

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Rezultati i uvidi: Rezultati u četvrtak, 25.6., 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.

Zadatak 1. (30 bodova) Za broj $m \in \mathbb{N}$ kažemo da je *zanimljiv* ako postoji $k \in \mathbb{N}$, $k > 1$, tako da vrijedi $m = 2^k$ ili $m = 7^k$. Napišite rekurzivnu funkciju koja prima prirodni broj n i vraća broj načina na koji se n može napisati kao suma zanimljivih brojeva. Broj pribrojnika u sumi može biti proizvoljan (uključivo i samo jedan), a rastave s permutiranim pribrojnicima smatramo istima, tj. poredak pribrojnika nije bitan. Osim broja n , funkcija smije imati i dodatne parametre. Obavezno napišite i primjer poziva funkcije.

Na primjer, za $n = 8$, funkcija treba vratiti 2, jer je $8 = 2^3 = 2^2 + 2^2$. Za $n = 53$, funkcija treba vratiti 1, jer je $53 = 2^2 + 7^2$, a rastave $53 = 2^2 + 7^2$ i $53 = 7^2 + 2^2$ smatramo istima.

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 2. (30 bodova) Svi racionalni brojevi iz intervala $\langle 0, 1 \rangle$ imaju ili konačan ili periodički decimalni zapis. Takve zapise reprezentiramo nepraznom vezanom listom znamenki koje dolaze iza decimalne točke. Ako je zapis konačan, poput 0.54122, onda je prikaz obična vezana lista — u elementu koji sadrži zadnju znamenku 2, pokazivač na sljedeći element liste je jednak NULL (lijeva slika). Ako je zapis periodički, poput 0.54122122122... (s periodom 122), onda pripadna lista sadrži samo znamenke do kraja prvog perioda, a pokazivač u elementu koji sadrži zadnju znamenku 2, pokazuje na **element** koji sadrži **prvu** znamenku perioda. Takvu listu zovemo **omča** (desna slika). Prazna lista je prikaz broja 0.



Napišite deklaraciju tipa za odgovarajući element liste.

- Napišite funkciju koja prima pokazivač na ovakvu listu. Ako je ta lista omča, funkcija treba vratiti 1, a kroz varijabilne argumente treba vratiti pokazivače na prvu i zadnju znamenku perioda. U protivnom, funkcija treba vratiti 0, a varijabilne argumente treba postaviti na NULL.
- Napišite funkciju koja prima pokazivač na ovakvu listu i varijabilne pokazivače na prvu i zadnju znamenku perioda (kao u (a) dijelu). Funkcija treba obrisati sve parne znamenke unutar perioda (ako se radi o omči), vratiti pokazivač na prvi element dobivene liste i ažurirati pripadne varijabilne pokazivače na prvu i zadnju znamenku perioda. Funkcija treba ispisati decimalni zapis broja dobivenog nakon eventualnog brisanja: prvo dolazi 0 i točka, onda znamenke ispred perioda, a za omču, period mora biti ispisan unutar uglatih zagrada. Primjerice, za ulaznu listu s lijeve slike, ispis je 0.54122, a za ulaznu listu s desne slike, ispis (nakon brisanja) je 0.54[1].

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 3. (20 + 10 = 30 bodova)

(a) Napišite funkciju `promijeniA` koja prima dvodimenzionalno polje znakova M , dimenzija $n \times n$, i cijeli broj $k \geq 0$. Ako je n paran broj, funkcija treba vratiti novo polje koje nastaje iz M zadanim nizom transformacija, a staro polje mora ostati netaknuto. Ako ulazno polje nema parnu dimenziju, funkcija mora ispisati da se dogodila greška i kao novo polje vratiti NULL. Transformacije polja M su, redom:

- 1) Za svaki par indeksa $i, j \in \{1, \dots, n\}$, $i \neq j$, ako se na mjestima (i, j) i (j, i) u polju M nalaze znamenke, onda treba na oba mjesta upisati njihov zbroj modulo 10, tj. $(M_{i,j} + M_{j,i}) \% 10$.
- 2) Zatim se u novodobivenom polju zamijene 1. i 2. redak, 3. i 4. redak, ..., $(n-1)$ -i i n -ti redak.
- 3) U dobivenom polju, k -ta donja glavna dijagonala se preokreće, a u k -toj gornjoj dijagonali se sva velika slova pretvaraju u mala. ($k=0$ označava glavnu dijagonalu i pritom treba napraviti oba postupka.)

