JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Rezultati i uvidi: Rezultati do petka 8.7.2022. navečer na webu. Termini uvida bit će objavljeni 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 (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 ($2 + 5 + 5 + 4 + 4 = 20$ bodova) Jedna video platforma za prikazivanje filmova želi čuvati podatke o filmovima koji se na njoj mogu gledati. O svakom filmu treba čuvati sljedeće podatke: naziv filma, godinu snimanja, trajanje (u minutama), te popis imena glumaca. Pritom su godina i trajanje cijeli brojevi, dok je naziv filma i ime pojedinog glumca string (čija duljina nije unaprijed ograničena!).

(a) Definirajte tip `film` za pohranu podataka o jednome filmu tako da bude moguća deklaracija `film a;`. U strukturi smiju biti samo polja koja su nužna za rješenje ovog zadatka.

(b) Napišite funkciju `film* novi(void)` koja će omogućiti korisniku da s tipkovnice učita novi unos u formatu

```
naziv(godina, trajanje) - niz imena glumaca odvojenih zarezom i jednim razmakom
```

primjerice:

```
The Ring(2002, 115) - Naomi Watts, Martin Henderson, David Dorfman, Brian Cox, Daveigh Chase
```

(pretpostavljamo da naziv filma ne sadrži znak '(', te da ime pojedinog glumca ne sadrži znak ',' - smijete pretpostaviti da svaki film ima barem jednog glumca). Funkcija vraća pokazivač na alociranu varijablu u koju su spremljeni uneseni podaci. Prilikom spremanja podataka, oni trebaju zauzimati točno koliko je memorije potrebno!

(c) Napišite funkciju `film* najbolji(film* filmovi, int m, char** glumci, int n)` koja prima polje filmova `filmovi` duljine `m`, te niz od `n` imena glumaca `glumci`. Funkcija treba vratiti pokazivač na onaj element polja koji ima najviše imena glumaca koja se pojavljuju i u polju `glumci`. Ako takvih filmova ima više, onda se uzima onaj koji je najranije snimljen. Ako i takvih ima više, uzima se onaj najmanjeg indeksa. Ukoliko ne postoji film u polju filmova koji ima bar 1 glumca iz polja `glumci`, funkcija vraća `NULL`.

(d) Napišite funkciju `void sortiraj(film* filmovi, int n)` koja sortira primljeno polje `filmovi` duljine `n` po imenu filma (uzlazno po abecedi, pri čemu ne razlikujemo velika i mala slova). Ukoliko dva filma imaju isto ime (uz zanemarivanje razlike velikih i malih slova), tada prije dolazi ono s ranijom godinom (a ako je i ona ista, onda nam je svejedno koji film dolazi ranije). Pritom je sortiranje potrebno izvesti algoritmom čija je prosječna vremenska složenost $\mathcal{O}(n \log n)$.

(e) Napišite funkciju `int postoji(film* filmovi, int n, char* naziv)` koja prima polje filmova `filmovi` duljine `n` i vraća 1 ako postoji barem jedan film u tom polju čiji je naziv jednak `naziv` (pri čemu ne razlikujemo velika i mala slova). Inače, funkcija vraća 0. Pritom je niz `filmovi` sortiran kako je opisano u (d) podzadatku. Ovdje je traženje filma sa zadanim nazivom potrebno izvesti algoritmom čija je složenost logaritamska.

Napomena: Kako bi riješili zadatak, nužno (u smislu ostvarivanja > 0 bodova) je riješiti podzadatak (a). Ne pridržavanje zahtjeva na složenost algoritma u (d), odnosno (e) podzadatku rezultira ostvarivanjem najviše 0 bodova u podzadatku u kojem se nije poštivao taj zahtjev. U ovome zadatku dozvoljeno je korištenje sljedećih biblioteka: `stdio.h`, `stdlib.h`, `string.h`.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 2 (2+6+12=20 bodova) Popis kolegija za određeni semestar nekog studija sprema se u vezanu listu. Za svaki kolegij pamtimo naziv kolegija (string duljine najviše 30), šifru kolegija (prirodni broj), broj kolegija prethodnika (nenegativan cijeli broj) te niz prirodnih brojeva koji predstavlja šifre kolegija prethodnika. Moguće je da kolegij nema kolegija prethodnika. U tom slučaju je broj kolegija prethodnika 0, a niz šifri kolegija prethodnika je određen `NULL` pokazivačem. Dodatno, za svaki kolegij pamtimo i godinu na kojoj se održava (prirodni broj) te vrstu semestra u kojem se održava (znak `'Z'` za zimski semestar i `'L'` za ljetni semestar).

