

Programiranje 2 – 1. pismeni ispit, 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 (25 bodova) Neka je dano cjelobrojno dvodimenzionalno polje naziva **Visina**, koje se sastoji od m redaka i n stupaca te je ispunjeno različitim nenegativnim vrijednostima. Napišite funkciju koja najprije dinamički alocira novo dvodimenzionalno polje **Rezultat** istih dimenzija te postavlja njegove vrijednosti identično kao u polju **Visina** (kreira novi objekt koji je kopija staroga). Funkcija, uz originalno polje i dimenzije polja, prima i indekse $i \in \{0, \dots, m-1\}$, $j \in \{0, \dots, n-1\}$ te zatim mijenja vrijednosti u polju **Rezultat** tako da, počevši od pozicije **Rezultat**[i][j], „poplavljuje“ sve susjedne ćelije (gore, dolje, lijevo, desno) koje imaju istu vrijednost kao početna ćelija. Vrijednosti svih „poplavljenih“ ćelija pritom se zamjenjuju s -1 . Funkcija vraća pokazivač na dvodimenzionalno polje **Rezultat**. Napišite pomoćnu funkciju koja oslobađa memoriju zauzetu od funkcije **Flood**.

Primjer:

Početna pozicija: $(i, j) = (2, 1)$

Visina

0	1	1	0	2	2
1	0	1	0	2	2
0	0	0	0	1	1
2	0	1	1	1	0
2	2	1	0	0	0

Rezultat

0	1	1	-1	2	2
1	-1	1	-1	2	2
-1	-1	-1	-1	1	1
2	-1	1	1	1	0
2	2	1	0	0	0

Programiranje 2 – 1. pismeni ispit, 27. 6. 2025.

Zadatak 2 (25 bodova) Napišite implementaciju funkcije s prototipom

```
int involutorna(FILE* in)
```

koja kao argument prima pokazivač na tekstualnu datoteku otvorenu za čitanje u kojoj je zapisana kvadratna matrica sa cjelobrojnim koeficijentima. Matrica je zapisana redak po redak u linijama datoteke te su elementi retka odvojeni razmacima. Funkcija vraća 1 ako je matrica iz datoteke na koju pokazuje `in` involutorna, a inače vraća 0. Podsjetimo se, matrica je involutorna ako je sama sebi inverz.

Primjer: Za pokazivač `in` koji pokazuje na tekstualnu datoteku sa sadržajem

```
2 -1
```

```
3 -2
```

funkcija `involutorna` treba vratiti 1 jer vrijedi

$$\begin{pmatrix} 2 & -1 \\ 3 & -2 \end{pmatrix} \cdot \begin{pmatrix} 2 & -1 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Programiranje 2 – 1. pismeni ispit, 27. 6. 2025.

Zadatak 3 (5+10+10 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 izbirljivi(racun restoran[], int n)` koja prima niz računa restorana duljine n tijekom jednog dana i sortira ga uzlazno prema vremenu izdavanja. Složenost sortiranja mora biti $\mathcal{O}(n \log n)$, ako je složenost $\mathcal{O}(n^2)$ ili veća nosi podzadatak nosi 0 bodova.

Napomene:

Zabranjeno je korištenje dodatnih nizova. U (b) zadatku možete alocirati memoriju samo za potrebu izrade računa. Ne treba provjeravati uspješnost (re)alokacije. Uz standardnu biblioteku `stdio.h`, u ovome zadatku dozvoljeno je korištenje biblioteke `stdlib.h` i `string.h`.

Programiranje 2 – 1. pismeni ispit, 27. 6. 2025.

Zadatak 4 (4+9+12 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 $\leq$ `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. Redoslijed pjesama u listi `*playlist_kratke` mora biti isti kao u izvornoj listi.

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`