

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Rezultati i uvidi: Rezultati će biti objavljeni na webu u petak 7.7. zajedno s informacijama o uvidima, koji će se održati u ponedjeljak 10.7.

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 i pametne satove 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 (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 (3+7+5+10* bodova) U ovom zadatku promatramo pojednostavljenu verziju sustava rangiranja tenisača na ATP ljestvici. Svaki tjedan, tenisači na temelju uspjeha na turnirima osvajaju određeni broj bodova, koji je uvijek **nenegativni cijeli broj**. Trenutni broj bodova pojedinog tenisača računa se na temelju bodova osvojenih u posljednja **52 tjedna** i to na način da se zbroji **19 najvećih** brojeva bodova među njima. Tenisači se tada sortiraju silazno po trenutnom broju bodova i na taj se način dobiva trenutna ATP ljestvica.

(a) Definirajte strukturu `tenisac` koja pamti podatke o jednom tenisaču: ime i prezime (stringovi od po najviše 30 znakova), starost u godinama (realan broj s najviše jednim decimalnim mjestom), kratica zemlje (točno 3 znaka), trenutni broj bodova i broj osvojenih bodova u svakom od posljednja 52 tjedna. Struktura mora biti definirana tako da bude moguća deklaracija oblika `tenisac t`; te uz zadane podatke smije sadržavati **još najviše jedan cijeli broj**.

(b) Napišite funkciju

```
void azuriraj(tenisac* t, int n, char** ime, char** prezime, int* bodovi, int m)
```

koja radi sljedeće. Ona prima niz tenisača `t` duljine `n` koji sadrži sve tenisače čije bodove pratimo. Ti su tenisači **sortirani leksikografski prvo po prezimenu pa po imenu**. Također, ona prima imena, prezimena te brojeve osvojenih bodova za `m` tenisača koji su sudjelovali na turnirima u posljednjem tjednu. Funkcija treba za svakog tenisača iz niza `t` ažurirati brojeve osvojenih bodova u posljednja 52 tjedna te trenutni broj bodova. Ako neki tenisač nije sudjelovao na turniru taj tjedan, smatramo da je osvojio 0 bodova. Možete pretpostaviti da nikoja dva tenisača nemaju istovremeno isto ime i prezime. Ažuriranje je potrebno izvesti algoritmom ukupne složenosti $\mathcal{O}(m \log n + n)$.

(c) Napišite funkciju `void ispisi(tenisac* t, int n, int starost)` koja radi sljedeće. Ona prima niz `t` od `n` tenisača koje treba sortirati silazno po trenutnom broju bodova te potom ispisati one čiji je broj godina manji od `starost`. Podatke o svakom tenisaču treba ispisati u zasebnom retku, u sljedećem obliku: rang, ime, prezime, starost u godinama, kratica zemlje **zajedno s rangom među tenisačima iz iste zemlje** te trenutni broj bodova.

Primjer: Jedan primjer valjanog ispisa unutar funkcije `ispisi`:

```
1 Novak Djokovic 36.0 SRB1 16100
2 Carlos Alcaraz 20.1 ESP1 13860
3 Daniil Medvedev 27.3 RUS1 12740
4 Rafael Nadal 37.0 ESP2 11055
```

Za dodatnih 10* bodova: Sortiranje i ispis tenisača u funkciji `ispisi` izvedite algoritmom složenosti $\mathcal{O}(n \log n)$.

Napomena: U ovom zadatku, uz standardnu biblioteku `stdio.h`, dozvoljeno je i korištenje biblioteka `stdlib.h` i `string.h`.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 2 (4+8+8 bodova) Podaci o igračima i igračicama badmintona zapisani su u dvije vezane liste **iste duljine muskarci** i **zene**, redom. Element svake od tih vezanih lista, struktura `player`, sadrži podatke o prezimenu (string maksimalne duljine 20) i rangu (cijeli broj) igrača/ice.

