

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 30.6., navečer na webu, a uvidi u ponedjeljak, 1.7., u 11 sati.

**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.

**Zadatak 1.** (15 bodova) Podatke o biciklistima u svjetskom kupu spremamo u obliku strukture koja sadrži: ime i prezime (dva stringa od po 30 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i polje od 21 broja (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `biciklist`.

Napišite funkciju koja učitava podatke za jednog biciklista iz već otvorene tekst datoteke, zadane kao argument funkcije. Čitanje počinje od trenutne pozicije u toj datoteci. Funkcija treba vratiti pokazivač na dinamički alociranu strukturu s učitanim podacima.

Podaci o jednom biciklistu zauzimaju tri reda teksta. 1. red: ime i prezime (odvojeni zarezom i jednom prazninom), s tim da ime i prezime smiju sadržavati praznine; 2. red: kratica ekipe, praznina i broj; 3. red: niz od  $m \leq 21$  blokova znakova, s rezultatima u prvih  $m$  utrka. Svaki blok je ili prazan (nije sudjelovao) ili sadrži rezultat u obliku `h:mm:ss,nn` (vrijeme završetka utrke — sati, minute, sekunde, a `nn` su stotinke sekunde). Između svaka dva bloka nalazi se niz praznina (nula ili više), jedan znak `'|'` i niz praznina (nula ili više). Zadnji blok završava krajem reda. Treba učitati i taj znak za kraj reda. Broj blokova  $m$  nije unaprijed zadan, već ga treba odrediti iz oblika podataka u tom redu.

Pretpostavljamo da su svi podaci napisani u korektnom formatu. Kod spremanja vremena, podatak treba pretvoriti u sekunde. Za prazne blokove i buduće utrke (one za kojih nema pripadnih blokova) treba spremiti broj `-1.0`.

Primjer podataka za jednog biciklista, s rezultatima (ovdje je  $m = 4$ ):

Miguel Angel, Lopez Moreno

AST 141

2:13:15,00 | |0:28:37,45 |

**Napomena:** Zabranjeno je alociranje i korištenje bilo kakvih dodatnih polja. Takvo rješenje donosi 0 bodova!

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 2.** (20 bodova) Podatke o biciklistima u svjetskom kupu spremamo u vezanoj listi. Za svakog biciklista pamtimo: ime i prezime (dva stringa od po 30 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i polje od 21 broja (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `bic` za jednog biciklista, koji sadrži samo podatke nužne za spremanje takve liste u memoriji.

- (a) Napišite funkciju `bic* izbaci(bic* first, int n, bic** spori)`, gdje je `first` pokazivač na početak vezane liste biciklista, a `n` je indeks utrke. Funkcija treba iz liste izbaciti sve one bicikliste koji u  $n$ -toj utrci imaju vrijeme za barem 20% veće od vremena pobjednika te utrke (onog s najmanjim vremenom). Izbačene bicikliste treba preseliti u zasebnu listu (poredak elemenata nije bitan), a njezin početak treba vratiti kroz varijabilni argument `spori`.
- (b) Napišite funkciju `bic* poredaj(bic* first, int n)`, gdje je `first` pokazivač na početak vezane liste biciklista, a `n` je indeks utrke. Zadanu listu treba sortirati uzlazno, po vremenu u  $n$ -toj utrci (pobjednik treba doći na početak). Ako više biciklista ima isto vrijeme, njihov poredak smije biti bilo koji (nije bitan). Za sortiranje **obavezno** iskoristite mergesort algoritam!

Smijete pretpostaviti da je indeks  $n$  korektno zadan ( $0 \leq n < 21$ ) i da su vremena u svim utrkama pozitivna.

**Napomena:** Nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 3.** (20 bodova) Podatke o biciklistima u svjetskom kupu spremamo u obliku strukture koja sadrži: ime i prezime (dva stringa od po 30 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj). Definirajte odgovarajući tip podataka `bic_bod`.

