

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Upute: Na skraćenom ispitu 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.

Zadatak 1 (12 bodova) Napišite program koji učitava nazive ulazne i izlazne datoteke te dva stringa, **stari** i **novi**. Program u ulaznoj datoteci zamjenjuje sva pojavljivanja stringa **stari** stringom **novi** i zapisuje rezultat u izlaznu datoteku. Smijete pretpostaviti da string **stari** ne sadrži znak '**\n**' (može sadržavati razmake) te da su sve linije u ulaznoj datoteci kraće od 100 znakova. **Za dodatnih 5 bodova**, neka program prima nazive ulazne i izlazne datoteke te dva stringa kao argumente komandne linije.

Primjer:

stari="taubeka"

novi="goluba"

Ulazna.txt

Bledi je mesec hodil po krovu,

A ja sem popeval popevku svoju:

Kak taubeka dva mi srećni smo bili,

Z jednoga peharčeka ljubav smo pili.

Kak taubeka dva mi srećni smo bili,

Z jednoga peharčeka ljubav smo pili

Izlazna.txt

Bledi je mesec hodil po krovu,

A ja sem popeval popevku svoju:

Kak goluba dva mi srećni smo bili,

Z jednoga peharčeka ljubav smo pili.

Kak goluba dva mi srećni smo bili,

Z jednoga peharčeka ljubav smo pili

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 2 (2+6+6 bodova) Na jednom sveučilištu studenti imaju pravo na subvencioniranu prehranu. Podaci o subvenciji nalaze se na tzv. "iksici", koja sadrži identifikacijski broj studenta - JMBAG (string duljine 10) te iznos subvencije za troškove prehrane na koji student ima pravo (nenegativan realan broj). Prilikom korištenja iksice, 70% iznosa računa podmiruje se sredstvima s iksice (tada se sredstva na iksici umanjuju za taj iskorišteni iznos). Jelovnik u restoranu predstavljen je kao niz jela, pri čemu svako jelo ima svoje ime (string duljine najviše 30) te cijenu (pozitivan realan broj). Svaki restoran bilježi promet tijekom jednog dana u obliku niza računa. Račun sadrži broj odabranih jela (prirodan broj), popis odabranih jela (niz jela), vrijeme kupnje zapisano u formatu `hh:mm` (string duljine 5) te ukupan iznos računa (nenegativan realan broj).

(a) Definirajte tipove `iksica`, `jelo` i `racun`, tako da bude moguća deklaracija `iksica x`, `jelo j` i `racun r`.

(b) Napišite funkciju

```
racun *provuci_iksicu(iksica *x, char vrijeme[], jelo jelovnik[], int m, char* odabir[], int k)
```

koja prima pokazivač na iksicu, string duljine 5 u kojem je zapisano vrijeme u formatu `hh:mm`, jelovnik duljine m te niz stringova duljine k . Ti stringovi predstavljaju imena odabranih jela s jelovnika. Funkcija izrađuje novi račun i vraća pokazivač na njega te ispisuje: **Za platiti: x** gdje je x iznos koji **student treba platiti** (nakon uračunate subvencije). Funkcija također treba smanjiti iznos subvencije na iksici za 70% iznosa računa, kako je prethodno opisano. Ako student **nema dovoljno sredstava** na iksici, funkcija vraća `NULL` i ispisuje: **ODBIJENO**

(c) Napišite funkciju `void vrijeme_sort(racun restoran[], int n)` koja prima niz računa restorana duljine n tijekom jednog dana i sortira ga uzlazno prema vremenu izdavanja.

Napomene:

- Zabranjeno je korištenje dodatnih nizova. U (b) dijelu zadatku možete alocirati memoriju samo za potrebu izrade računa. Ne treba provjeravati uspješnost (re)alokacije. Uz standardnu biblioteku `stdio.h`, dozvoljeno je korištenje biblioteke `stdlib.h` i `string.h`.
- Podzadatak (c) nosi **6** bodova, međutim ukoliko se umjesto algoritma sortiranja složenosti $\mathcal{O}(n \log n)$ koriste algoritmi složenosti $\mathcal{O}(n^2)$, zadatak nosi maksimalno **2** bodova. Korištenje `quickSort` algoritma iz biblioteke `stdlib.h` nosi dodatnih **4** boda, odnosno tako riješen zadatak u sumi nosi **10** bodova.

