

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Rezultati i uvidi:** Rezultati u nedjelju, 4.7.2021. navečer na webu. Informacije o uvidima bit će objavljene naknadno.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele isključite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno predajte sve papire sa zadacima, čak i ako neke zadatke niste rješavali. 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. Dozvoljeno je pisanje pomoćnih funkcija.

Uvjete navedene u zadacima (pozitivnost, ograde na  $n$  i sl.) ne treba provjeravati.

**Zadatak 1** (10 bodova) Kvadratna matrica cijelih brojeva reda  $n \in \mathbb{N}$  je zapisana u binarnoj datoteci `ulaz.dat` tako da je prvo zapisan red matrice, a zatim svi elementi redom po redcima. Napišite program koji mijenja binarnu datoteku tako da ona nakon izvršavanja programa sadrži transponiranu matricu. Morate napisati cijeli program uključujući otvaranje/zatvaranje datoteke.

**NAPOMENA:** Rješenje koje koristi polja nosi 0 bodova.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 2** (5+10+5 bodova) Jadrolinija svake godine vodi statistiku o broju automobila hrvatskih registarskih oznaka koje su prevezli sa svojih gatova na ljetovanje. Svaka prodana karta vozila se sprema u strukturu **Karta** koja ima jedinstveni *id*, *ime gata s kojega je vozilo prevezeno*, *ime gata odredišta* i *registarsku oznaku prevezenog automobila*. Možete pretpostaviti da su imena gatova stringovi od maksimalno 50 znakova, dok je registarska oznaka string oblika: **XX 123(4)-XX**, gdje **X** označava mjesto za proizvoljno veliko slovo hrvatske abecede uz zamjene  $\check{C} \rightarrow C$ ,  $\check{C} \rightarrow C$ ,  $\check{S} \rightarrow S$ ,  $\check{D} \rightarrow DJ$ ,  $\check{Z} \rightarrow Z$ , broj u zagradi je opcionalan, dakle registracija može sadržavati 3 ili 4 znamenke nakon 2 velika slova, zatim slijedi crtica i još dva velika slova. Prva dva velika slova označavaju ime grada i ona ne mogu biti proizvoljna (iznimno ćemo Šibenik - ŠI označiti kao SV). Sve važeće oznake gradova u Hrvatskoj su upisane u binarnu datoteku *oznakeGradova.dat* (sortirano uzlazno po leksikografskom uređaju oznaka gradova). Navedena datoteka je nastala spremanjem niza struktura **Oznaka** koje sadrže jedinstveni *id* oznake (nepovezan s id-em karte), *string oznake grada* (dva znaka) i *string punog imena grada* (string od maksimalno 20 znakova). Statistike dolazaka se spremaju u strukturu **Dolasci** u obliku *ukupnog broja računa izdanih vozilima* s registracijama iz zadanog grada. Pretpostavite da je polje od 34 strukture **Dolasci** globalno te da id oznake grada odgovara indeksu u globalnom polju u kojem se sprema statistika za zadani grad. U svim strukturama je *id* nenegativan cijeli broj. Pretpostavite da su potrebne datoteke već otvorene za čitanje.

- Napišite program koji čita redom jednu po jednu kartu iz binarne datoteke **racuni.dat** i ažurira globalno polje od 34 strukture tipa **Dolasci** (pronađe odgovarajući grad binarnim traženjem i ažurira odgovarajuću strukturu **Dolasci**). Pri učitavanju karte treba obavezno provjeriti da registarska oznaka odgovara oznaci vozila Republike Hrvatske (ovdje zanemarujemo sve individualizirane registarske oznake). Provjera mora utvrditi i je li oznaka grada ispravna, odnosno nalazi li se u sortiranom nizu struktura tipa **Oznaka**. Pretragu morate vršiti binarnim pretraživanjem, **rješenje koje koristi obično pretraživanje donosi maksimalno 5 bodova**). Osmislite strukture i definirajte funkcije tako da ne dolazi do nepotrebnog korištenja memorije i nepotrebnog poziva funkcija traženja.
- Napišite funkciju `void ispis()` koja sortirano u silaznom poretку ispisuje oznaku grada, puno ime grada i broj prevezenih automobila s registarskom oznakom toga grada. Funkcija ne smije mijenjati poredak elemenata u globalnom polju dolazaka niti koristiti pomoćna polja. **NAPOMENA:** Rješenje koje ne poštuje zadane uvjete donosi 0 bodova.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 3** (1+4+15 bodova) Podatci o prodanim kartama za trajekt na gatu Sv. Petra spremaju se u vezanu listu. Za svaku prodanu kartu pamtimo broj putnika (prirodni broj), registracijsku oznaku vozila (string duljine najviše 10) i volumni prostor vozila (nenegativni cijeli broj). Vozila s prioritetnim ukrcajem imaju negativnu vrijednost volumnog prostora; apsolutna vrijednost tog broja je njihov stvarni volumni prostor. Svaka nova prodana karta dolazi na početak liste.