Na početku sezone, prije prve utrke, svaki biciklist ima nula bodova. U svakoj utrci, samo prvih 20 biciklista dobiva bodove. Broj bodova zadan je globalnim cjelobrojnim poljem `bod` s 20 elemenata, tako da  $i$ -to mjesto u utrci donosi `bod[i - 1]` bodova (primjerice, `bod[0] = 40`, `bod[1] = 35`, ..., `bod[19] = 1`). Bodovi iz raznih utrka se zbrajaju.

- (a) Napišite funkciju `popis` koja prima prirodni broj  $n$  i vraća pokazivač na prvi element (dinamički kreiranog) polja svih biciklista koji su osvojili bar jedan bod u prvih  $n$  utrka. Broj biciklista u tom polju treba vratiti kroz varijabilni argument. Poredak  $i$ -te utrke ( $i \geq 1$ ) spremljen je u tekst-datoteci s imenom `z3a_i.txt`, gdje  $i$  označava tekstualni zapis broja  $i$  (poput `z3a_7.txt` ili `z3a_12.txt`). Svaki red u takvoj datoteci sadrži natjecateljski broj biciklista (prirodan broj) i pripadno vrijeme u utrci (`double`, u sekundama). U svakoj datoteci, zapisi su poredani prema poretku u toj utrci — od prvog mjesta (pobjednika) do zadnjeg. Dozvoljeno je da utrka ima i manje od 20 biciklista.
- (b) Napišite funkciju `rang_lista` koja prima polje biciklista i broj biciklista u polju. Funkcija treba zadano polje sortirati **silazno** po trenutnom broju bodova, koristeći poziv funkcije `qsort` iz standardne biblioteke `stdlib.h`. Ako više biciklista ima isti broj bodova, njihov poredak smije biti bilo koji. Za poziv funkcije `qsort`, prvo trebate napisati funkciju za usporedbu dva biciklista po trenutnom broju bodova. Pazite na rezultat usporedbe, jer `qsort` sortira uzlazno, a polje treba sortirati silazno!

**Napomene:** Dozvoljeno je koristiti jedan string za ime datoteke (najviše 20 znakova) i jedno dinamički alocirano polje za bicikliste koji nemaju nula bodova (treba ga realocirati po potrebi). Ne treba provjeravati otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 4.** (15 bodova) Podatke o biciklistima u svjetskom kupu spremamo u obliku strukture tipa `bic_bod` (isti tip kao u 3. zadatku), koja sadrži: ime i prezime (dva stringa od po 30 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj).

Napišite program koji s komandne linije učitava dva argumenta. Prvi je ime postojeće (ulazne) binarne datoteke, a drugi je ime nove (izlazne) tekst-datoteke. Binarna datoteka sadrži niz podataka o biciklistima, sortiran silazno po trenutnom broju bodova (trenutni bodovi smiju biti i nula). Prema tom poretku, treba kreirati startnu listu za sljedeću utrku.

Biciklisti su podijeljeni u grupe od po 15, redom prema poretku, s tim da zadnja grupa može biti i manja. Vodeća grupa (prvih 15, ako ih ima toliko) starta kao zadnja, a unutar te grupe, biciklisti startaju u obrnutom redoslijedu od poretka, tj. vodeći u poretku starta kao zadnji. Sve ostale grupe (od druge do zadnje, ako ih ima) startaju redom po grupama, a unutar svake grupe, biciklisti startaju slučajnim redom. Za generiranje slučajnog startnog poretka unutar jedne takve grupe iskoristite funkciju `rand` iz standardne biblioteke `stdlib.h`.

U izlaznu datoteku treba napisati podatke o biciklistima, redom kojim trebaju startati, po jedan biciklist u svakom redu. Poredak i format podataka u redu je: redni broj starta (počevši od 1), natjecateljski broj, kratica ekipe, prezime, ime.

**Napomene:** Dozvoljeno je koristiti jedno polje cijelih brojeva i dva polja biciklista, svako polje od po 15 elemenata. Ne trebate provjeravati argumente komandne linije i otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 30.6., navečer na webu, a uvidi u ponedjeljak, 1.7., u 11 sati.