- (a) Napišite funkciju `int izdvoji(player* first, int n)` koja vraća n -ti element silazno sortiranog niza rangova igrača/ica u listi `first`.
- (b) Struktura `mixed_double` sadrži podatke o igraču i igračici koji igraju parove, a to su dva prezimena (stringovi maksimalne duljine 20) i dva ranga (cijeli brojevi).
Napišite funkciju `mixed_double* stvori_parove(player* zene, player* muskarci)` koja vraća pokazivač na listu u čije su elemente, strukture `mixed_double`, uneseni podaci iz lista `zene` i `muskarci`, tako da su u paru igrač i igračica s **istog rednog mjesta** prema rangu u listama. Lista treba biti **uzlazno sortirana** prema rangu igračica. Nakon unosa podataka o igraču, odgovarajuću strukturu u listi `muskarci` treba **obrisati**. Lista `zene` ostaje **nepromijenjena**.
- (c) Napišite funkciju `mixed_double* invertiraj(mixed_double* first)` koja vraća pokazivač na invertiranu listu `first`, **bez** alociranja dodatne memorije.

Primjer:

Lista `zene`: `{{"Tai",2}, {"Yamaguchi",1}, {"Sindhu",5}}`

Lista `muskarci`: `{{"Axelsen",1}, {"Popov",4}, {"Antonsen",3}}`

Lista `parovi` na koju pokazuje povratna vrijednost funkcije `stvori_parove`:

`{{"Yamaguchi","Axelsen",1,1}, {"Tai","Antonsen",2,3}, {"Sindhu","Popov",5,4}}`

Lista na koju pokazuje povratna vrijednost funkcije `invertiraj` nakon poziva na listu `parovi`:

`{{"Sindhu","Popov",5,4}, {"Tai","Antonsen",2,3}, {"Yamaguchi","Axelsen",1,1}}`

Napomena: Dozvoljeno je korištenje jednog pomoćnog niza cijelih brojeva. Možete pretpostaviti da su prezime i rangovi igrača/ica jedinstveni za svakog od njih.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 3 (15 bodova) Grupa prijatelja putuje u London. Za svaki dan koji provedu u Londonu sastavili su plan razgledavanja te ga zapisali u tekstualnu datoteku naziva "datum.txt" (datum je string oblika dd-mm-yyyy). Svaki redak datoteke je oblika "hh:mm-znamenitost". U tekstualnoj datoteci "cijene.txt" zapisane su cijene ulaznica za pojedine znamenitosti. Svaki redak datoteke "cijene.txt" je oblika "znamenitost:cijena" gdje je cijena **prirodan broj** (cijena u eurima). Naziv svake znamenitosti ima najviše **50** znakova. Ako u datoteci "cijene.txt" nema neke znamenitosti, znači da se ulaz ne naplaćuje.

- Napišite funkciju `int jedan_dan(char *datum, FILE *cijene)` koja kao argumente prima string `datum` (u gore opisanom obliku) te pokazivač `cijene` na datoteku "cijene.txt" otvorenu za čitanje. Funkcija vraća ukupnu cijenu ulaznica za znamenitosti iz datoteke "datum.txt".
- Napišite glavni program koji s **komandne linije** prima datume koje će prijatelji provesti u Londonu (zadani su u kronološkom redosljedju). Program mora u tekstualnu datoteku "ukupno.txt" zapisati kolika je cijena ulaznica za pojedini dan (svaki dan zapisan je u jednom retku u obliku "datum:cijena") te ukupnu cijenu za cijelo putovanje. Datumi u izlaznoj datoteci moraju biti poredani kronološki.

