

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Napomene. Svako rješenje 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 **obavezno** pišite postupak!

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, te razne tablice, papiri i sl. nisu dozvoljeni! **Mobitele** isključite i pospremite daleko od sebe! Ako se ustanovi da **kod sebe** imate mobitel za vrijeme kolokvija, kolokvij se poništava i pokreće se stegovni postupak protiv vas.

Zadatak 1 (16 bodova) Napišite program koji kao prvi argument komandne linije prima nenegativan cijeli broj, a kao drugi argument nenegativan realni broj zapisan u notaciji pomične točke (bez eksponenta, kao kod formata %f). Program treba spremi učitane brojeve u odgovarajuće varijable tipa `int` i `double`, te ispisati njihove vrijednosti. Pretvorba za oba broja se vrši pozivom funkcije **pretvori** koja kao argumente prima string reprezentaciju učitano broja, a preko varijabilnog argumenta vraća `int` ili `double` reprezentaciju tog broja (ovisno o ulaznom broju). Napišite sve dodatne potrebne funkcije i definirajte formalne argumente funkcije **pretvori** (može ih biti i više od nabrojanih) da bi mogli ostvariti navedenu funkcionalnost. Obavezno napišite implementaciju funkcije **main**. Pripazite da izbjegnute upozorenja prevodioca uzrokovana konverzijama tipova argumenata funkcija.

Napomene: zabranjeno je korištenje funkcija `atof`, `atoi`, `atol`, `strtod`, `strtoul`, `strtoul`. Dopušteno je korištenje ostalih funkcija definiranih u datotekama zaglavlja `stdlib.h` i `ctype.h`.

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 2 (2+12 bodova) Nakon održanog turnira društvenih igara, organizatori su odlučili napraviti rang ljestvicu igrača koji su sudjelovali na turniru. Svaka igra koja se igrala na turniru bila je igra za 4 igrača.

Za svakog igrača na turniru znamo njegovo ime koje ga jedinstveno određuje (string duljine najviše 30) i ukupan broj ostvarenih bodova na turniru (nenegativan cijeli broj). Podatke o igraču pamtimo u strukturi `player`.

Nadalje, za svaku igru koja se igrala na turniru znamo njezin naziv (string duljine najviše 50), težinu (prirodan broj između 1 i 10, uključivo) te niz `ekipa` od 4 igrača koji su odigrali partiju te igre (niz od 4 strukture `player`). Podatke o igri pamtimo u strukturi `game`. Poznato je da je niz `ekipa` **sortiran silazno** prema broju ostvarenih bodova u odigranoj partiji. Ako sa **tezina** označimo težinu igre, broj ostvarenih bodova u toj partiji igrača na indeksu `i` računamo prema formuli $tezina * (4 - i)$.

(a) Deklarirajte tip podatka `player` koji pamti navedene podatke o igraču. Dodatno, varijabla `x` tog tipa se treba moći definirati sa `player x`; . Deklarirajte tip podatka `game` koji pamti navedene podatke o igri. Dodatno, varijabla `y` tog tipa se treba moći definirati sa `game y`;

(b) Napišite funkciju

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

koja kao argumente dobiva niz odigranih igara na turniru u nizu `odigrane`, niz svih igrača koji su sudjelovali na turniru u nizu `igraci` (s početnim bodovima postavljenim na 0) te prirodne brojeve `m` i `n` koji su redom duljine nizova `odigrane` i `igraci`. Funkcija treba sortirati niz `igraci` prema ukupnim bodovima igrača ostvarenim na turniru i na ekran ispisati rang listu igrača u formatu:

```
redni broj.ime igrača.(broj bodova)\n
```

Ako postoje igrači s istim brojem bodova, poredak među njima nije bitan, ali im se ispisuje isti redni broj.

Napomena. U nizu `odigrane` je moguće imati više od jedne strukture `game` s istim nazivom i težinom igre. Nije dozvoljeno korištenje dodatnih nizova.