**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.

**Zadatak 1.** (15 bodova) Podatke o biatloncima u svjetskom kupu spremamo u obliku strukture koja sadrži: prezime i ime (dva stringa od po 25 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i polje od 16 brojeva (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `biatlonac`.

Napišite funkciju koja učitava podatke za jednog biatlonca iz već otvorene tekst datoteke, zadane kao argument funkcije. Čitanje počinje od trenutne pozicije u toj datoteci. Funkcija treba vratiti pokazivač na dinamički alociranu strukturu s učitanim podacima.

Podaci o jednom biatloncu zauzimaju tri reda teksta. 1. red: prezime i ime (odvojeni zarezom i jednom prazninom), s tim da prezime i ime smiju sadržavati praznine; 2. red: kratica države, praznina i broj; 3. red: niz od  $m \leq 16$  blokova znakova, s rezultatima u prvih  $m$  utrka. Svaki blok je ili prazan (nije sudjelovao) ili sadrži rezultat u obliku `mm:ss,n` (vrijeme završetka utrke — minute, sekunde, a `n` su desetinke sekunde). Između svaka dva bloka nalazi se niz praznina (nula ili više), jedan znak `'/'` i niz praznina (nula ili više). Zadnji blok završava krajem reda. Treba učitati i taj znak za kraj reda. Broj blokova  $m$  nije unaprijed zadan, već ga treba odrediti iz oblika podataka u tom redu.

Pretpostavljamo da su svi podaci napisani u korektnom formatu. Kod spremanja vremena, podatak treba pretvoriti u sekunde. Za prazne blokove i buduće utrke (one za kojih nema pripadnih blokova) treba spremiti broj `-1.0`.

Primjer podataka za jednog biatlonca, s rezultatima (ovdje je  $m = 4$ ):

```
Boe, Johannes Thingnes
NOR 21
47:15,5 / / 18:37,4/
```

**Napomena:** Zabranjeno je alociranje i korištenje bilo kakvih dodatnih polja. Takvo rješenje donosi 0 bodova!

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 2.** (20 bodova) Podatke o biatloncima u svjetskom kupu spremamo u vezanoj listi. Za svakog biatlonca pamtimo: prezime i ime (dva stringa od po 25 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i polje od 16 brojeva (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `bia` za jednog biatlonca, koji sadrži samo podatke nužne za spremanje takve liste u memoriji.

- (a) Napišite funkciju `bia* izbaci(bia* first, int n, bia** spori)`, gdje je `first` pokazivač na početak vezane liste biatlonaca, a `n` je indeks utrke. Funkcija treba iz liste izbaciti sve one biatlonce koji u  $n$ -toj utrci imaju vrijeme za barem 15% veće od prosječnog vremena svih u toj utrci. Izbačene biatlonce treba preseliti u zasebnu listu (poredak elemenata nije bitan), a njezin početak treba vratiti kroz varijabilni argument `spori`.
- (b) Napišite funkciju `bia* poredaj(bia* first, int n)`, gdje je `first` pokazivač na početak vezane liste biatlonaca, a `n` je indeks utrke. Zadanu listu treba sortirati uzlazno, po vremenu u  $n$ -toj utrci (pobjednik treba doći na početak). Ako više biatlonaca ima isto vrijeme, njihov poredak smije biti bilo koji (nije bitan). Za sortiranje **obavezno** iskoristite mergesort algoritam!

Smijete pretpostaviti da je indeks  $n$  korektno zadan ( $0 \leq n < 16$ ) i da su vremena u svim utrkama pozitivna.

**Napomena:** Nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 3.** (20 bodova) Podatke o biatloncima u svjetskom kupu spremamo u obliku strukture koja sadrži: prezime i ime (dva stringa od po 25 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj). Definirajte odgovarajući tip podataka `bia_bod`.