Primjer ulaznih datoteka i izlazne datoteke nakon izvršavanja glavnog programa čiji argumenti komandne linije su "27-07-2023", "28-07-2023" i "29-07-2023":

27-07-2023.txt

10:00-Buchinghamska palaca
14:25-Westminister Abbey
17:20-Big Ben

29-07-2023.txt

08:10-Kew Gardens
12:10-London Zoo

cijene.txt

Westminister Abbey:27
London Eye:31
London Zoo:28
Buchinghamska palaca:30
Kew Gardens:9

ukupno.txt

27-07-2023:57
28-07-2023:31
29-07-2023:37
ukupno:125

28-07-2023.txt

09:15-Prirodoslovni muzej
15:30-British Museum
19:50-London Eye

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje najviše četiri dodatna stringa pojedinačne duljine najviše 50 znakova**. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`. Ne morate provjeravati uspješnost funkcija za rad s datotekama.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 4 (2+13 bodova) U trgovini kućnog namještaja, stolovi su na rasprodaji tako da su na početne cijene primijenjeni popusti od 60 ili 70 posto. O svakom stolu pamtimo sljedeće podatke: početnu cijenu (prirodni broj), popust koji se primjenjuje (60 ili 70) te prirodni broj koji označava redni broj pod kojim se ta struktura nalazi u binarnoj datoteci **FILE*** **in** otvorenoj za čitanje i pisanje.

- (a) Definirajte strukturu **stol** koja pamti navedene podatke o jednom stolu. Struktura mora biti definirana tako da bude moguća deklaracija oblika **stol s**; te smije sadržavati samo zadane podatke.
- (b) Napišite funkciju **void grupiraj(FILE* in)** koja mijenja binarnu datoteku **in** grupirajući stolove tako da su prvo zapisani oni koji su na popustu od 60 posto pa zatim oni koji su na popustu od 70 posto, tako da unutar grupa redosljed bude **isti** kao u početnoj datoteci.

Primjer: Radi jednostavnosti, izostavit ćemo podatke o početnoj cijeni.

Početna binarna datoteka **in**: {1. 60}, {2. 70}, {3. 60}, {4. 70}, {5. 70}, {6. 60}, {7. 60}

Završna binarna datoteka **in**: {1. 60}, {3. 60}, {6. 60}, {7. 60}, {2. 70}, {4. 70}, {5. 70}

Napomena: Nije dozvoljeno korištenje pomoćnih polja niti dodatnih datoteka.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Rezultati i uvidi: Rezultati će biti objavljeni na webu u petak 7.7. zajedno s informacijama o uvidima, koji će se održati u ponedjeljak 10.7.

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 i pametne satove 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 (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 (3+7+5+10* bodova) U ovom zadatku promatramo pojednostavljenu verziju sustava rangiranja tenisačica na WTA ljestvici. Svaki tjedan, tenisačice na temelju uspjeha na turnirima osvajaju određeni broj bodova, koji je uvijek **nenegativni cijeli broj**. Trenutni broj bodova pojedine tenisačice računa se na temelju bodova osvojenih u posljednja **52 tjedna** i to na način da se zbroji **16 najvećih** brojeva bodova među njima. Tenisačice se tada sortiraju silazno po trenutnom broju bodova i na taj se način dobiva trenutna WTA ljestvica.

(a) Definirajte strukturu `tenisacica` koja pamti podatke o jednoj tenisačici: ime i prezime (stringovi od po najviše 30 znakova), starost u godinama (realan broj s najviše jednim decimalnim mjestom), kratica zemlje (točno 3 znaka), trenutni broj bodova i broj osvojenih bodova u svakom od posljednja 52 tjedna. Struktura mora biti definirana tako da bude moguća deklaracija oblika `tenisacica t`; te uz zadane podatke smije sadržavati **još najviše jedan cijeli broj**.

(b) Napišite funkciju

```
void azuriraj(tenisacica* t, int n, char** ime, char** prezime, int* bodovi, int m)
```

koja radi sljedeće. Ona prima niz tenisačica `t` duljine `n` koji sadrži sve tenisačice čije bodove pratimo. Te su tenisačice **sortirane leksikografski prvo po prezimenu pa po imenu**. Također, ona prima imena, prezimena te brojeve osvojenih bodova za `m` tenisačica koje su sudjelovale na turnirima u posljednjem tjednu. Funkcija treba za svaku tenisačicu iz niza `t` ažurirati brojeve osvojenih bodova u posljednja 52 tjedna te trenutni broj bodova. Ako neka tenisačica nije sudjelovala na turniru taj tjedan, smatramo da je osvojila 0 bodova. Možete pretpostaviti da nikoje dvije tenisačice nemaju istovremeno isto ime i prezime. Ažuriranje je potrebno izvesti algoritmom ukupne složenosti $\mathcal{O}(m \log n + n)$.