(a) Napišite deklaraciju odgovarajućeg tipa podataka za kolegij, na način da bude moguće definirati varijablu naredbom `kolegij a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.

(b) Napišite funkciju `kolegij* azuriraj(kolegij* novi, int sljedbenici[], int br_sljedbenika, kolegij* first)` koja prima pokazivač `novi` na novi kolegij i pokazivač `first` na semestar kojeg treba ažurirati. Ako se novi kolegij održava na istoj godini i istoj vrsti semestra kao i svi kolegiji u listi `first` (ili ako je lista prazna), kolegij dodajemo na početak liste. U protivnom, listu treba ažurirati tako da se novi kolegij doda kao kolegij prethodnik kolegijima iz niza `sljedbenici`. Ukoliko se u listi ne nalazi niti jedan sljedbenik novog kolegija, lista ostaje nepromijenjena. Funkcija vraća pokazivač na prvi element ažurirane liste.

(c) Napišite funkciju `kolegij* upisi(kolegij** semestar, int polozeni[], int br_polozenih)` koja prima pokazivač na pokazivač na listu kolegija semestra kojeg upisuje neki student te niz koji predstavlja šifre položenih kolegija tog studenta. Funkcija razdvaja semestar na dvije vezane liste. Prva sadrži kolegije koje student može upisati i funkcija vraća pokazivač na prvi element te liste. Ukoliko student ne može upisati niti jedan kolegij funkcija vraća `NULL`. Druga se sastoji od kolegija koje student ne može upisati i pokazivač na prvi element ove liste se vraća preko varijabilnog argumenta `semestar`. Dodatno, lista s kolegijima koji se ne mogu upisati mora biti sortirana uzlazno po broju nepoloženih kolegija prethodnika. Sortiranje i razdvajanje se treba napraviti samo razmještanjem elemenata bez kopiranja memorije ili dodatnih alokacija.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 3 (10+2=12 bodova) Cezarova šifra predstavlja način šifriranja teksta u kojem se svako slovo teksta zamjenjuje odgovarajućim slovom abecede, pomaknutim za određeni broj mjesta. Taj broj mjesta za koji se pomičemo nazivamo ključ. Na primjer, ako je ključ 3, svako slovo ćemo zamijeniti sa slovom koje je za 3 mjesta udaljeno od njega (A ćemo zamijeniti s D, c ćemo zamijeniti s f, Z ćemo zamijeniti s C, itd.)

(a) Napišite funkciju `sifriraj` koja prima pokazivač na tekstualnu datoteku koja je otvorena za čitanje i pisanje te cijeli broj `k` koji predstavlja ključ za Cezarovu šifru. Funkcija mora tekst u datoteci šifrirati koristeći Cezarovu šifru sa ključem `k`. Pretpostavljamo da datoteka sadrži samo slova engleske abecede.

(b) Napišite glavni program koji testira funkciju `sifriraj`. U njemu se učitava ime tekstualne datoteke koja se treba šifrirati te prirodni broj koji predstavlja ključ za Cezarovu šifru. Provjerite uspješnost otvaranja datoteke te u slučaju greške prekinite program i vratite neki prirodni broj po izboru. Nakon završetka rada s datotekom, datoteku treba zatvoriti.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 4 (2 + 10 + 3 = 15 bodova) Na teniskom turniru sudjeluje **128 igrača**. Turnir se odvija na sljedeći način. Igraju se **mečevi**, a u svakom meču sudjeluju dva igrača. Svaki meč sastoji se od nekoliko **setova**. Svaki set osvaja jedan od igrača, a meč pobjeđuje igrač koji prvi osvoji **tri seta** (pretpostavljamo da nema predanih mečeva). Najprije se u prvom kolu igrači podijele u 64 para te igrači iz svakog para međusobno igraju jedan meč. Nakon svakog meča, gubitnik ispada iz turnira, a pobjednik nastavlja dalje. Zatim se, u drugom kolu, pobjednici mečeva prvog kola podijele u 32 para te ponovo igrači iz svakog para međusobno igraju jedan meč. Gubitnici mečeva ispadaju, a pobjednici prolaze u sljedeće kolo. Ovaj se postupak nastavlja sve do posljednjeg kola, tj. finala, u kojemu se igra samo jedan meč. Pobjednik tog meča proglašava se **pobjednikom** turnira.