Na početku sezone, prije prve utrke, svaki biatlonac ima nula bodova. U svakoj utrci, samo prvih 40 biatlonaca dobiva bodove. Broj bodova zadan je globalnim cjelobrojnim poljem `bod` s 40 elemenata, tako da  $i$ -to mjesto u utrci donosi `bod[i - 1]` bodova (primjerice, `bod[0] = 60`, `bod[1] = 55`, ..., `bod[39] = 1`). Bodovi iz raznih utrka se zbrajaju.

- (a) Napišite funkciju `popis` koja prima prirodni broj  $n$  i vraća pokazivač na prvi element (dinamički kreiranog) polja svih biatlonaca koji su osvojili bar jedan bod u prvih  $n$  utrka. Broj biatlonaca u tom polju treba vratiti kroz varijabilni argument. Poredak  $i$ -te utrke ( $i \geq 1$ ) spremljen je u tekst-datoteci s imenom `z3b_i.txt`, gdje  $i$  označava tekstualni zapis broja  $i$  (poput `z3b_7.txt` ili `z3b_12.txt`). Svaki red u takvoj datoteci sadrži natjecateljski broj biatlonca (prirodan broj) i pripadno vrijeme u utrci (`double`, u sekundama). U svakoj datoteci, zapisi su poredani prema poretku u toj utrci — od prvog mjesta (pobjednika) do zadnjeg. Dozvoljeno je da utrka ima i manje od 40 biatlonaca.
- (b) Napišite funkciju `rang_lista` koja prima polje biatlonaca i broj biatlonaca u polju. Funkcija treba zadano polje sortirati **silazno** po trenutnom broju bodova, koristeći poziv funkcije `qsort` iz standardne biblioteke `stdlib.h`. Ako više biatlonaca ima isti broj bodova, njihov poredak smije biti bilo koji. Za poziv funkcije `qsort`, prvo trebate napisati funkciju za usporedbu dva biatlonca po trenutnom broju bodova. Pazite na rezultat usporedbe, jer `qsort` sortira uzlazno, a polje treba sortirati silazno!

**Napomene:** Dozvoljeno je koristiti jedan string za ime datoteke (najviše 20 znakova) i jedno dinamički alocirano polje za biatlonce koji nemaju nula bodova (treba ga realocirati po potrebi). Ne treba provjeravati otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 4.** (15 bodova) Podatke o biatloncima u svjetskom kupu spremamo u obliku strukture tipa `bia_bod` (isti tip kao u 3. zadatku), koja sadrži: prezime i ime (dva stringa od po 25 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj).

Napišite program koji s komandne linije učitava dva argumenta. Prvi je ime postojeće (ulazne) binarne datoteke, a drugi je ime nove (izlazne) tekst-datoteke. Binarna datoteka sadrži niz podataka o biatloncima, sortiran silazno po trenutnom broju bodova (trenutni bodovi smiju biti i nula). Prema tom poretku, treba kreirati startnu listu za sljedeću utrku.

Biatlonci su podijeljeni u grupe od po 20, redom prema poretku, s tim da zadnja grupa može biti i manja. Vodeća grupa (prvih 20, ako ih ima toliko) starta kao prva, a unutar te grupe, biatlonci startaju u obrnutom redosljedju od poretka, tj. vodeći u poretku starta kao zadnji u grupi. Sve ostale grupe (od druge do zadnje, ako ih ima) startaju redom po grupama, a unutar svake grupe, biatlonci startaju slučajnim redom. Za generiranje slučajnog startnog poretka unutar jedne takve grupe iskoristite funkciju `rand` iz standardne biblioteke `stdlib.h`.

U izlaznu datoteku treba napisati podatke o biatloncima, redom kojim trebaju startati, po jedan biatlonac u svakom redu. Poredak i format podataka u redu je: redni broj starta (počevši od 1), natjecateljski broj, kratica države, prezime, ime.

**Napomene:** Dozvoljeno je koristiti jedno polje cijelih brojeva i jedno polje biatlonaca, svako polje od po 20 elemenata. Ne trebate provjeravati argumente komandne linije i otvaranje datoteka.



## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 30.6., navečer na webu, a uvidi u ponedjeljak, 1.7., u 11 sati.