(c) Napišite funkciju `void ispisi(tenisacica* t, int n, int starost)` koja radi sljedeće. Ona prima niz `t` od `n` tenisačica koje treba sortirati silazno po trenutnom broju bodova te potom ispisati one čiji je broj godina manji od `starost`. Podatke o svakoj tenisačici treba ispisati u zasebnom retku, u sljedećem obliku: rang, ime, prezime, starost u godinama, kratica zemlje **zajedno s rangom među tenisačicama iz iste zemlje** te trenutni broj bodova.

Primjer: Jedan primjer valjanog ispisa unutar funkcije `ispisi`:

```
1 Iga Swiatek 22.0 POL1 13550
2 Aryna Sabalenka 25.1 BLR1 10355
3 Coco Gauff 19.2 USA1 8665
4 Jessica Pegula 29.3 USA2 7765
```

Za dodatnih 10* bodova: Sortiranje i ispis tenisačica u funkciji `ispisi` izvedite algoritmom složenosti $\mathcal{O}(n \log n)$.

Napomena: U ovom zadatku, uz standardnu biblioteku `stdio.h`, dozvoljeno je i korištenje biblioteka `stdlib.h` i `string.h`.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 2 (4+8+8 bodova) Podaci o igračima i igračicama badmintona zapisani su u dvije vezane liste **iste duljine muskarci** i **zene**, redom. Element svake od tih vezanih lista, struktura `player`, sadrži podatke o prezimenu (string maksimalne duljine 20) i rangu (cijeli broj) igrača/ice.

- (a) Napišite funkciju `int izdvoji(player* first, int n)` koja vraća n -ti element silazno sortiranog niza rangova igrača/ica u listi `first`.
- (b) Struktura `mixed_double` sadrži podatke o igraču i igračici koji igraju parove, a to su dva prezimena (stringovi maksimalne duljine 20) i dva ranga (cijeli brojevi).
Napišite funkciju `mixed_double* stvori_parove(player* zene, player* muskarci)` koja vraća pokazivač na listu u čije su elemente, strukture `mixed_double`, uneseni podaci iz lista `zene` i `muskarci`, tako da su u paru igrač i igračica s **istog rednog mjesta** prema rangu u listama. Lista treba biti **uzlazno sortirana** prema rangu igračica. Nakon unosa podataka o igraču, odgovarajuću strukturu u listi `muskarci` treba **obrisati**. Lista `zene` ostaje **nepromijenjena**.
- (c) Napišite funkciju `mixed_double* invertiraj(mixed_double* first)` koja vraća pokazivač na invertiranu listu `first`, **bez** alociranja dodatne memorije.

Primjer:

Lista `zene`: `{{"Young",3}, {"Marin",2}, {"Chen",6}}`

Lista `muskarci`: `{{"Loh",3}, {"Lee",7}, {"Ginting",5}}`

Lista `parovi` na koju pokazuje povratna vrijednost funkcije `stvori_parove`:

`{{"Marin","Loh",2,3}, {"Young","Ginting",3,5}, {"Chen","Lee",6,7}}`

Lista na koju pokazuje povratna vrijednost funkcije `invertiraj` nakon poziva na listu `parovi`:

`{{"Chen","Lee",6,7}, {"Young","Ginting",3,5}, {"Marin","Loh",2,3}}`

Napomena: Dozvoljeno je korištenje jednog pomoćnog niza cijelih brojeva. Možete pretpostaviti da su prezime i rangovi igrača/ica jedinstveni za svakog od njih.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 3 (15 bodova) Grupa prijatelja putuje u Rim. Za svaki dan koji provedu u Rimu sastavili su plan razgledavanja te ga zapisali u tekstualnu datoteku naziva "datum.txt" (datum je string oblika dd-mm-yyyy). Svaki redak datoteke je oblika "hh:mm-znamenitost". U tekstualnoj datoteci "cijene.txt" zapisane su cijene ulaznica za pojedine znamenitosti. Svaki redak datoteke "cijene.txt" je oblika "znamenitost:cijena" gdje je cijena **prirodan broj** (cijena u eurima). Naziv svake znamenitosti ima najviše **50 znakova**. Ako u datoteci "cijene.txt" nema neke znamenitosti, znači da se ulaz ne naplaćuje.