- Napišite deklaraciju odgovarajućeg tipa podatka za prodanu kartu, na način da bude moguće definirati varijablu naredbom **karta a**;. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- Napišite funkciju **karta\* unesi(karta\* head, karta\* nova)** koja prima pokazivače **head** i **nova** gdje je **head** pokazivač na početak liste dok je **nova** pokazivač na novu kartu. Funkcija ubacuje novu kartu na početak liste i vraća pokazivač na prvi element tako dobivene vezane liste. Početna vezana lista može biti i prazna.
- Napišite funkciju **karta\* ukrcaj(karta\* head, int kapacitet\_trajekta)** koja briše karte ukrcanih vozila. Ukrcaj se obavlja na sljedeći način. Prvo se ukrcavaju prioriteta vozila po redu prodaje karata (vozilo ranije kupljene karte bit će, ako je moguće, prije ukrcano). Njihovo ukrcavanje završava kada su sva ukrcana ili kada se naiđe na prvu kartu prioriteta vozila koje ne može biti ukrcano (ukupni volumni prostor ukrcanih vozila mora biti manji ili jednak kapacitetu trajekta). Nakon toga se, na isti način, ukrcavaju neprioritetna vozila. Funkcija vraća pokazivač na početak vezane liste karata, a u slučaju da su svi elementi liste izbrisani, vraća NULL. Poredak karata u listi je po redu prodaje tako da se na početku liste nalazi posljednja prodana karta.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 4** (20 bodova) Napišite funkciju `int rijeci_desno(FILE *in, char ****niz)` koja prima pokazivač na tekstnu datoteku proizvoljne veličine (datoteka je već otvorena za čitanje) te preko varijabilnog parametra vraća dvodimenzionalni niz stringova.

Niz mora imati jednak broj redaka i stupaca, i to najmanji dovoljan da u njega stanu sve riječi (stringovi koji ne sadrže bjeline, tj. znakove razmaka, tabulatora i novog reda) iz datoteke; pretpostavite da svaka riječ ima najviše 20 znakova. Početak niza sadrži prazne stringove, a ostatak niza riječi iz datoteke (praznih stringova na početku niza ima točno onoliko koliko treba da se niz popuni).

Riječi moraju u nizu biti poredane leksikografski (pojedini znakovi se uspoređuju prema svojoj ASCII vrijednosti). Povratna vrijednost funkcije je broj redaka u nizu.

Primjer sadržaja datoteke:

`Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.`

Odgovarajući niz (prazni stringovi su označeni sa X):

|             |       |            |         |         |
|-------------|-------|------------|---------|---------|
| X           | X     | X          | X       | X       |
| X           | Lorem | adipiscing | aliqua. | amet,   |
| consectetur | do    | dolor      | dolore  | eiusmod |
| elit,       | et    | incidunt   | ipsum   | labore  |
| magna       | sed   | sit        | tempor  | ut      |

Napomene: Pri rješavanju ovog zadatka smijete koristiti nizove. Također, smijete koristiti **isključivo** zaglavlja `stdio.h`, `stdlib.h` te `string.h`.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Rezultati i uvidi:** Rezultati u nedjelju, 4.7.2021. navečer na webu. Informacije o uvidima bit će objavljene naknadno.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele isključite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno predajte sve papire sa zadacima, čak i ako neke zadatke niste rješavali. 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. Dozvoljeno je pisanje pomoćnih funkcija.