Primjer. Pokažimo na primjeru kako ažuriramo bodove igrača u nizu `igraci`. Za niz igrača

```
igraci: P1={Petar, 0}, P2={Luka, 0}, P3={Iva, 0}, P4={Marija, 0}, P5={Lucija, 0}, P6={Ante, 0}
```

ako je niz `odigrane` jednak

```
{Splendor, 2, [P5,P2,P6,P1]}, {Catan, 7, [P3,P1,P5,P4]}, {Splendor, 2, [P3,P6,P2,P4]},
```

nakon poziva funkcije `rang_lista(odigrane,igraci,3,6)`, niz `igraci` treba biti

```
igraci: P3={Iva, 36} P1={Petar, 23}, P5={Lucija, 15}, P2={Luka, 10}, P6={Ante, 10}, P4={Marija, 9},
```

a ispis:

1. Iva (36)
2. Petar (23)
3. Lucija (15)
4. Luka (10)
4. Ante (10)
6. Marija (9)

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 3 (2+11 bodova) Svi automobili dostupni za prodaju u nekoj firmi nalaze se u vezanoj listi. Za svaki automobil pamtimo naziv (string s najviše 20 znakova), cijenu (u prirodnim brojevima) i količinu prijeđenih kilometara.

- (a) Napišite deklaraciju odgovarajućeg tipa podatka za automobil na način da bude moguće definirati varijablu naredbom `auto a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `auto* izbor(auto* auti, char** nazivi, int n, int maxcijena, int maxkm)` koja prima pokazivač na listu automobila, niz stringova u kojem se nalaze nazivi automobila, duljinu tog niza te prirodne brojeve koji predstavljaju cijenu i kilometražu. Funkcija kreira novu vezanu listu u kojoj se nalaze automobili iz vezane liste `auti` čiji se naziv nalazi u nizu `nazivi`, a čija je cijena manja ili jednaka od `maxcijena` i broj prijeđenih kilometara manji ili jednak od `maxkm`.
- (c) Napišite funkciju `auto* invertiraj(auto* izbor)` koja vraća pokazivač na invertiranu vezanu listu liste `izbor`.

Napomena. Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**. Dozvoljeno je korištenje funkcija iz `stdlib.h` i `string.h`.

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 4 (2+10 bodova) Zapis o kupovini nekog proizvoda u trgovini se čuva u strukturi `proizvod` koja se sastoji od sljedećih podataka: ime proizvoda (string od najviše 50 znakova), cijena proizvoda (double) i broj kupljenih proizvoda (int). Podaci o svim kupovinama se čuvaju u binarnoj datoteci. Isti proizvod se može pojaviti više puta u datoteci. Možete pretpostaviti da trgovina može imati maksimalno 50 različitih proizvoda. Ne moraju se svi proizvodi pojaviti u datoteci.

- Definirajte strukturu `proizvod` koja pamti navedene podatke o jednom proizvodu. Struktura mora biti definirana tako da bude moguća deklaracija oblika `proizvod p`; te smije sadržavati samo zadane podatke.
- Napišite funkciju `void obradiPodatke(FILE* in, FILE* out)` koja kao argumente prima pokazivač `in` na binarnu datoteku otvorenu za čitanje te pokazivač `out` na tekstualnu datoteku otvorenu za pisanje. Funkcija u tekstualnu datoteku upisuje ukupnu količinu kupljenih proizvoda iz binarne datoteke u sljedećem obliku "ime cijena ukupnaKolicina". Nakon svakog proizvoda ide skok u novi red. Redoslijed proizvoda nije bitan. Možete pretpostaviti da binarna datoteka nije prazna.

Primjer:

Početna binarna datoteka `in`:

```
{Mlijeko, 5.2, 3}, {Sir, 8.7, 1}, {Jaja, 0.5, 10}, {Mlijeko, 5.2, 1},  
{Mlijeko, 5.2, 2}, {Sir, 8.7, 5}
```

Završna tekstualna datoteka `out`:

```
Mlijeko 5.2 6  
Sir 8.7 6  
Jaja 0.5 10
```

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje jednog niza struktura proizvod duljine 50**. Nije dozvoljeno korištenje dodatnih datoteka. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`.

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Napomene. Svako rješenje 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 **obavezno** pišite postupak!

