

## Programiranje 2, ispit, septembar1 2019.

**Napomena:** Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsis-**  
**tenta\_ImePrezime\_BrojIndeksa**. Na primer, **NC\_PeraPeric\_mi14231**. Sve zadatke sačuvati u ovom  
direktorijumu. Zadatke imenovati sa **1.c**, **2.c**, **3.c** i **4.c**.  
Izrada zadataka traje 2 sata.

1. U datoteci `plate.txt` dat je spisak zarada u različitim gradovima oblika `Grad Plata`. Gradovi se mogu pojaviti više puta. Naziv grada je maksimalne dužine 20 karaktera. Plata je realan broj (koristiti realne brojeve dvostruke tačnosti za zapis plata). Podatke sortirati rastuće prema nazivu grada, a potom opadajuće prema platama. Kao argument komandne linije zadaje se broj  $k$  ( $k/ge1$ ). Ispisati sortirane podatke (naziv grada i plata na tri decimale) tako što se za svaki grad ispiše po  $k$  plata. Ako neki grad ima manje od  $k$  plata onda ispisati one plate koje za taj rad postoje i ne prijavljivati grešku. Nije poznato koliko ima podataka u datoteci. Neka korak realokacije bude 10. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1:

Primer 2:

Pozivanje: `./a.out 3`

Pozivanje: `./a.out 5`

Datoteka `plate.txt`:

Beograd 43400  
Valjevo 23800.88  
Beograd 88232.22  
Beograd 88232.52  
Sabac 31200  
Lazarevac 44500.32  
Lazarevac 120304.33  
Valjevo 56200  
Beograd 220300  
Lazarevac 76900.23  
Beograd 147500

Standardni izlaz:

Beograd 220300.000  
Beograd 147500.000  
Beograd 88232.520  
Lazarevac 120304.330  
Lazarevac 76900.230  
Lazarevac 44500.320  
Sabac 31200.000  
Valjevo 56200.000  
Valjevo 23800.880

Datoteka `plate.txt`:

Beograd 43400  
Valjevo 23800.88  
Beograd 88232.22  
Beograd 88232.52  
Sabac 31200  
Lazarevac 44500.32  
Lazarevac 120304.33  
Valjevo 56200  
Beograd 220300  
Lazarevac 76900.23  
Beograd 147500  
Beograd 90230  
Valjevo 77820  
Krusevac 33545.45  
Jagodina 44900.20  
Beograd 78230  
Jagodina 334500  
Krusevac 38232.22  
Lazarevac 53290  
Valjevo 89600

Standardni izlaz:

Beograd 220300.000  
Beograd 147500.000  
Beograd 90230.000  
Beograd 88232.520  
Beograd 88232.220  
Jagodina 334500.000  
Jagodina 44900.200  
Krusevac 38232.220  
Krusevac 33545.450  
Lazarevac 120304.330  
Lazarevac 76900.230  
Lazarevac 53290.000  
Lazarevac 44500.320  
Sabac 31200.000  
Valjevo 89600.000  
Valjevo 77820.000  
Valjevo 56200.000  
Valjevo 23800.880

-----  
Primer 3:

Primer 4:

Pozivanje: `./a.out`

Pozivanje: `./a.out -10`

Standardni izlaz za greske:  
-1

Standardni izlaz za greske:  
-1

2. Sa standardnog ulaza unosi se ceo broj. Ispisati broj vodećih nula i broj krajnjih nula u binarnom zapisu broja.  
*Napomena: Koristiti tip `int` za čuvanje broja. Prikazani izlaz je u sistemu u kome je `int` 32 bita.*

Primer 1:

Standardni ulaz:  
5

Primer 2:

Standardni ulaz:  
216

Primer 3:

Standardni ulaz:  
26304

Primer 4:

Standardni ulaz:  
-10

Standardni izlaz:  
29 0

Standardni izlaz:  
24 3

Standardni izlaz:  
17 6

Standardni izlaz:  
0 1

3. Sa standardnog ulaza unosi se lista celih brojeva. Na, primer, pretpostaviti da je uneta lista  $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_n \rightarrow b_1 \rightarrow b_2 \rightarrow \dots \rightarrow b_n$ . Izmeniti listu tako da je novodobijena lista oblika  $a_1 \rightarrow b_1 \rightarrow a_2 \rightarrow b_2 \rightarrow \dots \rightarrow a_n \rightarrow b_n$ . Nije dozvoljeno kreiranje nove liste ili novih čvorova. Pretpostaviti da uneta lista ima paran broj elemenata (i ne resavati slucaj kada lista ima neparan broj elemenata). *Pomoćna napomena: Zadatak se može rešiti korišćenjem dva pokazivača, jedan ide od početka liste ( $a_1$ ), a drugi od sredine ( $b_1$ )* *Napomena: Za rad sa listama obavezno koristiti datu biblioteku (`liste.c` i `liste.h`). Zadatak se mora rešiti korišćenjem listi, u suprotnom broj osvojenih poena je 0.*

Primer 1:

Standardni ulaz:  
20 10 -7 17 0 -30

Standardni izlaz:  
[20, 17, 10, 0, -7, -30]

Primer 2:

Standardni ulaz:  
5 1

Standardni izlaz:  
[5, 1]

Primer 3:

Standardni ulaz:  
-9 -10 8 -200 -400 500

Standardni izlaz:  
[-9, -200, -10, -400, 8, 500]

Primer 4:

Standardni ulaz:  
1 2 3 10 20 30

Standardni izlaz:  
[1, 10, 2, 20, 3, 30]

4. Napisati funkciju koja proverava da li je binarno pretraživačko strukturalno (tj. binarno uređeno) stablo simetrično (pogledati sliku 1). Funkcija vraća 1 ako je stablo simetrično, a 0 inače (strukturno označava da se ne posmatraju vrednosti već samo struktura stabla). Sa standardnog ulaza se učitavaju elementi 3 binarna pretraživačka strukturalna (tj. binarna uređena) stabla koji su celi brojevi. Svako stablo se učitava do (EOF). Za svako stablo pozvati napisanu funkciju i ispisati rezultat na standardni izlaz. *Napomena: Za rad sa binarnim pretraživačkim stablima obavezno koristiti datu biblioteku (`stabla.h` i `stabla.c`). Zadatak se mora rešiti korišćenjem binarnog pretraživačkog stabla i ispis traženih vrednosti mora biti izvršen obilaskom čvorova stabla. U suprotnom broj osvojenih poena je 0.*

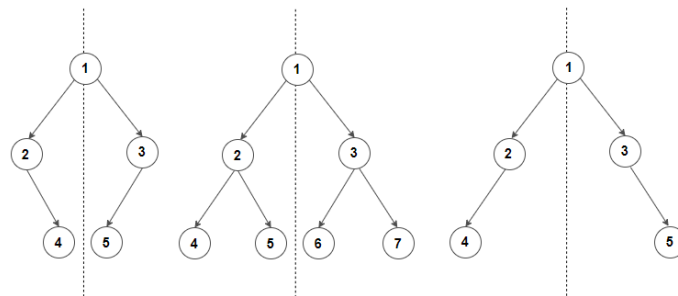


Figure 1: Simetrična stabla

Primer 1:

Standardni ulaz:  
10 5 15 8 12  
10 5 15 8 12 1 20  
10 5 15 1 20

Standardni izlaz:  
1 1 1

Primer 2:

Standardni ulaz:  
10 5 1 15 12  
10 5 15 20 30 8 1  
10 5 15 1 20

Standardni izlaz:  
0 0 1

Primer 3:

Standardni ulaz:  
1 2 -1  
10 5 1 6

Standardni izlaz:  
1 1 0

Primer 4:

Standardni ulaz:  
10 5 15 1 8 12 20 -5 6 13 30  
10 5 15 1 8 12 20 -5 9 13 30  
10 5 20 8 15 7 18

Standardni izlaz:  
1 0 1