Uvjete navedene u zadacima (pozitivnost, ograde na  $n$  i sl.) ne treba provjeravati.

**Zadatak 1** (10 bodova) Kvadratna matrica cijelih brojeva reda  $n \in \mathbb{N}$  je zapisana u binarnoj datoteci `ulaz.dat` tako da je prvo zapisan red matrice, a zatim svi elementi redom po redcima. Napišite program koji mijenja binarnu datoteku tako da ona nakon poziva programa sadrži matricu s redcima u obrnutom poretku (prvi redak postaje zadnji, zadnji postaje prvi, drugi postaje predzadnji itd). Morate napisati cijeli program uključujući otvaranje/zatvaranje datoteke.

**NAPOMENA:** Rješenje koje koristi polja nosi 0 bodova.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 2** (5+10+5 bodova) Jadrolinija svake godine vodi statistiku o broju automobila hrvatskih registarskih oznaka koje su prevezli sa svojih gatova na ljetovanje. Svaka prodana karta vozila se sprema u strukturu **Karta** koja ima jedinstveni *id*, *ime gata s kojega je vozilo prevezeno*, *ime gata odredišta* i *registarsku oznaku prevezenog automobila*. Možete pretpostaviti da su imena gatova stringovi od maksimalno 50 znakova, dok je registarska oznaka string oblika: **XX 123(4)-XX**, gdje **X** označava mjesto za proizvoljno veliko slovo hrvatske abecede uz zamjene  $\dot{C} \rightarrow C$ ,  $\dot{C} \rightarrow \check{C}$ ,  $\dot{S} \rightarrow S$ ,  $\dot{D} \rightarrow DJ$ ,  $\dot{Z} \rightarrow Z$ , broj u zagradi je opcionalan, dakle registracija može sadržavati 3 ili 4 znamenke nakon 2 velika slova, zatim slijedi crtica i još dva velika slova. Prva dva velika slova označavaju ime grada i ona ne mogu biti proizvoljna (iznimno ćemo Šibenik - ŠI označiti kao SV). Sve važeće oznake gradova u Hrvatskoj su upisane u binarnu datoteku *oznakeGradova.dat* (sortirano silazno po leksikografskom uređaju oznaka gradova). Navedena datoteka je nastala spremanjem niza struktura **Oznaka** koje sadrže jedinstveni *id* oznake (nepovezan s id-em karte), *string oznake grada* (dva znaka) i *string punog imena grada* (string od maksimalno 20 znakova). Statistike dolazaka se spremaju u strukturu **Dolasci** u obliku *ukupnog broja računa izdanih vozilima* s registracijama iz zadanog grada. Pretpostavite da je polje od 34 strukture **Dolasci** globalno te da id oznake grada odgovara indeksu u globalnom polju u kojem se sprema statistika za zadani grad. U svim strukturama je *id* nenegativan cijeli broj. Pretpostavite da su potrebne datoteke već otvorene za čitanje.

- Napišite program koji čita redom jednu po jednu kartu iz binarne datoteke **racuni.dat** i ažurira globalno polje od 34 strukture tipa **Dolasci** (pronađe odgovarajući grad binarnim traženjem i ažurira odgovarajuću strukturu **Dolasci**). Pri učitavanju karte treba obavezno provjeriti da registarska oznaka odgovara oznaci vozila Republike Hrvatske (ovdje zanemarujemo sve individualizirane registarske oznake). Provjera mora utvrditi i je li oznaka grada ispravna, odnosno nalazi li se u sortiranom nizu struktura tipa **Oznaka**. Pretragu morate vršiti binarnim pretraživanjem, **rješenje koje koristi obično pretraživanje donosi maksimalno 5 bodova**). Osmislite strukture i definirajte funkcije tako da ne dolazi do nepotrebnog korištenja memorije i nepotrebnog poziva funkcija traženja.
- Napišite funkciju `void ispis()` koja sortirano u uzlaznom poretku ispisuje oznaku grada, puno ime grada i broj prevezenih automobila s registarskom oznakom toga grada. Funkcija ne smije mijenjati poredak elemenata u globalnom polju dolazaka niti koristiti pomoćna polja. **NAPOMENA:** Rješenje koje ne poštuje zadane uvjete donosi 0 bodova.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 3** (1+4+15 bodova) Podatci o vozilima koja čekaju ukrcaj na trajekt na gatu Sv. Duje spremaju se u vezanu listu. Vozilo ima registracijsku oznaku (string duljine najviše 10 znakova) i volumni prostor vozila (cijeli broj). Vozila s prioriternim ukrcajem imaju negativnu vrijednost volumnog prostora; apsolutna vrijednost tog broja je njihov stvarni volumni prostor. Svako novopridošlo vozilo dolazi na početak liste.