- Napišite funkciju `int dan(char *datum, FILE *cijene)` koja kao argumente prima string `datum` (u gore opisanom obliku) te pokazivač `cijene` na datoteku "cijene.txt" otvorenu za čitanje. Funkcija vraća ukupnu cijenu ulaznica za znamenitosti iz datoteke "datum.txt".
- Napišite glavni program koji s **komandne linije** prima datume koje će prijatelji provesti u Rimu (zadani su u kronološkom redoslijedu). Program mora u tekstualnu datoteku "ukupno.txt" zapisati kolika je cijena ulaznica za pojedini dan (svaki dan zapisan je u jednom retku u obliku "datum:cijena") te ukupnu cijenu za cijelo putovanje. Datumi u izlaznoj datoteci moraju biti poredani kronološki.

Primjer ulaznih datoteka i izlazne datoteke nakon izvršavanja glavnog programa čiji argumenti komandne linije su "17-08-2023", "18-08-2023" i "19-08-2023":

17-08-2023.txt

09:30-Kolosej
13:20-Forum Romanum
18:45-Spomenik Viktoru Emanuelu

19-08-2023.txt

09:15-Vatikanski muzeji
15:30-Sikstinska kapela
16:45-Bazilika sv. Petra

ukupno.txt

17-08-2023:16
18-08-2023:13
19-08-2023:44
ukupno:73

18-08-2023.txt

09:30-Fontana di Trevi
10:20-Pantheon
12:10-Piazza Navona
15:20-Andeoska tvrđava

cijene.txt

Vatikanski muzeji:27
Sikstinska kapela:17
Forum Romanum:7
Kolosej:9
Andeoska tvrđava:13

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje najviše četiri dodatna stringa pojedinačne duljine najviše 50 znakova**. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`. Ne morate provjeravati uspješnost funkcija za rad s datotekama.

Programiranje 2 – drugi kolokvij, 30. 6. 2023.

Zadatak 4 (2+13 bodova) U trgovini kućnog namještaja, stolovi su na rasprodaji tako da su na početne cijene primijenjeni popusti od 65 ili 75 posto. O svakom stolu pamtimo sljedeće podatke: početnu cijenu (prirodni broj), popust koji se primjenjuje (65 ili 75) te prirodni broj koji označava redni broj pod kojim se ta struktura nalazi u binarnoj datoteci **FILE*** **in** otvorenoj za čitanje i pisanje.

- Definirajte strukturu **stol** koja pamti navedene podatke o jednom stolu. Struktura mora biti definirana tako da bude moguća deklaracija oblika **stol s**; te smije sadržavati samo zadane podatke.
- Napišite funkciju **void grupiraj(FILE* in)** koja mijenja binarnu datoteku **in** grupirajući stolove tako da su prvo zapisani oni koji su na popustu od 65 posto pa zatim oni koji su na popustu od 75 posto, tako da unutar grupa redosljed bude **isti** kao u početnoj datoteci.

Primjer: Radi jednostavnosti, izostavit ćemo podatke o početnoj cijeni.

Početna binarna datoteka **in**: {1. 65}, {2. 75}, {3. 65}, {4. 75}, {5. 75}, {6. 65}, {7. 65}

Završna binarna datoteka **in**: {1. 65}, {3. 65}, {6. 65}, {7. 65}, {2. 75}, {4. 75}, {5. 75}

Napomena: Nije dozvoljeno korištenje pomoćnih polja niti dodatnih datoteka.