**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.

**Zadatak 1.** (15 bodova) Podatke o vozačima automobila u svjetskom kupu spremamo u obliku strukture koja sadrži: prezime i ime (dva stringa od po 40 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i polje od 18 brojeva (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `vozac`.

Napišite funkciju koja učitava podatke za jednog vozača iz već otvorene tekst datoteke, zadane kao argument funkcije. Čitanje počinje od trenutne pozicije u toj datoteci. Funkcija treba vratiti pokazivač na dinamički alociranu strukturu sa učitanim podacima.

Podaci o jednom vozaču zauzimaju tri reda teksta. 1. red: prezime i ime (odvojeni zarezom i jednom prazninom), s tim da prezime i ime smiju sadržavati praznine; 2. red: kratica ekipe, praznina i broj; 3. red: niz od  $m \leq 18$  blokova znakova, s rezultatima u prvih  $m$  utrka. Svaki blok je ili prazan (nije sudjelovao) ili sadrži rezultat u obliku `h:mm:ss,nnn` (vrijeme završetka utrke — sati, minute, sekunde, a `nnn` su tisućinke sekunde). Između svaka dva bloka nalazi se niz praznina (nula ili više), jedan znak '\$' i niz praznina (nula ili više). Zadnji blok završava krajem reda. Treba učitati i taj znak za kraj reda. Broj blokova  $m$  nije unaprijed zadan, već ga treba odrediti iz oblika podataka u tom redu.

Pretpostavljamo da su svi podaci napisani u korektnom formatu. Kod spremanja vremena, podatak treba pretvoriti u sekunde. Za prazne blokove i buduće utrke (one za kojih nema pripadnih blokova) treba spremati broj `-1.0`.

Primjer podataka za jednog vozača, s rezultatima (ovdje je  $m = 4$ ):

Lopez Pereira, Oscar Carlos

TOY 11

2:13:15,327 \$ \$0:28:37,045 \$

**Napomena:** Zabranjeno je alociranje i korištenje bilo kakvih dodatnih polja. Takvo rješenje donosi 0 bodova!

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 2.** (20 bodova) Podatke o vozačima automobila u svjetskom kupu spremamo u vezanoj listi. Za svakog vozača pamtimo: prezime i ime (dva stringa od po 40 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i polje od 18 brojeva (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `voz` za jednog vozača, koji sadrži samo podatke nužne za spremanje takve liste u memoriji.

- (a) Napišite funkciju `voz* izbaci(voz* first, int n, voz** spori)`, gdje je `first` pokazivač na početak vezane liste vozača, a `n` je indeks utrke. Funkcija treba iz liste izbaciti sve one vozače koji u  $n$ -toj utrci imaju vrijeme strogo veće od 90% vremena zadnjeg u toj utrci (onog s najvećim vremenom). Izbačene vozače treba preseliti u zasebnu listu (poredak elemenata nije bitan), a njezin početak treba vratiti kroz varijabilni argument `spori`.
- (b) Napišite funkciju `voz* poredaj(voz* first, int n)`, gdje je `first` pokazivač na početak vezane liste vozača, a `n` je indeks utrke. Zadanu listu treba sortirati uzlazno, po vremenu u  $n$ -toj utrci (pobjednik treba doći na početak). Ako više vozača ima isto vrijeme, njihov poredak smije biti bilo koji (nije bitan). Za sortiranje **obavezno** iskoristite mergesort algoritam!

Smijete pretpostaviti da je indeks  $n$  korektno zadan ( $0 \leq n < 18$ ) i da su vremena u svim utrkama pozitivna.

**Napomena:** Nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 3.** (20 bodova) Podatke o vozačima automobila u svjetskom kupu spremamo u obliku strukture koja sadrži: prezime i ime (dva stringa od po 40 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj). Definirajte odgovarajući tip podataka `voz_bod`.