Npr. za $k=1$ i sljedeće polje M , dimenzija 4×4 , imamo

$$M = \begin{bmatrix} a & 2 & b & c \\ 3 & D & e & f \\ 4 & G & H & p \\ v & w & q & 8 \end{bmatrix} \xrightarrow{1)} \begin{bmatrix} a & 5 & b & c \\ 5 & D & e & f \\ 4 & G & H & p \\ v & w & q & 8 \end{bmatrix} \xrightarrow{2)} \begin{bmatrix} 5 & D & e & f \\ a & 5 & b & c \\ v & w & q & 8 \\ 4 & G & H & p \end{bmatrix} \xrightarrow{3)} \begin{bmatrix} 5 & d & e & f \\ H & 5 & b & c \\ v & w & q & 8 \\ 4 & G & a & p \end{bmatrix}.$$

(b) Napišite funkciju `int brojA(char **M, int m, int n, char *str)` koja u dvodimenzionalnom polju M , dimenzija $m \times n$, treba izbrojati sve pojave stringa `str` odozdo nagore (\uparrow). Npr. za string `str = "DEF"` i za polje

$$M = \begin{bmatrix} a & a & F & a & D \\ a & a & E & a & E \\ F & a & D & E & F \\ E & a & E & a & a \\ D & a & a & a & a \end{bmatrix}$$

funkcija treba vratiti vrijednost 2.

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 4. (30 bodova) Na kolegiju je upisano ≤ 99 studenata. Svaki student ima svoj jedinstveni matični broj između 1 i 99. U tekstualnim datotekama `prvi_kolokvij.txt` i `drugi_kolokvij.txt` zapisani su ostvareni bodovi za studente koji su pristupili pojedinom kolokviju. Obje datoteke imaju isti oblik. U prvom redu zapisan je broj studenata n_i koji su pristupili i -tom kolokviju. Nakon tog retka, dolazi još točno n_i redaka oblika

```
matbr bodovi_1 bodovi_2
```

gdje je `matbr` matični broj studenta, a `bodovi_1` i `bodovi_2` su bodovi (nenegativni cijeli brojevi) koje je taj student ostvario na prvom, odnosno drugom zadatku tog kolokvija. Reciprojedinie datoteke nisu sortirani, ali se svaki student u njoj javlja najviše jednom.

Za svakog studenta koji se javlja u barem jednoj od te dvije datoteke, potrebno je stvoriti jedinstvenu tekstualnu datoteku `matbr.txt` (npr. `17.txt`) u kojoj piše PAD ako je suma svih njegovih bodova na svim kolokvijima strogo manja od prosjeka sumi svih bodova svih studenata na svim kolokvijima, a inače POLOZENO.

U sljedećem primjeru, suma svih bodova svih studenata je 500 pa je prosjek po studentu $\frac{500}{5} = 100$ bodova. Dakle, svi studenti sa 100 ili više bodova su položili kolegij.

<code>prvi_kolokvij.txt:</code>	<code>drugi_kolokvij.txt:</code>	Izlazne datoteke:
4	2	<code>1.txt:</code> POLOZENO
98 51 50	29 25 26	<code>17.txt:</code> PAD
1 100 0	99 40 60	<code>29.txt:</code> POLOZENO
29 24 25		<code>98.txt:</code> POLOZENO
17 50 49		<code>99.txt:</code> POLOZENO

U rješenju napišite samo dio programa koji radi opisani posao (tj. ne morate pisati `#include` direktive, funkciju `main`, itd.). Možete koristiti pomoćna polja, a ulazne datoteke možete otvarati i zatvarati proizvoljan broj puta. Ne morate provjeravati uspješnost otvaranja niti pisanja datoteka.

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Rezultati i uvidi: Rezultati u četvrtak, 25.6., navečer na webu. Termini uvida bit će objavljeni naknadno.

