

Programiranje 2, ispit, jun 2019.

Grupa I

Napomena: Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsis-
tenta_ImePrezime_BrojIndeksa_1**. Na primer, **NC_PeraPeric_mi14231_1**. 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 se nalaze podaci o transakcijama korisnika po karticama. Korisnik može imati više kartica zadatih sa brojem koji ima 16 cifara. U svakom redu datoteke se nalazi broj kartice, a potom realan broj koji označava iznos. Pozitivan iznos označava uplatu, a negativan iznos isplatu sa računa. Kao argumente komadne linije korisnik zadaje ime i prezime i opciju. Otvoriti datoteku čiji je naziv `ime_prezime.txt` (zameniti ime, prezime sa onim što je uneo korisnik) i odrediti saldo za svaku karticu (ukupnu sumu svih uplata i isplata). Ukoliko je korisnik zadao opciju `-kartica` sortirati podatke prema broju kartice i ispisati na standardni izlaz. Ukoliko je korisnik zadao opciju `-suma` sortirati opadajuće prema sumi na karticama i ispisati na standardni izlaz. Ne praviti nikakve pretpostavke o ukupnom broju transakcija, koristiti funkciju za realokaciju memorije sa korakom 10. U slučaju greške ispisati `-1` na standardni izlaz za greške. Pretpostaviti da su brojevi kartica ispravno zadati i ne proveravati njihovu ispravnost.

Primer 1:

```
Pozivanje: ./a.out Marko Markovic -kartica
Datoteka Marko_Markovic.txt:
1245678915261782    500
9081972619261617    1000
1245678915261782    -200
9181009191010236    -400.89
9181009191010236    2002.99
9081972619261617    56789.74
1245678915261782    300.99
1245678915261782    7816.25
9081972619261617    -5006.12
4890019103002038    13.3
```

```
Standarni izlaz:
1245678915261782 8417.240000
4890019103002038 13.300000
9081972619261617 52783.620000
9181009191010236 1602.100000
```

Primer 2:

```
Pozivanje: ./a.out Marko Markovic -suma
Datoteka Marko_Markovic.txt:
1245678915261782    500
9081972619261617    1000
1245678915261782    -200
9181009191010236    -400.89
9181009191010236    2002.99
9081972619261617    56789.74
1245678915261782    300.99
1245678915261782    7816.25
9081972619261617    -5006.12
4890019103002038    13.3
```

```
Standarni izlaz:
9081972619261617 52783.620000
1245678915261782 8417.240000
9181009191010236 1602.100000
4890019103002038 13.300000
```

Primer 3:

```
Pozivanje: ./a.out Janko Jankovic -novac
Datoteka Janko_Jankovic.txt ne postoji
```

```
Standardni izlaz za greske:
-1
```

Primer 4:

```
Pozivanje: ./a.out Jelena Protic -suma
Datoteka Jelena_Protic.txt:
1818228384779393    230.15
9876798273829376    220
9876798273829376    10.30
```

```
Standardni izlaz:
9876798273829376 230.300000
1818228384779393 230.150000
```

2. Napisati funkciju `unsigned zameni(unsigned x)` koja zamenjuje prva dva bajta sa druga dva bajta, a druga dva bajta sa prva dva bajta broja `x` (a ostali bitovi ostaju nepromenjeni, bajtovi se broje sa desna na levo). Ispisati tako transformisan broj na standardni izlaz.

Primer 1:

```
Standardni ulaz:
0xffffaaaa
```

Primer 2:

```
Standardni ulaz:
0x0
```

Primer 3:

```
Standardni ulaz:
0xff0055
```

Primer 4:

```
Standardni ulaz:
0x7f
```

Standardni izlaz:	Standardni izlaz:	Standardni izlaz:	Standardni izlaz:
0xaaaaffff	0x0	0x5500ff	0x7f0000

3. Sa standardnog ulaza unosi se lista celih brojeva. Izmeniti listu tako da između svaka dva parna broja u listi umeće nulu. Ispisati dobijenu listu na standardni izlaz. *Napomena: Za rad sa listama obavezno koristiti datu biblioteku (liste.c i liste.h). Zadatak se mora rešiti korišćenjem listi i uneta lista mora biti zaista modifikovana. U suprotnom broj osvojenih poena je 0.*