Na početku sezone, prije prve utrke, svaki vozač ima nula bodova. U svakoj utrci, samo prvih 15 vozača dobiva bodove. Broj bodova zadan je globalnim cjelobrojnim poljem `bod` s 15 elemenata, tako da  $i$ -to mjesto u utrci donosi `bod[i - 1]` bodova (primjerice, `bod[0] = 25`, `bod[1] = 22`, ..., `bod[14] = 1`). Bodovi iz raznih utrka se zbrajaju.

- (a) Napišite funkciju `popis` koja prima prirodni broj  $n$  i vraća pokazivač na prvi element (dinamički kreiranog) polja svih vozača koji su osvojili bar jedan bod u prvih  $n$  utrka. Broj vozača u tom polju treba vratiti kroz varijabilni argument. Poredak  $i$ -te utrke ( $i \geq 1$ ) spremljen je u tekst-datoteci s imenom `z3c_i.txt`, gdje  $i$  označava tekstualni zapis broja  $i$  (poput `z3c_7.txt` ili `z3c_12.txt`). Svaki red u takvoj datoteci sadrži natjecateljski broj vozača (prirodan broj) i pripadno vrijeme u utrci (`double`, u sekundama). U svakoj datoteci, zapisi su poredani prema poretku u toj utrci — od prvog mjesta (pobjednika) do zadnjeg. Dozvoljeno je da utrka ima i manje od 15 vozača.
- (b) Napišite funkciju `rang_lista` koja prima polje vozača i broj vozača u polju. Funkcija treba zadano polje sortirati **silazno** po trenutnom broju bodova, koristeći poziv funkcije `qsort` iz standardne biblioteke `stdlib.h`. Ako više vozača ima isti broj bodova, njihov poredak smije biti bilo koji. Za poziv funkcije `qsort`, prvo trebate napisati funkciju za usporedbu dva vozača po trenutnom broju bodova. Pazite na rezultat usporedbe, jer `qsort` sortira uzlazno, a polje treba sortirati silazno!

**Napomene:** Dozvoljeno je koristiti jedan string za ime datoteke (najviše 20 znakova) i jedno dinamički alocirano polje za vozače koji nemaju nula bodova (treba ga realocirati po potrebi). Ne treba provjeravati otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 4.** (15 bodova) Podatke o vozačima automobila u svjetskom kupu spremamo u obliku strukture tipa `voz_bod` (isti tip kao u 3. zadatku), koja sadrži: prezime i ime (dva stringa od po 40 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu ekipe (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj).

Napišite program koji s komandne linije učitava dva argumenta. Prvi je ime postojeće (ulazne) binarne datoteke, a drugi je ime nove (izlazne) tekst-datoteke. Binarna datoteka sadrži niz podataka o vozačima, sortiran silazno po trenutnom broju bodova (trenutni bodovi smiju biti i nula). Prema tom poretku, treba kreirati startnu listu za sljedeću utrku.

Vozači su podijeljeni u grupe od po 10, redom prema poretku, s tim da zadnja grupa može biti i manja. Vodeća grupa (prvih 10, ako ih ima toliko) starta kao zadnja, a unutar te grupe, vozači startaju redosljedom prema poretku, tj. vodeći u poretku starta kao prvi u grupi. Sve ostale grupe (od druge do zadnje, ako ih ima) startaju redom po grupama, a unutar svake grupe, vozači startaju slučajnim redom. Za generiranje slučajnog startnog poretka unutar jedne takve grupe iskoristite funkciju `rand` iz standardne biblioteke `stdlib.h`.

U izlaznu datoteku treba napisati podatke o vozačima, redom kojim trebaju startati, po jedan vozač u svakom redu. Poredak i format podataka u redu je: redni broj starta (počevši od 1), natjecateljski broj, kratica ekipe, prezime, ime.

**Napomene:** Dozvoljeno je koristiti jedno polje cijelih brojeva i dva polja vozača, svako polje od po 10 elemenata. Ne trebate provjeravati argumente komandne linije i otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 30.6., navečer na webu, a uvidi u ponedjeljak, 1.7., u 11 sati.