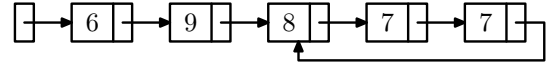
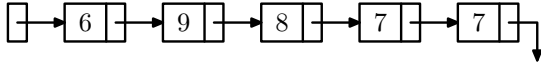
Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele isključite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno 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.

Zadatak 1. (30 bodova) Za broj $m \in \mathbb{N}$ kažemo da je *zanimljiv* ako postoji $k \in \mathbb{N}$, $k > 1$, tako da vrijedi $m = 3^k$ ili $m = 5^k$. Napišite rekurzivnu funkciju koja prima prirodni broj n i vraća broj načina na koji se n može napisati kao suma zanimljivih brojeva. Broj pribrojnika u sumi može biti proizvoljan (uključivo i samo jedan), a rastave s permutiranim pribrojnicima smatramo istima, tj. poredak pribrojnika nije bitan. Osim broja n , funkcija smije imati i dodatne parametre. Obavezno napišite i primjer poziva funkcije.

Na primjer, za $n = 27$, funkcija treba vratiti 2, jer je $27 = 3^3 = 3^2 + 3^2 + 3^2$. Za $n = 34$, funkcija treba vratiti 1, jer je $34 = 3^2 + 5^2$, a rastave $34 = 3^2 + 5^2$ i $34 = 5^2 + 3^2$ smatramo istima.

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 2. (30 bodova) Svi racionalni brojevi iz intervala $\langle 0, 1 \rangle$ imaju ili konačan ili periodički decimalni zapis. Takve zapise reprezentiramo nepraznom vezanom listom znamenki koje dolaze iza decimalne točke. Ako je zapis konačan, poput 0.69877, onda je prikaz obična vezana lista — u elementu koji sadrži zadnju znamenku 7, pokazivač na sljedeći element liste je jednak NULL (lijeva slika). Ako je zapis periodički, poput 0.69877877877... (s periodom 877), onda pripadna lista sadrži samo znamenke do kraja prvog perioda, a pokazivač u elementu koji sadrži zadnju znamenku 7, pokazuje na **element** koji sadrži **prvu** znamenku perioda. Takvu listu zovemo **omča** (desna slika). Prazna lista je prikaz broja 0.



Napišite deklaraciju tipa za odgovarajući element liste.

- Napišite funkciju koja prima pokazivač na ovakvu listu. Ako je ta lista omča, funkcija treba vratiti 1, a kroz varijabilne argumente treba vratiti pokazivače na prvu i zadnju znamenku perioda. U protivnom, funkcija treba vratiti 0, a varijabilne argumente treba postaviti na NULL.
- Napišite funkciju koja prima pokazivač na ovakvu listu i varijabilne pokazivače na prvu i zadnju znamenku perioda (kao u (a) dijelu). Funkcija treba obrisati sve neparne znamenke unutar perioda (ako se radi o omči), vratiti pokazivač na prvi element dobivene liste i ažurirati pripadne varijabilne pokazivače na prvu i zadnju znamenku perioda. Funkcija treba ispisati decimalni zapis broja dobivenog nakon eventualnog brisanja: prvo dolazi 0 i točka, onda znamenke ispred perioda, a za omču, period mora biti ispisan unutar uglatih zagrada. Primjerice, za ulaznu listu s lijeve slike, ispis je 0.69877, a za ulaznu listu s desne slike, ispis (nakon brisanja) je 0.69[8].

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 3. (20 + 10 = 30 bodova)

(a) Napišite funkciju `promijeniB` koja prima dvodimenzionalno polje znakova M , dimenzija $n \times n$, i cijeli broj $k \geq 0$. Ako je n paran broj, funkcija treba vratiti novo polje koje nastaje iz M zadanim nizom transformacija, a staro polje mora ostati netaknuto. Ako ulazno polje nema parnu dimenziju, funkcija mora ispisati da se dogodila greška i kao novo polje vratiti `NULL`. Transformacije polja M su, redom:

- 1) Za svaki par indeksa $i, j \in \{1, \dots, n\}$, $i \neq j$, ako se na mjestima (i, j) i (j, i) u polju M nalaze znamenke, onda treba na oba mjesta upisati njihov umnožak modulo 10, tj. $(M_{i,j} \cdot M_{j,i}) \% 10$.
- 2) Zatim se u novodobivenom polju zamijene 1. i 2. stupac, 3. i 4. stupac, \dots , $(n-1)$ -i i n -ti stupac.
- 3) U dobivenom polju, k -ta gornja sporedna dijagonala se preokreće, a u k -toj donjoj sporednoj dijagonali se sva mala slova pretvaraju u velika. ($k = 0$ označava glavnu sporednu dijagonalu i pritom treba napraviti oba postupka.)