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 3 (2+4+6 bodova) Sve pjesme koje pripadaju popisu za reproduciranje (playlisti) su spremljene u vezanoj listi. Za svaku pjesmu pamtim naziv (string s najviše 20 znakova), izvođača (string s najviše 30 znakova), trajanje u sekundama (cijeli broj) i je li označena kao omiljena (cijeli broj koji ima vrijednost 1 ako je i 0 ako nije).

- (a) Napišite deklaraciju odgovarajućeg tipa podataka za pjesmu na način da bude moguće definirati varijablu naredbom `pjesma p;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `pjesma* izdvoji_omiljene(pjesma* playlist)` koja prima pokazivač na listu pjesama. Funkcija stvara novu vezanu listu koja sadrži sve pjesme iz liste `playlist` koje su označene kao omiljene i vraća pokazivač na početak novostvorene liste, a ako nema takvih pjesama onda vraća `NULL`. Za elemente nove liste je potrebno alocirati memoriju, te izvorna lista `playlist` ne smije biti promijenjena.
- (c) Napišite funkciju `void premjesti_kratke(pjesma** playlist, pjesma** playlist_kratke, int max_sec)`. Funkcija prima dvostruki pokazivač na popis pjesama `playlist`, dvostruki pokazivač `playlist_kratke` preko kojeg će se vratiti lista kratkih pjesama, te `max_sec` koji definira maksimalno trajanje u sekundama za koje se pjesma smatra *kratkom* (kratka je ako traje $\leq \text{max_sec}$). Lista na koju pokazuje `*playlist_kratke` inicijalno je prazna (`NULL`) i funkcija je popunjava. Funkcija treba premjestiti sve kratke pjesme iz liste `*playlist` u listu na koju pokazuje `*playlist_kratke`. Premještanje treba izvršiti **bez alociranja dodatne memorije**, tj. preslagivanjem postojećih čvorova.

Napomena:

Dozvoljeno je korištenje pomoćnih funkcija. Nije dozvoljeno korištenje pomoćnih nizova. Možete pretpostaviti da su sve ulazne liste ispravno formirane. Jedine dozvoljene biblioteke u ovom zadatku su `stdlib.h` i `string.h`.

Primjer:

- `moja_playlista -> {"HeyJude", "TheBeatles", 431, 1} -> {"Stairway", "LedZeppelin", 482, 0} -> {"Imagine", "JohnLennon", 183, 1} -> {"Yesterday", "TheBeatles", 125, 0} -> {"LightMyFire", "TheDoors", 428, 1} -> NULL`

```
pjesma* omiljene = izdvoji_omiljene(moja_playlista);
```

- `omiljene->{"HeyJude", "TheBeatles", 431, 1} -> {"Imagine", "JohnLennon", 183, 1} -> {"LightMyFire", "TheDoors", 428, 1} -> NULL`

```
premjesti_kratke(&moja_playlista, &lista_kratkih, 200);
```

- `lista_kratkih -> {"Imagine", "JohnLennon", 183, 1} -> {"Yesterday", "TheBeatles", 125, 0} -> NULL`
- `moja_playlista -> {"HeyJude", "TheBeatles", 431, 1} -> {"Stairway", "LedZeppelin", 482, 0} -> {"LightMyFire", "TheDoors", 428, 1} -> NULL`

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 4 (5+8 bodova) Nakon sajma čokolada, posjetitelji su isprobanim čokoladama dodjeljivali ocjene od 1 do 5. Podaci o ocjenjenoj čokoladi su spremljeni u strukturi `cokolada` koja sadrži informacije o udjelu kakaa (cjelobrojna varijabla) koji može biti broj od 0 do 100 uključivo, dodatku (string od najviše 20 znakova) koji može biti točno jedan od sljedećih stringova: `ljesnjak`, `badem`, `keks` te dodijeljenoj ocjeni (cjelobrojna varijabla). Udio kakaa i dodatak **jedinstveno** određuju čokoladu.