**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.

**Zadatak 1.** (15 bodova) Podatke o nordijskim trkačima u svjetskom kupu spremamo u obliku strukture koja sadrži: ime i prezime (dva stringa od po 35 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i polje od 24 broja (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `trkac`.

Napišite funkciju koja učitava podatke za jednog trkača iz već otvorene tekst datoteke, zadane kao argument funkcije. Čitanje počinje od trenutne pozicije u toj datoteci. Funkcija treba vratiti pokazivač na dinamički alociranu strukturu s učitanim podacima.

Podaci o jednom trkaču zauzimaju tri reda teksta. 1. red: ime i prezime (odvojeni zarezom i jednom prazninom), s tim da ime i prezime smiju sadržavati praznine; 2. red: kratica države, praznina i broj; 3. red: niz od  $m \leq 24$  blokova znakova, s rezultatima u prvih  $m$  utrka. Svaki blok je ili prazan (nije sudjelovao) ili sadrži rezultat u obliku `h:mm:ss,n` (vrijeme završetka utrke — sati, minute, sekunde, a `n` su desetinke sekunde). Između svaka dva bloka nalazi se niz praznina (nula ili više), jedan znak `'!'` i niz praznina (nula ili više). Zadnji blok završava krajem reda. Treba učitati i taj znak za kraj reda. Broj blokova  $m$  nije unaprijed zadan, već ga treba odrediti iz oblika podataka u tom redu.

Pretpostavljamo da su svi podaci napisani u korektnom formatu. Kod spremanja vremena, podatak treba pretvoriti u sekunde. Za prazne blokove i buduće utrke (one za kojih nema pripadnih blokova) treba spremiti broj `-1.0`.

Primjer podataka za jednog trkača, s rezultatima (ovdje je  $m = 4$ ):

```
Johannes Hoesflot, Klaebo
NOR 67
2:13:15,0 ! ! 0:28:37,4!
```

**Napomena:** Zabranjeno je alociranje i korištenje bilo kakvih dodatnih polja. Takvo rješenje donosi 0 bodova!

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 2.** (20 bodova) Podatke o nordijskim trkačima u svjetskom kupu spremamo u vezanoj listi. Za svakog trkača pamtimo: ime i prezime (dva stringa od po 35 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i polje od 24 broja (`double`, rezultat u  $i$ -toj utrci, u sekundama). Definirajte odgovarajući tip podataka `trk` za jednog trkača, koji sadrži samo podatke nužne za spremanje takve liste u memoriji.

- (a) Napišite funkciju `trk* izbaci(trk* first, int n, trk** spori)`, gdje je `first` pokazivač na početak vezane liste trkača, a `n` je indeks utrke. Funkcija treba iz liste izbaciti sve one trkače koji u  $n$ -toj utrci imaju vrijeme strogo veće od 65% zbroja vremena prvog i zadnjeg u toj utrci. Izbačene trkače treba preseliti u zasebnu listu (poredak elemenata nije bitan), a njezin početak treba vratiti kroz varijabilni argument `spori`.
- (b) Napišite funkciju `trk* poredaj(trk* first, int n)`, gdje je `first` pokazivač na početak vezane liste trkača, a `n` je indeks utrke. Zadanu listu treba sortirati uzlazno, po vremenu u  $n$ -toj utrci (pobjednik treba doći na početak). Ako više trkača ima isto vrijeme, njihov poredak smije biti bilo koji (nije bitan). Za sortiranje **obavezno** iskoristite mergesort algoritam!

Smijete pretpostaviti da je indeks  $n$  korektno zadan ( $0 \leq n < 24$ ) i da su vremena u svim utrkama pozitivna.

**Napomena:** Nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 3.** (20 bodova) Podatke o nordijskim trkačima u svjetskom kupu spremamo u obliku strukture koja sadrži: ime i prezime (dva stringa od po 35 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj). Definirajte odgovarajući tip podataka `trk_bod`.