Npr. za $k = 1$ i sljedeće polje M , dimenzija 4×4 , imamo

$$M = \begin{bmatrix} a & 2 & b & c \\ 3 & d & e & f \\ 4 & G & H & p \\ v & w & q & 8 \end{bmatrix} \xrightarrow{1)} \begin{bmatrix} a & 6 & b & c \\ 6 & d & e & f \\ 4 & G & H & p \\ v & w & q & 8 \end{bmatrix} \xrightarrow{2)} \begin{bmatrix} 6 & a & c & b \\ d & 6 & f & e \\ G & 4 & p & H \\ w & v & 8 & q \end{bmatrix} \xrightarrow{3)} \begin{bmatrix} 6 & a & G & b \\ d & 6 & f & E \\ c & 4 & P & H \\ w & V & 8 & q \end{bmatrix}.$$

(b) Napišite funkciju `int brojB(char **M, int m, int n, char *str)` koja u dvodimenzionalnom polju M , dimenzija $m \times n$, treba izbrojati sve pojave stringa `str` odozgo nadolje (\downarrow). Npr. za string `str = "DEF"` i za polje

$$M = \begin{bmatrix} a & a & a & a & D \\ a & a & E & a & E \\ F & a & D & E & F \\ E & a & E & a & a \\ D & a & F & a & a \end{bmatrix}$$

funkcija treba vratiti vrijednost 2.

Programiranje 2 – završna provjera znanja, 19. 6. 2020.

Zadatak 4. (30 bodova) Na kolegiju je upisano ≤ 99 studenata. Svaki student ima svoj jedinstveni matični broj između 1 i 99. U tekstualnim datotekama `prva_provjera.txt` i `druga_provjera.txt` zapisani su ostvareni bodovi za studente koji su pristupili pojedinoj provjeri. Obje datoteke imaju isti oblik. U prvom redu zapisan je broj studenata n_i koji su pristupili i -toj provjeri. Nakon tog retka, dolazi još točno $2n_i$ redaka oblika

`matbr bodovi`

gdje je `matbr` matični broj studenta, a `bodovi` su bodovi (nenegativni cijeli broj) koje je taj student ostvario na jednom od dva zadatka iz te provjere (obje provjere imaju po dva zadatka). Recimo pojedine datoteke nisu sortirane, ali se svaki student koji se uopće javlja u datoteci, u njoj javlja točno dva puta, po jednom za svaki od dva zadatka.

Za svakog studenta koji se javlja u barem jednoj od te dvije datoteke, potrebno je stvoriti jedinstvenu tekstualnu datoteku `matbr.txt` (npr. `24.txt`) u kojoj piše `PAD` ako je suma svih njegovih bodova na svim provjerama strogo manja od 100, a inače `POLOZENO`.

Primjer ulaznih i izlaznih datoteka:

<code>prva_provjera.txt:</code>	<code>druga_provjera.txt:</code>	Izlazne datoteke:
3	2	<code>17.txt: PAD</code>
98 51	17 1	<code>24.txt: POLOZENO</code>
17 48	17 0	<code>29.txt: POLOZENO</code>
17 50	29 100	<code>98.txt: POLOZENO</code>
98 50	29 100	
24 50		
24 50		

U rješenju napišite samo dio programa koji radi opisani posao (tj. ne morate pisati `#include` direktive, funkciju `main`, itd.). Možete koristiti pomoćna polja, a ulazne datoteke možete otvarati i zatvarati proizvoljan broj puta. Ne morate provjeravati uspješnost otvaranja niti pisanja datoteka.