- Napišite deklaraciju odgovarajućeg tipa podatka za vozilo, na način da bude moguće definirati varijablu naredbom `vozilo a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- Napišite funkciju `vozilo* unosi(vozilo* head, vozilo* novo)` koja prima pokazivače `head` i `novo` gdje je `head` pokazivač na početak vezane liste dok je `novo` pokazivač na novo vozilo koje je došlo na gat. Funkcija ubacuje novo vozilo na početak liste i vraća pokazivač na prvi element tako dobivene vezane liste. Početna vezana lista može biti i prazna.
- Napišite funkciju `vozilo* ukrcaj(vozilo* head, int kapacitet_trajekta)` koja briše ona vozila iz liste koja su ukrcana u trajekt. Ukrcaj se obavlja na sljedeći način. Prvo se ukrcavaju prioriterna vozila po redu dolaska na gat (vozilo došlo ranije bit će, ako je moguće, prije ukrcano). Njihovo ukrcavanje završava kada su sva ukrcana ili kada se naiđe na prvo prioriterno vozilo koje ne može biti ukrcano (ukupni volumni prostor ukrcanih vozila mora biti manji ili jednak kapacitetu trajekta). Nakon toga se, na isti način, ukrcavaju neprioritetna vozila. Funkcija vraća pokazivač na početak vezane liste vozila, a u slučaju da su svi elementi liste izbrisani, vraća `NULL`. Poredak vozila u listi je po redu dolaska tako da se na početku nalazi vozilo koje je posljednje došlo.

## Programiranje 2 – drugi kolokvij, 25. 6. 2021.

**Zadatak 4** (20 bodova) Napišite funkciju `int rijeci_lijevo(FILE *in, char ****niz)` koja prima pokazivač na tekstnu datoteku proizvoljne veličine (datoteka je već otvorena za čitanje) te preko varijabilnog parametra vraća dvodimenzionalni niz stringova.

Niz mora imati jednak broj redaka i stupaca, i to najmanji dovoljan da u njega stanu sve riječi (stringovi koji ne sadrže bjeline, tj. znakove razmaka, tabulatora i novog reda) iz datoteke; pretpostavite da svaka riječ ima najviše 20 znakova. Početak niza sadrži riječi iz datoteka, a ostatak niza sadrži prazne stringove (ima ih točno onoliko koliko treba da se niz popuni do kraja).

Riječi moraju u nizu biti poredane obrnutim leksikografskim redoslijedom (pojedini znakovi se uspoređuju prema svojoj ASCII vrijednosti). Povratna vrijednost funkcije je broj redaka u nizu.

Primjer sadržaja datoteke:

`Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.`

Odgovarajući niz (prazni stringovi su označeni sa X):

|                      |                      |                         |                    |                          |
|----------------------|----------------------|-------------------------|--------------------|--------------------------|
| <code>ut</code>      | <code>tempor</code>  | <code>sit</code>        | <code>sed</code>   | <code>magna</code>       |
| <code>labore</code>  | <code>ipsum</code>   | <code>incidunt</code>   | <code>et</code>    | <code>elit,</code>       |
| <code>eiusmod</code> | <code>dolore</code>  | <code>dolor</code>      | <code>do</code>    | <code>consectetur</code> |
| <code>amet,</code>   | <code>aliqua.</code> | <code>adipiscing</code> | <code>Lorem</code> | <code>X</code>           |
| <code>X</code>       | <code>X</code>       | <code>X</code>          | <code>X</code>     | <code>X</code>           |

Napomene: Pri rješavanju ovog zadatka smijete koristiti nizove. Također, smijete koristiti **isključivo** zaglavlja `stdio.h`, `stdlib.h` te `string.h`.