Na početku sezone, prije prve utrke, svaki trkač ima nula bodova. U svakoj utrci, samo prvih 30 trkača dobiva bodove. Broj bodova zadan je globalnim cjelobrojnim poljem `bod` s 30 elemenata, tako da  $i$ -to mjesto u utrci donosi `bod[i - 1]` bodova (primjerice, `bod[0] = 50`, `bod[1] = 45`, ..., `bod[29] = 1`). Bodovi iz raznih utrka se zbrajaju.

- (a) Napišite funkciju `popis` koja prima prirodni broj  $n$  i vraća pokazivač na prvi element (dinamički kreiranog) polja svih trkača koji su osvojili bar jedan bod u prvih  $n$  utrka. Broj trkača u tom polju treba vratiti kroz varijabilni argument. Poredak  $i$ -te utrke ( $i \geq 1$ ) spremljen je u tekst-datoteci s imenom `z3d_i.txt`, gdje  $i$  označava tekstualni zapis broja  $i$  (poput `z3d_7.txt` ili `z3d_12.txt`). Svaki red u takvoj datoteci sadrži natjecateljski broj trkača (prirodan broj) i pripadno vrijeme u utrci (`double`, u sekundama). U svakoj datoteci, zapisi su poredani prema poretku u toj utrci — od prvog mjesta (pobjednika) do zadnjeg. Dozvoljeno je da utrka ima i manje od 30 trkača.
- (b) Napišite funkciju `rang_lista` koja prima polje trkača i broj trkača u polju. Funkcija treba zadano polje sortirati **silazno** po trenutnom broju bodova, koristeći poziv funkcije `qsort` iz standardne biblioteke `stdlib.h`. Ako više trkača ima isti broj bodova, njihov poredak smije biti bilo koji. Za poziv funkcije `qsort`, prvo trebate napisati funkciju za usporedbu dva trkača po trenutnom broju bodova. Pazite na rezultat usporedbe, jer `qsort` sortira uzlazno, a polje treba sortirati silazno!

**Napomene:** Dozvoljeno je koristiti jedan string za ime datoteke (najviše 20 znakova) i jedno dinamički alocirano polje za trkače koji nemaju nula bodova (treba ga realocirati po potrebi). Ne treba provjeravati otvaranje datoteka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2019.

**Zadatak 4.** (15 bodova) Podatke o nordijskim trkačima u svjetskom kupu spremamo u obliku strukture tipa `trk_bod` (isti tip kao u 3. zadatku), koja sadrži: ime i prezime (dva stringa od po 35 znakova), natjecateljski broj (prirodni broj, jedinstven za svakog), kraticu države (3 znaka) i trenutni broj bodova u svjetskom kupu (cijeli broj).

Napišite program koji s komandne linije učitava dva argumenta. Prvi je ime postojeće (ulazne) binarne datoteke, a drugi je ime nove (izlazne) tekst-datoteke. Binarna datoteka sadrži niz podataka o trkačima, sortiran silazno po trenutnom broju bodova (trenutni bodovi smiju biti i nula). Prema tom poretku, treba kreirati startnu listu za sljedeću utrku.

Trkači su podijeljeni u grupe od po 25, redom prema poretku, s tim da zadnja grupa može biti i manja. Vodeća grupa (prvih 25, ako ih ima toliko) starta kao prva, a unutar te grupe, trkači startaju redosljedom prema poretku, tj. vodeći u poretku starta kao prvi. Sve ostale grupe (od druge do zadnje, ako ih ima) startaju redom po grupama, a unutar svake grupe, trkači startaju slučajnim redom. Za generiranje slučajnog startnog poretka unutar jedne takve grupe iskoristite funkciju `rand` iz standardne biblioteke `stdlib.h`.

U izlaznu datoteku treba napisati podatke o trkačima, redom kojim trebaju startati, po jedan trkač u svakom redu. Poredak i format podataka u redu je: redni broj starta (počevši od 1), natjecateljski broj, kratica države, prezime, ime.

**Napomene:** Dozvoljeno je koristiti jedno polje cijelih brojeva i jedno polja trkača, svako polje od po 25 elemenata. Ne trebate provjeravati argumente komandne linije i otvaranje datoteka.