku, tako da ih sortira leksikografski uzlazno obzirom na oznaku leta.

U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.

Programiranje 2 – drugi kolokvij, 15. 6. 2015.

Rezultati i uvidi u kolokvije: Rezultati u ponedjeljak, 22.6., navečer na webu, a uvidi u utorak, 23.6., u 14 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 ($10 + 10 = 20$ bodova) Definirana je struktura `kuglica` za pohranu podataka o kuglicama za loto, oblika

```
typedef struct {
    int broj;
    char boja;
} kuglica;
```

Broj kuglice je cijeli broj između 1 i 10, a boja je znak 'p' (plava), 'c' (crvena), 'z' (zelena) ili 'n' (narančasta).

- Napišite funkciju `kuglica promijesaj(...)` koja prima niz od 40 različitih kuglica, promiješa ih na slučajan način te vraća prvu kuglicu u nizu nakon miješanja.
- Napišite funkciju `int ima_bubanj(...)` koja prima pokazivač na datoteku `kuglica`, već otvorenu za binarno čitanje (možete pretpostaviti da se pokazivač nalazi na početku datoteke). Funkcija treba vratiti 1 ako se od kuglica u datoteci može složiti barem jedan set od 40 različitih kuglica koje idu u bubanj za loto. U suprotnom, funkcija vraća 0.

Za potrebe ovog zadatka možete/morate koristiti dodatne nizove te funkciju `rand` iz biblioteke `stdlib.h`.

Zadatak 2 ($2 + 3 + 7 + 8 = 20$ bodova) Jedan autobusni prijevoznik podatke o linijama ima spremljene u vezanim listama tako da svaki element vezane liste predstavlja jednu stanicu i sadrži **naziv** (niz znakova od najviše 20 znakova) i cjelobrojnu vrijednost **vrijeme** koja označava vrijeme u sekundama potrebno autobusu da prijeđe put između prethodne i te stanice u određenoj liniji. U prvoj stanici, varijabla **vrijeme** ima vrijednost 0. Nazivi stanica se razlikuju.

- Definirajte tip podatka **stanica** koji će služiti za pohranu jednog elementa liste, na način da bude moguće deklarirati varijablu **stanica x**.
- Napišite funkciju **rednibr()** koja prima pokazivač na početak vezane liste **prvi** i naziv stanice **x**, te vraća redni broj stanice u pripadnoj liniji (polazna stanica ima redni broj 1). Ukoliko ne postoji stanica naziva **x**, funkcija treba vratiti 0.
- Napišite funkciju **natrag()** koja, po adresi, prima pokazivač na početak vezane liste i invertira vezanu listu tako da odgovara povratnoj liniji (ako je ulazna vezana lista predstavljala liniju $A \rightarrow B$, funkcija je treba preoblikovati da predstavlja liniju $B \rightarrow A$). Funkcija ne vraća ništa kao funkcijsku vrijednost. Pazite da napravite i odgovarajuću promjenu u varijablama **vrijeme**.
- Napišite funkciju prototipa

```
stanica* put(stanica** prvi, const char* x, const char* y, int* t)
```

koja (ako je potrebno) invertira vezanu listu tako da stanica **x** bude manjeg rednog broja od stanice **y**, vraća pokazivač na stanicu **x** i preko varijabilnog parametra **t** vraća vrijeme potrebno autobusu da iz stanice **x** dođe u stanicu **y**. Ukoliko jedna od stanica ne postoji u navedenoj liniji, funkcija vraća NULL pokazivač i vrijeme postavlja na -1 .

Dozvoljeno je korištenje funkcija iz biblioteke **string.h**. Podzadatak a) je nužan za rješavanje preostalih podzadataka, dok funkcije iz podzadataka b) i c) možete koristiti u podzadatku d) i ako ih niste napisali.

Zadatak 3 (3+6+6 = 15 bodova) Recenzija glazbenog albuma sastoji se od četiri podatka: imena izvođača, imena albuma, numeričke ocjene (prirodni broj od 1 do 10) i kratkog teksta u kojem je dan osvrt na album.

- a) Definirajte tip podatka `album` koji će služiti za čuvanje recenzije jednog albuma. Duljina recenzije pojedinog albuma nije unaprijed ograničena, dok za duljine imena albuma i izvođača možete postaviti "globalna" ograničenja (uočite da imena mogu sadržavati razmake).
- b) Napišite funkciju `void novi(album *x)` koja će omogućiti korisniku da s tipkovnice učitava recenziju novog albuma te spremi odgovarajući unos u varijablu tipa `album` koju smo funkciji prenijeli po adresi. Nove albume unosimo u sljedećem formatu:

```
Naziv albuma: <naslov>
Izvođač: <ime izvođača>
Ocjena: k
Recenzija: <tekst recenzije>
```

Znakovi `< i >` nisu dio unosa, primjerice: `Naziv albuma: Funeral`

- c) Napišite funkciju `void recenziraj(album *popis, int n, char* naslov)` koja prima popis recenzija (ostvaren kao niz podataka tipa `album`, duljine n) i naziv nekog albuma. Funkcija treba ispisati recenziju albuma s odgovarajućim nazivom (u formatu opisanom u b) zadatku) te naslove svih albuma istog izvođača. U slučaju da album nije pronađen, potrebno je ispisati prikladnu poruku. Možete pretpostaviti da je niz albuma sortirani (leksikografski uzlazno) prema nazivu albuma.

Za dodatnih 5 bodova (ne ulaze u 80%, tj. 80% = 12 bodova) pronađite traženi album u logaritamskoj složenosti.

Zadatak 4 (20 bodova) Trajektna luka koristi tekstualnu datoteku u kojoj vodi evidenciju o prometu na sljedeći način. U svakoj liniji zapisana su tri podatka, odvojena razmacima. Prvi je oznaka broda, koja je niz od najviše 10 znakova. Zatim, vrijeme polaska, u formatu hh:mm, te je na kraju zapisano odredište, koje je niz od najviše 30 znakova i može sadržavati razmake. Primjer jedne linije je

CX2180 15:30 RIJEKA

Napišite program koji uzme, kao argumente komandne linije, imena datoteke evidencije i izlazne datoteke. Program treba prepisati linije iz datoteke evidencije u izlaznu datoteku, tako da ih sortira leksikografski silazno obzirom na oznaku broda. U ovom zadatku dozvoljeno je korištenje dodatnih nizova, samo pazite na rezervaciju memorije za njih.