(a) Definirajte strukturu `mec` koja pamti podatke o jednom odigranom meču: ime prvog igrača, ime drugog igrača te brojeve osvojenih setova pojedinog igrača. Imena igrača su stringovi duljine najviše 50. Struktura mora biti definirana tako da bude moguća deklaracija oblika `mec m;` te smije sadržavati samo zadane podatke.

(b) Napišite funkciju `void pobjednik(FILE* ulaz, FILE* izlaz)` koja radi sljedeće. Funkcija prima pokazivače `FILE* ulaz` na binarnu datoteku koja je otvorena za čitanje te `FILE* izlaz` na tekstualnu datoteku koja je otvorena za pisanje. U binarnoj datoteci na koju pokazuje `ulaz`, u obliku niza struktura tipa `mec` zapisani su podatci o svim mečevima koji su odigrani tijekom turnira, **ne nužno u kronološkom redoslijedu**. Funkcija `pobjednik` treba na temelju tih podataka odrediti pobjednika turnira te njegovo ime zapisati u datoteku na koju pokazuje `izlaz`.

(c) Napišite program koji s komandne linije prima dva argumenta. Prvi je ime ulazne binarne datoteke, a drugi je ime izlazne tekstualne datoteke. Program treba otvoriti obje datoteke, pozvati funkciju `pobjednik` iz (b) dijela za te dvije datoteke i na kraju zatvoriti obje datoteke.

Napomena: Ne trebate provjeravati moguće greške kod čitanja argumenata komandne linije i rada s datotekama. U ovom zadatku smijete koristiti samo biblioteke `stdio.h` i `string.h` te **ne smijete koristiti dodatne nizove**.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Rezultati i uvidi: Rezultati do petka 8.7.2022. navečer na webu. Termini uvida bit će objavljeni 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 (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 ($2 + 5 + 5 + 4 + 4 = 20$ bodova) Jedna video platforma za prikazivanje filmova želi čuvati podatke o glumcima koji se pojavljuju u nekim od filmova koji se na njoj mogu gledati. O svakom glumcu treba čuvati sljedeće podatke: ime glumca, broj dobivenih nagrada Oscar, godina rođenja, te popis filmova u kojima taj glumac glumi. Pritom su broj Oscara i godina cijeli brojevi, dok je ime glumca i naziv pojedinog filma string (čija duljina nije unaprijed ograničena!).

(a) Definirajte tip `glumac` za pohranu podataka o jednome glumcu tako da bude moguća deklaracija `glumac g;`. U strukturi smiju biti samo polja koja su nužna za rješenje ovog zadatka.

(b) Napišite funkciju `glumac* unesi(void)` koja će omogućiti korisniku da s tipkovnice učita novi unos u formatu

```
ime glumca[broj Oscara; godina rođenja]:  niz naziva filmova odvojenih zarezom i jednim razmakom
```

primjerice:

```
Cate Blanchett[2; 1969]: The Lord of the Rings, The Monuments Men, Thor: Ragnarok, The Hobbit
```

(pretpostavljamo da ime glumca ne sadrži znak `'['`, te da naziv pojedinog filma ne sadrži znak `','` - smijete pretpostaviti da svaki glumac ima barem jedan film u kojem je glumio). Funkcija vraća pokazivač na alociranu varijablu u koju su spremljeni uneseni podaci. Prilikom spremanja podataka, oni trebaju zauzimati točno koliko je memorije potrebno!

(c) Napišite funkciju `glumac* naj(glumac* glumci, int n, char** filmovi, int m)` koja prima polje glumaca `glumci` duljine `n`, te niz od `m` naziva filmova `filmovi`. Funkcija treba vratiti pokazivač na onaj element polja koji ima najviše naziva filmova koji se pojavljuju i u polju `filmovi`. Ako takvih glumaca ima više, onda se uzima onaj s većim brojem Oscara. Ako i takvih ima više, uzima se onaj najmanjeg indeksa. Ukoliko ne postoji glumac u polju glumaca koji ima bar 1 naziv filma iz polja `filmovi`, funkcija vraća `NULL`.