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, te razne tablice, papiri i sl. nisu dozvoljeni! **Mobitele** isključite i pospremite daleko od sebe! Ako se ustanovi da **kod sebe** imate mobitel za vrijeme kolokvija, kolokvij se poništava i pokreće se stegovni postupak protiv vas.

Zadatak 1 (16 bodova) Napišite program koji kao prvi argument komandne linije prima nenegativan cijeli broj, a kao drugi argument nenegativni realni broj zapisan u znanstvenoj notaciji (s eksponentom, kao kod formata %e ili %E). Program treba spremi učitane brojeve u odgovarajuće varijable tipa `int` i `double`, te ispisati njihove vrijednosti. Pretvorba za oba broja se vrši pozivom funkcije **pretvori** koja kao argumente prima string reprezentaciju učitano broja, a preko varijabilnog argumenta vraća `int` ili `double` reprezentaciju tog broja (ovisno o ulaznom broju). Napišite sve dodatne funkcije i definirajte formalne argumente funkcije **pretvori** (može ih biti i više od nabrojanih) da bi mogli ostvariti navedenu funkcionalnost. Obavezno napišite implementaciju funkcije **main**. Pripazite da izbjegnute upozorenja prevodioca uzrokovana konverzijama tipova argumenata funkcija.

Napomene: zabranjeno korištenje funkcija `atof`, `atoi`, `atol`, `strtod`, `strtol`, `strtoul`. Dopušteno je korištenje ostalih funkcija definiranih u datotekama zaglavlja `stdlib.h` i `ctype.h`. Pretpostavite da je ulazni realni broj oblika cjelobrojnidioe(ili E)eksponent (npr. 12e3, 1234E56). Eksponent broja je u ovom zadatku uvijek nenegativan cijeli broj.

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 2 (2+12 bodova) Nakon održanog turnira društvenih igara, organizatori su odlučili napraviti rang ljestvicu igrača koji su sudjelovali na turniru. Svaka igra koja se igrala na turniru bila je igra za 3 igrača.

Za svakog igrača na turniru znamo njegovo ime koje ga jedinstveno određuje (string duljine najviše 30) i ukupan broj ostvarenih bodova na turniru (nenegativan cijeli broj). Podatke o igraču pamtimo u strukturi `player`.

Nadalje, za svaku igru koja se igrala na turniru znamo njezin naziv (string duljine najviše 50), težinu (prirodan broj između 1 i 10, uključivo) te niz ekipa od 3 igrača koji su odigrali partiju te igre (niz od 3 strukture `player`). Podatke o igri pamtimo u strukturi `game`. Poznato je da je niz ekipa **sortiran silazno** prema broju ostvarenih bodova u odigranoj partiji. Ako sa **tezina** označimo težinu igre, broj ostvarenih bodova u toj partiji igrača na indeksu `i` računamo prema formuli $tezina * (3 - i)$.

(a) Deklarirajte tip podatka `player` koji pamti navedene podatke o igraču. Dodatno, varijabla `x` tog tipa se treba moći definirati sa `player x`;. Deklarirajte tip podatka `game` koji pamti navedene podatke o igri. Dodatno, varijabla `y` tog tipa se treba moći definirati sa `game y`;

(b) Napišite funkciju

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

koja kao argumente dobiva niz odigranih igara na turniru u nizu `odigrane`, niz svih igrača koji su sudjelovali na turniru u nizu `igraci` (s početnim bodovima postavljenim na 0) te prirodne brojeve `m` i `n` koji su redom duljine nizova `odigrane` i `igraci`. Funkcija treba sortirati niz `igraci` prema ukupnim bodovima igrača ostvarenim na turniru i na ekran ispisati rang listu igrača u formatu:

```
redni broj.ime igrača.(broj bodova)\n
```

Ako postoje igrači s istim brojem bodova, poredak među njima nije bitan, ali im se ispisuje isti redni broj.

Napomena. U nizu `odigrane` je moguće imati više od jedne strukture `game` s istim nazivom i težinom igre. Nije dozvoljeno korištenje dodatnih nizova.

