

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 10.9., navečer na webu, a uvidi u ponedjeljak, 11.9., u 12 sati.

**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. 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.

**Zadatak 1.** (30 bodova) U kutu šahovske ploče s  $8 \times 8$  polja nalazi se top. On se po ploči može kretati vodoravno i okomito. Svaki potez ima smjer ( $\leftarrow$ ,  $\rightarrow$ ,  $\downarrow$ ,  $\uparrow$ ) i broj mjesta za koji se figura pomakla. Napišite rekurzivnu funkciju koja prima prirodan broj  $n$  i vraća broj načina na koji se top može kretati po šahovskoj ploči, tako da u  $n$  poteza završi na dijagonali suprotnoj od one na kojoj se nalazio na početku. Ako je top na polju  $(1, 1)$ , suprotna dijagonala spaja polja  $(1, 8)$  i  $(8, 1)$ . Stajanje na istom polju nije potez. Funkcija smije imati i druge argumente. Napišite i primjer poziva funkcije.

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 2.** (10 + 15 bodova) Kvadratna matrica reda  $n$  sprema se u memoriji kao niz nizova, tj. kao polje od  $n$  pokazivača na redove, a svaki red je polje od  $n$  elemenata.

- (a) Za dva zadana niza brojeva  $(x_i, i \in \mathbb{N})$  i  $(y_j, j \in \mathbb{N})$ , Cauchyeva matrica  $C$  se definira kao  $c_{i,j} = \frac{1}{x_i - y_j}$ . Zadano je  $x_i = \ln(i)$ , za  $i \in \mathbb{N}$ , i  $y_j = \frac{1}{j-0.5} - 3$ , za  $j \in \mathbb{N}$ .

Napišite funkciju `generirajCauchy` koja, za zadani red matrice  $n \in \mathbb{N}$ , kreira kvadratnu Cauchyevu matricu  $C$ , koristeći zadane  $x_i$  i  $y_j$ . Morate dinamički alocirati odgovarajuću količinu memorije, prije nego što konstruirate matricu! U ovom dijelu zadatka smijete koristiti funkcije iz zaglavlja `math.h`.

- (b) Napišite funkciju `negativnaRotacija` koja zadanu kvadratnu matricu  $A$ , reda  $n$ , transformira na način da svaki element  $a_{i,j}$  spremi na poziciju  $a_{j,n-i+1}$  (niti jedan element ne smije biti prebrisan).

**Napomena:** Ne smijete koristiti dodatne nizove ili polja za rješavanje ovoga zadatka!

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 3.** (25 bodova) Obožavatelji automobilskih utrka čuvaju podatke o vozačima u vezanoj listi. Za svakog vozača pamti se njegovo ime, ime ekipe kojoj pripada (stringovi do 20 znakova), vrijeme ostvareno na posljednjoj utrci (`double`, u sekundama) te ukupan broj bodova osvojenih u prethodnim utrkama (`int`), ne uključujući posljednju.

- (a) Napišite definiciju odgovarajućeg tipa podatka za čvor (element) liste vozača, tako da je moguće definirati varijable naredbom `vozac v`.
- (b) Napišite funkciju `void diskvalificiraj(vozac **lista, char ime[])` koja prima vezanu listu vozača i ime vozača. Funkcija treba izbaciti sve vozače koji su u istoj ekipi kao zadani vozač, uključujući i njega. Ako zadani vozač ne postoji, funkcija ne radi ništa.  
Oslobodite memoriju prilikom izbacivanja vozača iz liste. Funkcija prima i vraća listu preko varijabilnog parametra.
- (c) Napišite funkciju `vozac* rangiraj(vozac *lista, int bodovi)` koja prima vezanu listu vozača i cijeli broj. Funkcija svakom vozaču dodaje bodove ovisno o vremenu ostvarenom na posljednjoj utrci, tako da prvi (onaj s najmanjim vremenom) dobiva `bodovi` bodova, a svaki sljedeći dvostruko manje od svog prethodnika (koristite cjelobrojno dijeljenje). Nije dozvoljeno alocirati nove čvorove. Funkcija vraća pokazivač na modificiranu listu.

**Uputa:** Preuredite listu tako da bude sortirana prema vremenu.

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 4.** (30 bodova) U binarnoj datoteci `banka.dat` zapisan je niz struktura tipa `klijent`, gdje je `klijent` tip podatka koji sadrži ime i prezime klijenta banke (niz od najviše 50 znakova) i njegovo stanje na računu (tipa `double`). U tekstualnoj datoteci `transakcije.txt` nalaze se podaci o transakcijama između klijenata banke — u svakom retku zapisana je po jedna transakcija oblika

```
platitelj=>primatelj: iznos
```

Smijete pretpostaviti da platitelj i primatelj ne sadrže specijalne znakove `'='`, `'>'` i `':'`.

Napišite program koji po redu čita transakcije i ažurira stanje na računima klijenata u datoteci `banka.dat`. Nije dozvoljeno korištenje dodatnih nizova, a transakcije kod kojih je nepoznat platitelj ili primatelj (ne postoji u datoteci `banka.dat`), trebaju se zanemariti.

**Uputa:** Funkcija koja vraća trenutnu poziciju u datoteci je

```
long int ftell(FILE *fp)
```

Rezultat je nenegativan broj byteova od početka datoteke, a `-1L` je signal greške.

Funkcija za “direktno” pozicioniranje u datoteci je

```
int fseek(FILE *fp, long int offset, int origin)
```

gdje je `offset` pomak u byteovima, a `origin` “ishodište” za pomak = jedna od vrijednosti: `SEEK_SET` (početak), `SEEK_CUR` (trenutna pozicija) ili `SEEK_END` (kraj). Izlaz različit od nule je signal greške.