(d) Napišite funkciju `void poslozi(glumac* glumci, int m)` koja sortira primljeno polje `glumci` duljine `m` po imenu glumca (uzlazno po abecedi, pri čemu ne razlikujemo velika i mala slova). Ukoliko dva glumca imaju isto ime (uz zanemarivanje razlike velikih i malih slova), tada prije dolazi onaj s manjim brojem Oscara (a ako je i on isti, onda nam je svejedno koji glumac dolazi ranije). Pritom je sortiranje potrebno izvesti algoritmom čija je prosječna vremenska složenost $\mathcal{O}(n \log n)$.

(e) Napišite funkciju `int trazi(glumac* glumci, int m, char* ime)` koja prima polje glumaca `glumci` duljine `m` i vraća 1 ako postoji barem jedan glumac u tom polju čije je ime jednako `ime` (pri čemu ne razlikujemo velika i mala slova). Inače, funkcija vraća 0. Pritom je niz `glumci` sortiran kako je opisano u (d) podzadatku. Ovdje je traženje glumca sa zadanim imenom potrebno izvesti algoritmom čija je složenost logaritamska.

Napomena: Kako bi riješili zadatak, nužno (u smislu ostvarivanja > 0 bodova) je riješiti podzadatak (a). Ne pridržavanje zahtjeva na složenost algoritma u (d), odnosno (e) podzadatku rezultira ostvarivanjem najviše 0 bodova u podzadatku u kojem se nije poštivao taj zahtjev. U ovome zadatku dozvoljeno je korištenje sljedećih biblioteka: `stdio.h`, `stdlib.h`, `string.h`.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 2 (2+6+12 = 20 bodova) Popis kolegija za određeni semestar nekog studija sprema se u vezanu listu. Za svaki kolegij pamtimo naziv kolegija (string duljine najviše 30), šifru kolegija (prirodni broj), broj kolegija prethodnika (nenegativan cijeli broj) te niz prirodnih brojeva koji predstavlja šifre kolegija prethodnika. Moguće je da kolegij nema kolegija prethodnika. U tom slučaju je broj kolegija prethodnika 0, a niz šifri kolegija prethodnika je određen `NULL` pokazivačem. Dodatno, za svaki kolegij pamtimo i godinu na kojoj se održava (prirodni broj) te vrstu semestra u kojem se održava (znak `'Z'` za zimski semestar i `'L'` za ljetni semestar).

(a) Napišite deklaraciju odgovarajućeg tipa podataka za kolegij, na način da bude moguće definirati varijablu naredbom `kolegij a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.

(b) Napišite funkciju `kolegij* azuriraj(kolegij* novi, int sljedbenici[], int br_sljedbenika, kolegij* first)` koja prima pokazivač `novi` na novi kolegij i pokazivač `first` na semestar kojeg treba ažurirati. Ako se novi kolegij održava na istoj godini i istoj vrsti semestra kao i svi kolegiji u listi `first` (ili ako je lista prazna), kolegij dodajemo na kraj liste. U protivnom, listu treba ažurirati tako da se novi kolegij doda kao kolegij prethodnik kolegijima iz niza `sljedbenici`. Ukoliko se u listi ne nalazi niti jedan sljedbenik novog kolegija, lista ostaje nepromijenjena. Funkcija vraća pokazivač na prvi element ažurirane liste.

(c) Napišite funkciju `kolegij* upisi(kolegij** semestar, int polozeni[], int br_polozenih)` koja prima pokazivač na pokazivač na listu kolegija semestra kojeg upisuje neki student te niz koji predstavlja šifre položenih kolegija tog studenta. Funkcija razdvaja semestar na dvije vezane liste. Prva sadrži kolegije koje student može upisati i funkcija vraća pokazivač na prvi element te liste. Ukoliko student ne može upisati niti jedan kolegij funkcija vraća `NULL`. Druga lista se sastoji od kolegija koje student ne može upisati i pokazivač na prvi element ove liste se vraća preko varijabilnog argumenta `semestar`. Dodatno, lista s kolegijima koji se ne mogu upisati mora biti sortirana silazno po broju nepoloženih kolegija prethodnika. Sortiranje i razdvajanje se treba napraviti samo razmještanjem elemenata bez kopiranja memorije ili dodatnih alokacija.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 3 (10+2 = 12 bodova) Cezarova šifra predstavlja način šifriranja teksta u kojem se svako slovo teksta zamjenjuje odgovarajućim slovom abecede, pomaknutim za određeni broj mjesta. Taj broj mjesta za koji se pomičemo nazivamo ključ. Na primjer, ako je ključ 3, svako slovo ćemo zamijeniti sa slovom koje je za 3 mjesta udaljeno od njega (A ćemo zamijeniti sa D, c ćemo zamijeniti sa f, Z ćemo zamijeniti sa C, itd.)
Dešifriranje predstavlja rekonstrukciju originalnog teksta iz šifriranog teksta. U slučaju Cezarove šifre, proces dešifriranja predstavlja zamjenu svakog slova pomaknutim za određeni broj mjesta ulijevo. Na primjer, ako je ključ šifriranja bio 3, svako slovo se mijenja slovom udaljenim za 3 mjesta od njega ulijevo (A ćemo zamijeniti s X, c ćemo zamijeniti sa z, Z ćemo zamijeniti s W, itd.).