Primjer. Pokažimo na primjeru kako ažuriramo bodove igrača u nizu `igraci`. Za niz igrača

```
igraci: P1={Vedran, 0}, P2={Nikola, 0}, P3={Matej, 0}, P4={Ana, 0}, P5={Ema, 0}, P6={Laura, 0}
```

ako je niz `odigrane` jednak

```
{Carcassonne, 6, [P3,P1,P2]}, {Monopoly, 3, [P6,P5,P4]}, {Carcassonne, 6, [P5,P2,P1]},
```

nakon poziva funkcije `rang_lista(odigrane,igraci,3,6)`, niz `igraci` treba biti (do na poredak igrača s istim bodovima):

```
igraci: P5={Ema, 24} P2={Nikola, 18}, P3={Matej, 18}, P1={Vedran, 18}, P6={Laura, 9}, P4={Ana, 3},
```

a ispis:

1. Ema (24)
2. Nikola (18)
2. Matej (18)
2. Vedran (18)
5. Laura (9)
6. Ana (3)

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 3 (15 bodova) Svi automobili dostupni za prodaju u nekoj firmi nalaze se u vezanoj listi. Za svaki automobil pamtimo naziv (string s najviše 20 znakova), cijenu (u prirodnim brojevima) i godinu proizvodnje.

- (a) Napišite deklaraciju odgovarajućeg tipa podatka za automobil na način da bude moguće definirati varijablu naredbom `auto a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `auto* izbor(auto* auti, char** nazivi, int n, int maxcijena, int godina)` koja prima pokazivač na listu automobila, niz stringova u kojem se nalaze nazivi automobila, duljinu tog niza te prirodne brojeve koji predstavljaju cijenu i godinu proizvodnje. Funkcija kreira novu vezanu listu u kojoj se nalaze automobili iz vezane liste `auti` čiji se naziv nalazi u nizu `nazivi`, a čija je cijena manja ili jednaka od `maxcijena` i godina proizvodnje veća ili jednaka od `godina`.
- (c) Napišite funkciju `auto* invertiraj(auto* izbor)` koja vraća pokazivač na invertiranu vezanu listu liste `izbor`.

Napomena. Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**. Dozvoljeno je korištenje funkcija iz `stdlib.h` i `string.h`.

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Zadatak 4 (2+10 bodova)

Zapis o kupovini nekog proizvoda u voćarni se čuva u strukturi `voce` koja se sastoji od sljedećih podataka: ime voća (string od najviše 50 znakova), cijena voća (double) i težina kupljenog voća (double). Podaci o svim kupovinama se čuvaju u binarnoj datoteci. Isti proizvod se može pojaviti više puta u datoteci. Možete pretpostaviti da trgovina može imati maksimalno 50 različitih vrsta voća. Ne moraju se svi proizvodi pojaviti u datoteci.

- Definirajte strukturu `voce` koja pamti navedene podatke o jednom proizvodu. Struktura mora biti definirana tako da bude moguća deklaracija oblika `voce v`; te smije sadržavati samo zadane podatke.
- Napišite funkciju `void obradiPodatke(FILE* in, FILE* out)` koja kao argumente prima pokazivač `in` na binarnu datoteku otvorenu za čitanje te pokazivač `out` na tekstualnu datoteku otvorenu za pisanje. Funkcija u tekstualnu datoteku upisuje ukupnu težinu kupljenih proizvoda iz binarne datoteke u sljedećem obliku "ime: cijena, ukupnaTezina". Nakon svakog proizvoda ide skok u novi red. Redoslijed proizvoda nije bitan. Možete pretpostaviti da binarna datoteka nije prazna.

Primjer:

Početna binarna datoteka `in`:

```
{Jabuka, 5.2, 3.5}, {Banana, 8.7, 1.2}, {Jagoda, 10, 0.7}, {Jabuka, 5.2, 1.2},  
{Jabuka, 5.2, 2}, {Banana, 8.7, 0.8}
```

Završna tekstualna datoteka `out`:

```
Jabuka: 5.2, 6.7  
Banana: 8.7, 2.0  
Jagoda: 10, 0.7
```

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje jednog niza struktura `voce` duljine 50**. Nije dozvoljeno korištenje dodatnih datoteka. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`.