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Rezultati i uvidi u kolokvije:** Rezultati u nedjelju, 10.9., navečer na webu, a uvidi u ponedjeljak, 11.9., u 12 sati.

**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. 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.

**Zadatak 1.** (30 bodova) U kutu šahovske ploče s  $8 \times 8$  polja nalazi se lovac. On se po ploči može kretati dijagonalno. Svaki potez ima smjer ( $\searrow$ ,  $\swarrow$ ,  $\nearrow$ ,  $\nwarrow$ ) i broj mjesta za koji se figura pomakla. Napišite rekurzivnu funkciju koja prima prirodan broj  $n$  i vraća broj načina na koji se lovac može kretati po šahovskoj ploči, tako da u  $n$  poteza završi na dijagonali na kojoj se nalazio na početku. Stajanje na istom polju nije potez. Funkcija smije imati i druge argumente. Napišite i primjer poziva funkcije.

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 2.** (10 + 15 bodova) Kvadratna matrica reda  $n$  sprema se u memoriji kao niz nizova, tj. kao polje od  $n$  pokazivača na redove, a svaki red je polje od  $n$  elemenata.

- (a) Za dva zadana niza brojeva  $(x_i, i \in \mathbb{N})$  i  $(y_j, j \in \mathbb{N})$ , Cauchyeva matrica  $C$  se definira kao  $c_{i,j} = \frac{1}{x_i - y_j}$ . Zadano je  $x_i = \ln(i)$ , za  $i \in \mathbb{N}$ , i  $y_j = \frac{1}{j+0.5} - 2$ , za  $j \in \mathbb{N}$ .

Napišite funkciju `generirajCauchy` koja, za zadani red matrice  $n \in \mathbb{N}$ , kreira kvadratnu Cauchyevu matricu  $C$ , koristeći zadane  $x_i$  i  $y_j$ . Morate dinamički alocirati odgovarajuću količinu memorije, prije nego što konstruirate matricu! U ovom dijelu zadatka smijete koristiti funkcije iz zaglavlja `math.h`.

- (b) Napišite funkciju `pozitivnaRotacija` koja zadanu kvadratnu matricu  $B$ , reda  $n$ , transformira na način da svaki element  $b_{i,j}$  spremi na poziciju  $b_{n-j+1,i}$  (niti jedan element ne smije biti prebrisan).

**Napomena:** Ne smijete koristiti dodatne nizove ili polja za rješavanje ovoga zadatka!

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 3.** (25 bodova) Obožavatelji pjevačkog talent showa čuvaju podatke o sudionicima u vezanoj listi. Za svakog pjevača pamti se njegovo ime, država porijekla (stringovi do 30 znakova), ocjena sudaca na posljednjem natjecanju (`int`) te ukupan broj bodova sa svih dosadašnjih natjecanja (`int`), ne uključujući posljednje.

- (a) Napišite definiciju odgovarajućeg tipa podatka za čvor (`element`) liste pjevača, tako da je moguće definirati varijable naredbom `pjevac p`.
- (b) Napišite funkciju `void odaberi(pjevac **lista, char ime[])` koja prima vezanu listu i ime pjevača. Funkcija treba izbaciti iz liste sve pjevače koji su iz iste države kao i zadani, uključujući i njega. Ako zadano ime ne postoji, funkcija ne radi ništa.  
Oslobodite memoriju prilikom izbacivanja iz liste. Funkcija prima i vraća listu preko varijabilnog parametra.
- (c) Napišite funkciju `pjevac* dodaj_bodove(pjevac* lista, int bodovi)` koja prima vezanu listu pjevača i cijeli broj. Funkcija svakom pjevaču dodaje bodove ovisno o ocjeni na posljednjem natjecanju, tako da prvi (onaj s najvećom ocjenom) dobiva `bodovi` bodova, a svaki sljedeći trećinu bodova svog prethodnika (koristite cjelobrojno dijeljenje). Nije dozvoljeno alociranje novih čvorova. Funkcija vraća pokazivač na modificiranu listu.

**Uputa:** Preuredite listu tako da bude sortirana prema ocjeni.

## Programiranje 2 – popravni kolokvij, 5. 9. 2017.

**Zadatak 4.** (30 bodova) U binarnoj datoteci `burza.dat` zapisan je niz struktura tipa `trgovac`, gdje je `trgovac` tip podatka koji sadrži ime i prezime trgovca na burzi zlata (niz od najviše 50 znakova) i količinu zlata koju posjeduje u gramima (tipa `double`). U tekstualnoj datoteci `promet.txt` nalaze se podaci o transakcijama između trgovaca na burzi — u svakom retku zapisana je po jedna transakcija oblika

```
prodavac->kolicina->kupac
```

Smijete pretpostaviti da prodavač i kupac ne sadrže specijalne znakove `'-'` i `'>'`.

Napišite program koji po redu čita transakcije i ažurira količine zlata koje posjeduju trgovci u datoteci `burza.dat`. Nije dozvoljeno korištenje dodatnih nizova, a transakcije kod kojih je nepoznat prodavač ili kupac (ne postoji u datoteci `burza.dat`), trebaju se zanemariti.

**Uputa:** Funkcija koja vraća trenutnu poziciju u datoteci je

```
long int ftell(FILE *fp)
```

Rezultat je nenegativan broj byteova od početka datoteke, a `-1L` je signal greške.

Funkcija za “direktno” pozicioniranje u datoteci je

```
int fseek(FILE *fp, long int offset, int origin)
```

gdje je `offset` pomak u byteovima, a `origin` “ishodište” za pomak = jedna od vrijednosti: `SEEK_SET` (početak), `SEEK_CUR` (trenutna pozicija) ili `SEEK_END` (kraj). Izlaz različit od nule je signal greške.