(a) Napišite funkciju `desifriraj` koja prima pokazivač na tekstualnu datoteku koja je otvorena za čitanje i pisanje te cijeli broj `k` koji predstavlja ključ za Cezarovu šifru. Funkcija mora tekst u datoteci dešifrirati koristeći Cezarovu šifru sa ključem `k`. Pretpostavljamo da datoteka sadrži samo slova engleske abecede.

(b) Napišite glavni program koji testira funkciju `desifriraj`. U njemu se učitava ime tekstualne datoteke koja se treba dešifrirati te prirodni broj koji predstavlja ključ za Cezarovu šifru. Provjerite uspješnost otvaranja datoteke te u slučaju greške prekinite program i vratite neki prirodni broj po izboru. Nakon završetka rada s datotekom, datoteku treba zatvoriti.

JMBAG

IME I PREZIME

Programiranje 2 – drugi kolokvij, 1. 7. 2022.

Zadatak 4 (2 + 10 + 3 = 15 bodova) U završnom dijelu nogometnog prvenstva sudjeluje **16 reprezentacija**. Taj se dio prvenstva odvija na sljedeći način. Igraju se **utakmice**, a u svakoj utakmici sudjeluju dvije reprezentacije. One postižu određeni broj **pogodaka**, a pobjeđuje ona koja ih postigne **više** (nema neriješenih ishoda). Najprije se u prvom kolu reprezentacije podijele u 8 parova te reprezentacije iz svakog para međusobno igraju jednu utakmicu. Nakon svake utakmice, gubitnička reprezentacija ispada iz turnira, a pobjednička nastavlja dalje. Zatim se, u drugom kolu, reprezentacije koje su pobijedile u prvom kolu podijele u 4 para te ponovo reprezentacije iz svakog para međusobno igraju jednu utakmicu. Gubitničke reprezentacije ispadaju, a pobjedničke prolaze u sljedeće kolo. Ovaj se postupak nastavlja sve do posljednjeg kola, tj. finala, u kojemu se igra samo jedna utakmica. Reprezentacija koja pobijedi tu utakmicu proglašava se **prvakom**.

(a) Definirajte strukturu `utakmica` koja pamti podatke o jednoj odigranoj utakmici: ime prve reprezentacije, ime druge reprezentacije te brojeve postignutih pogodaka pojedine reprezentacije. Imena reprezentacija su stringovi duljine najviše 50. Struktura mora biti definirana tako da bude moguća deklaracija oblika `utakmica u;` te smije sadržavati samo zadane podatke.

(b) Napišite funkciju `void prvak(FILE* ulaz, FILE* izlaz)` koja radi sljedeće. Funkcija prima pokazivače `FILE* ulaz` na binarnu datoteku koja je otvorena za čitanje te `FILE* izlaz` na tekstualnu datoteku koja je otvorena za pisanje. U binarnoj datoteci na koju pokazuje `ulaz`, u obliku niza struktura tipa `utakmica` zapisani su podatci o svim utakmicama koje su odigrane tijekom turnira, **ne nužno u kronološkom redoslijedu**. Funkcija `prvak` treba na temelju tih podataka odrediti reprezentaciju koja je prvak te njeno ime zapisati u datoteku na koju pokazuje `izlaz`.

(c) Napišite program koji s komandne linije prima dva argumenta. Prvi je ime ulazne binarne datoteke, a drugi je ime izlazne tekstualne datoteke. Program treba otvoriti obje datoteke, pozvati funkciju `prvak` iz (b) dijela za te dvije datoteke i na kraju zatvoriti obje datoteke.

Napomena: Ne trebate provjeravati moguće greške kod čitanja argumenata komandne linije i rada s datotekama. U ovom zadatku smijete koristiti samo biblioteke `stdio.h` i `string.h` te **ne smijete koristiti dodatne nizove**.