Primer 1:

Standardni ulaz:
20 10 7 17 16 -30 1

Standardni izlaz:
[20, 0, 10, 7, 17, 16, 0, -30, 1]

Primer 2:

Standardni ulaz:
5

Standardni izlaz:
[5]

Primer 3:

Standardni ulaz:
9 10 8 200 400 500

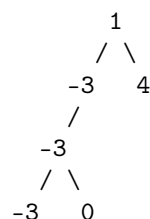
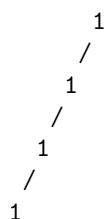
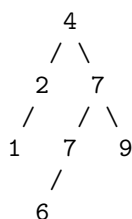
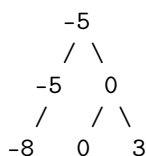
Standardni izlaz:
[9, 10, 0, 8, 0, 200, 0, 400, 0, 500]

Primer 4:

Standardni ulaz:
1 2 3

Standardni izlaz:
[1, 2, 3]

4. Sa standardnog ulaza se učitavaju elementi binarnog pretraživačkog (tj. binarnog uređenog) stabla koji su celi brojevi, sve do kraja ulaza (EOF). Prvo formirati stablo od unetih brojeva, a zatim ispisati na standardni izlaz one vrednosti čvorova stabla čija je vrednost jednaka zbiru vrednosti njegovih neposrednih potomaka, pri čemu ako čvor nema nekog potomka njegov doprinos vrednosti zbira je 0. Listovi nemaju nijednog potomka, što znači da se njihova vrednost ispisuje isključivo ukoliko je jednaka 0. Tražene vrednosti ispisati u onom redosledu koji se dobija **infiksnim obilaskom stabla** (slevo nadesno). U slučaju greške ispisati -1 na standardni izlaz za greške. *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.*



Primer 1:

Standardni ulaz:
-5 -5 0 0 3 -8

Standardni izlaz:
-5 0

Primer 2:

Standardni ulaz:
4 2 1 7 7 9 6

Standardni izlaz:

Primer 3:

Standardni ulaz:
1 1 1 1

Standardni izlaz:
1 1 1

Primer 4:

Standardni ulaz:
1 4 -3 -3 -3 0

Standardni izlaz:
-3 -3 0 1

Programiranje 2, ispit, jun 2019.

Grupa II

Napomena: Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsis-tenta_ImeIPrezime_BrojIndeksa_2**. Na primer, **NC_PeraPeric_mi14231_2**. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa **1.c**, **2.c**, **3.c** i **4.c**.
Izrada zadataka traje 2 sata.

1. Na takmičenju u programiranju učesnici se rangiraju u odnosu na vreme izvršavanja njihovih programa. U datoteci čije se ime zadaje kao argument komandne linije dati su nazivi izvršnih fajlova i vremena njihovog izvršavanja u formatu **hh:mm:ss.sss** (hh-broj sati, mm-broj minuta, ss-broj sekundi i sss-broj milisekundi). Potrebno je napraviti konačnu rang listu kandidata. Napisati program koji čita podatke o izvršnim fajlovima iz date datoteke, sortira ih rastuće u odnosu na vreme izvršavanja i ispisuje konačnu rang listu na standardni izlaz. Pretpostaviti da su nazivi izvršnih fajlova maksimalne dužine 50 karaktera. Ne praviti nikakve pretpostavke o ukupnom broju izvršnih fajlova, koristiti funkciju za realokaciju memorije sa korakom 10. U slučaju greške ispisati -1 na standardni izlaz za greške.

Primer 1:

Pozivanje: ./a.out dat.txt

Datoteka dat.txt:

a.out 00:01:12.670
prvi 00:01:09.401
1.out 00:01:17.092
tmp.out 00:01:19.607
izvrsni 00:01:03.342
zavrsna_verzija 00:00:58.921
1_1 00:01:00.738

Standarni izlaz:

zavrsna_verzija 0:0:58.921
1_1 0:1:0.738
izvrsni 0:1:3.342
prvi 0:1:9.401
a.out 0:1:12.670
1.out 0:1:17.92
tmp.out 0:1:19.607

Primer 2:

Pozivanje: ./a.out rez.txt

Datoteka rez.txt:

k011_izvrsni 00:02:00.670
k020_izvrsni 00:02:09.001
k007_izvrsni 00:02:17.002
k190_izvrsni 00:01:59.630
k082_izvrsni 00:02:06.332
k113_izvrsni 00:01:58.947
k056_izvrsni 00:02:01.767

Standarni izlaz:

k113_izvrsni 0:1:58.947
k190_izvrsni 0:1:59.630
k011_izvrsni 0:2:0.670
k056_izvrsni 0:2:1.767
k082_izvrsni 0:2:6.332
k020_izvrsni 0:2:9.1
k007_izvrsni 0:2:17.2

Primer 3:

Pozivanje: ./a.out tmp.txt

Datoteka tmp.txt:

rb092.out 01:02:00.472
rb119.out 00:56:06.021
rb002.out 00:59:04.010
rb073.out 01:00:55.934
rb108.out 01:02:14.732
rb033.out 00:58:27.375

Standardni izlaz:

rb119.out 0:56:6.21
rb033.out 0:58:27.375
rb002.out 0:59:4.10
rb073.out 1:0:55.934
rb092.out 1:2:0.472
rb108.out 1:2:14.732

Primer 4:

Pozivanje: ./a.out dat.txt

Datoteka dat.txt ne postoji

Standardni izlaz za greske:

-1

2. Napisati funkciju `unsigned invertuj(unsigned x)` koja invertuje bitove na neparnim pozicijama unutar prva dva bajta, a invertuje bitove na parnim pozicijama unutar druga dva bajta broja `x` (a ostali bitovi ostaju nepromenjeni, bajtovi se broje sa desna na levo). Ispisati tako transformisan broj na standardni izlaz.

Primer 1:

Standardni ulaz:

0xffffffff

Standardni izlaz:

0x5555aaaa

Primer 2:

Standardni ulaz:

0x0

Standardni izlaz:

0xaaaa5555

Primer 3:

Standardni ulaz:

0xff00ff

Standardni izlaz:

0xaa5555aa

Primer 4:

Standardni ulaz:

0x7a

Standardni izlaz:

0xaaaa552f

3. Sa standardnog ulaza unose se elementi liste koji su celi brojevi, sve do kraja ulaza (EOF). Prvo formirati listu od unetih brojeva, a zatim ispisati na standardni izlaz vrednosti elementa liste čija je vrednost veća ili jednaka od vrednosti levog suseda. Prvi element liste nema levog suseda. U slučaju greške ispisati -1 na standardni izlaz za greške. *Napomena: Za rad sa listama obavezno koristiti datu biblioteku (**liste.h** i **liste.c**). Zadatak se mora rešiti korišćenjem listi i ispis traženih vrednosti mora biti izvršen obilaskom elemenata liste. U suprotnom broj osvojenih poena je 0.*

Primer 1:

Standardni ulaz:

2 -3 4 7 12 5 0

Standardni izlaz:

4 7 12

Primer 2:

Standardni ulaz:

-1 3 4 1 -3 2 2

Standardni izlaz:

3 4 2 2

Primer 3:

Standardni ulaz:

1 1 1 1 1 1

Standardni izlaz:

1 1 1 1 1

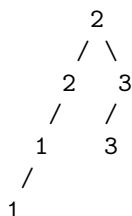
Primer 4:

Standardni ulaz:

15

Standardni izlaz:

4. Sa standardnog ulaza se učitavaju elementi binarnog pretraživačkog stabla (binarnog uređenog stabla), sve do kraja ulaza. Svaki čvor duplirati i umetnuti kao levo dete postojećeg čvora (pogledati primer ispod). *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.*



Primer 1:

Standardni ulaz:

2 1 3

Standardni izlaz:

1 1 2 2 3 3

Primer 2:

Standardni ulaz:

10 3 90 15 6 7 8 45 20 -1

Standardni izlaz:

-1 -1 3 3 6 6 7 7 8 8 10 10 15 15 20 20 45 45 90 90

Primer 3:

Standardni ulaz:

5 3 4 10

Standardni izlaz:

3 3 4 4 5 5 10 10

Primer 4:

Standardni ulaz:

8 -10 11

Standardni izlaz:

-10 -10 8 8 11 11