Jedna je tvornica čokolade spremila rezultate anketiranja u binarnu datoteku `anketa.bin` u obliku niza struktura tipa `cokolada`.

- (a) Definirajte strukturu `cokolada` čiji su podaci opisani ranije. Struktura mora biti definirana tako da bude moguća deklaracija oblika `cokolada c`; te smije sadržavati samo zadane podatke.
- (b) Napišite implementaciju funkcije s prototipom

```
void najdraza(FILE* in, FILE* out)
```

koja kao argumente prima pokazivače na datoteke. Pri tome, `in` je pokazivač na binarnu datoteku `anketa.bin` otvorenu za čitanje, a `out` je pokazivač na tekstualnu datoteku otvorenu za pisanje. Funkcija treba u datoteku na koju pokazuje `out` upisati maksimalnu prosječnu ocjenu ostvarenu među svim čokoladama. Preciznije, prosječnu ocjenu čokolade računamo kao omjer sume ocjena za tu čokoladu kroz broj ocjena za tu čokoladu.

Primjer:

Za binarnu datoteku `anketa.bin` sa podacima:

```
{ljesnjak,40,5}, {keks,30,4}, {ljesnjak,60,3}, {ljesnjak,40,3}, {badem,80,2}, {ljesnjak,60,5}
```

maksimalna prosječna ocjena je 4 (ostvarena za čokolade `ljesnjak,40`, `ljesnjak,60` i `keks,30`) pa u tekstualnu datoteku na koju pokazuje `out` treba upisati 4.000 (nije bitno na koliko decimala).

Napomena: Nije dozvoljeno korištenje pomoćnih polja ili datoteka. Dozvoljeno je korištenje pomoćnih varijabli tipa `cokolada`, kao i implementacija pomoćnih funkcija.

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Upute: Na skraćenom ispitu 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.

Zadatak 1 (12 bodova) Napišite program koji učitava nazive ulazne i izlazne datoteke te dva stringa, `old` i `new`. Program u ulaznoj datoteci zamjenjuje sva pojavljivanja stringa `old` stringom `new` i zapisuje rezultat u izlaznu datoteku. Smijete pretpostaviti da string `old` ne sadrži znak `'\n'` (može sadržavati razmake) te da su sve linije u ulaznoj datoteci kraće od 100 znakova. **Za dodatnih 5 bodova**, neka program prima nazive ulazne i izlazne datoteke te dva stringa kao argumente komandne linije.

Primjer:

`old="peharčeka"`

`new="vrča"`

`Ulazna.txt`

Bledi se mesec za oblak skrival,
A ja sem s popevkom dragu dozival:
Kak taubeka dva mi srečni smo bili,
Z jednoga peharčeka ljubav smo pili.
Kak taubeka dva mi srečni smo bili,
Z jednoga peharčeka ljubav smo pili.

`Izlazna.txt`

Bledi se mesec za oblak skrival,
A ja sem s popevkom dragu dozival:
Kak taubeka dva mi srečni smo bili,
Z jednoga vrča ljubav smo pili.
Kak taubeka dva mi srečni smo bili,
Z jednoga vrča ljubav smo pili.

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 2 (2+6+6 bodova)

Na jednom sveučilištu studenti imaju pravo na subvencioniranu prehranu. Podaci o subvenciji nalaze se na tzv. "iksici", koja sadrži identifikacijski broj studenta - JMBAG (string duljine 10) te iznos subvencije za troškove prehrane na koji student ima pravo (nenegativan realan broj). Prilikom korištenja iksice, 70% iznosa računa podmiruje se sredstvima s iksice (tada se sredstva na iksici umanjuju za taj iskorišteni iznos). Jelovnik u restoranu predstavljen je kao niz jela, pri čemu svako jelo ima svoje ime (string duljine najviše 30) te cijenu (pozitivan realan broj). Svaki restoran bilježi promet tijekom jednog dana u obliku niza računa. Račun sadrži broj odabranih jela (prirodan broj), popis odabranih jela (niz jela), vrijeme kupnje zapisano u formatu `hh:mm` (string duljine 5) te ukupan iznos računa (nenegativan realan broj).

- (a) Definirajte tipove `iksica`, `jelo` i `racun`, tako da bude moguća deklaracija `iksica x`, `jelo y` i `racun z`.
- (b) Napišite funkciju

```
racun *provuci_iksicu(iksica *x, char time[], jelo jelovnik[], int j, char* odabir[], int r)
```

koja prima pokazivač na `iksicu`, string duljine 5 u kojem je zapisano vrijeme u formatu `hh:mm`, jelovnik duljine `j` te niz stringova duljine `r`. Ti stringovi predstavljaju imena odabranih jela s jelovnika. Funkcija izrađuje novi račun i vraća pokazivač na njega te ispisuje: **Za platiti: x** gdje je `x` iznos koji **student treba platiti** (nakon uračunate subvencije). Funkcija također treba smanjiti iznos subvencije na `iksici` za 70% iznosa računa, kako je prethodno opisano. Ako student **nema dovoljno sredstava** na `iksici`, funkcija vraća `NULL` i ispisuje: **ODBIJENO**.

- (c) Napišite funkciju `void time_sort(racun menza[], int m)` koja prima niz računa restorana duljine `m` tijekom jednog dana i sortira ga silazno prema vremenu izdavanja.

Napomene:

- Zabranjeno je korištenje dodatnih nizova. U (b) dijelu zadatku možete alocirati memoriju samo za potrebu izrade računa. Ne treba provjeravati uspješnost (re)alokacije. Uz standardnu biblioteku `stdio.h`, dozvoljeno je korištenje biblioteke `stdlib.h` i `string.h`.
- Podzadatak (c) nosi **6** bodova, međutim ukoliko se umjesto algoritma sortiranja složenosti $\mathcal{O}(n \log n)$ koriste algoritmi složenosti $\mathcal{O}(n^2)$, zadatak nosi maksimalno **2** bodova. Korištenje `quickSort` algoritma iz biblioteke `stdlib.h` nosi dodatnih **4** boda, odnosno tako riješen zadatak u sumi nosi **10** bodova.

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 3 (2+4+6 bodova) Sve pjesme koje pripadaju popisu za reproduciranje (playlisti) su spremljene u vezanoj listi. Za svaku pjesmu pamtim naziv (string s najviše 20 znakova), izvođača (string s najviše 30 znakova), trajanje u sekundama (cijeli broj) i je li označena kao omiljena (cijeli broj koji ima vrijednost 1 ako je i 0 ako nije).

- (a) Napišite deklaraciju odgovarajućeg tipa podataka za pjesmu na način da bude moguće definirati varijablu naredbom `pjesma p;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `pjesma* izdvoji_neomiljene(pjesma* playlist)` koja prima pokazivač na listu pjesama. Funkcija stvara novu vezanu listu koja sadrži sve pjesme iz liste `playlist` koje nisu označene kao omiljene i vraća pokazivač na početak novostvorene liste, a ako nema takvih pjesama onda vraća `NULL`. Za elemente nove liste je potrebno alocirati memoriju, te izvorna lista `playlist` ne smije biti promijenjena.
- (c) Napišite funkciju `void premjesti_duge(pjesma** playlist, pjesma** playlist_duge, int min_sec)`. Funkcija prima dvostruki pokazivač na popis pjesama `playlist`, dvostruki pokazivač `playlist_duge` preko kojeg će se vratiti lista dugih pjesama, te `min_sec` koji definira minimalno trajanje u sekundama za koje se pjesma smatra *dugom* (duga je ako traje $\geq \text{min_sec}$). Lista na koju pokazuje `*playlist_duge` inicijalno je prazna (`NULL`) i funkcija je popunjava.

Funkcija treba premjestiti sve duge pjesme iz liste `*playlist` u listu na koju pokazuje `*playlist_duge`. Premještanje treba izvršiti **bez alociranja dodatne memorije**, tj. preslagivanjem postojećih čvorova.

Napomena:

Dozvoljeno je korištenje pomoćnih funkcija. Nije dozvoljeno korištenje pomoćnih nizova. Možete pretpostaviti da su sve ulazne liste ispravno formirane. Jedine dozvoljene biblioteke u ovom zadatku su `stdlib.h` i `string.h`.

Primjer:

```
• moja_playlista -> {"HeyJude", "TheBeatles", 431, 1} -> {"Stairway", "LedZeppelin", 482, 0} ->
  {"Imagine", "JohnLennon", 183, 1} -> {"Yesterday", "TheBeatles", 125, 0} ->
  {"LightMyFire", "TheDoors", 428, 1} -> NULL
```

```
pjesma* neomiljene = izdvoji_neomiljene(moja_playlista);
```

```
• neomiljene -> {"Stairway", "LedZeppelin", 482, 0} -> {"Yesterday", "TheBeatles", 125, 0} -> NULL
```

```
premjesti_duge(&moja_playlista, &lista_dugih, 200);
```

```
• lista_dugih -> {"HeyJude", "TheBeatles", 431, 1} -> {"Stairway", "LedZeppelin", 482, 0} ->
  {"LightMyFire", "TheDoors", 428, 1} -> NULL
```

```
• moja_playlista -> {"Imagine", "JohnLennon", 183, 1} -> {"Yesterday", "TheBeatles", 125, 0} -> NULL
```

Programiranje 2 – drugi kolokvij, 27. 6. 2025.

Zadatak 4 (5+8 bodova) Nakon sajma slastica, posjetitelji su isprobanim slasticama dodjeljivali ocjene od 1 do 5. Podaci o ocjenjenoj slastici su spremljeni u strukturi `slastica` koja sadrži informacije o udjelu čokolade (cjelobrojna varijabla) koji može biti broj od 0 do 100 uključivo, dodatku (string od najviše 20 znakova) koji može biti točno jedan od sljedećih stringova: `ljesnjak`, `badem`, `keks` te dodijeljenoj ocjeni (cjelobrojna varijabla). Udio čokolade i dodatak **jedinstveno** određuju slasticu.

Jedna je tvornica slastica spremila rezultate anketiranja u binarnu datoteku `anketa.bin` u obliku niza struktura tipa `slastica`.

- (a) Definirajte strukturu `slastica` čiji su podaci opisani ranije. Struktura mora biti definirana tako da bude moguća deklaracija oblika `slastica s`; te smije sadržavati samo zadane podatke.
- (b) Napišite implementaciju funkcije s prototipom

```
void najdraza(FILE* in, FILE* out)
```

koja kao argumente prima pokazivače na datoteke. Pri tome, `in` je pokazivač na binarnu datoteku `anketa.bin` otvorenu za čitanje, a `out` je pokazivač na tekstualnu datoteku otvorenu za pisanje. Funkcija treba u datoteku na koju pokazuje `out` upisati maksimalnu prosječnu ocjenu ostvarenu među svim slasticama. Preciznije, prosječnu ocjenu slastice računamo kao omjer suma ocjena za tu slasticu kroz broj ocjena za tu slasticu.

Primjer: Za binarnu datoteku `anketa.bin` sa podacima:

```
{ljesnjak,40,5}, {keks,30,4}, {ljesnjak,60,3}, {ljesnjak,40,3}, {badem,80,2}, {ljesnjak,60,5}
```

maksimalna prosječna ocjena je 4 (ostvarena za slastice `ljesnjak,40`, `ljesnjak,60` i `keks,30`) pa u tekstualnu datoteku na koju pokazuje `out` treba upisati 4.000 (nije bitno na koliko decimala).

Napomena: Nije dozvoljeno korištenje pomoćnih polja ili datoteka. Dozvoljeno je korištenje pomoćnih varijabli tipa `slastica`, kao i implementacija pomoćnih funkcija.