

Programiranje 2, I smer, kolokvijum, 1. grupa, 22.04.2017.

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsistenta_ImeIPrezime_BrojIndeksa**. Na primer, **AZ_PeraPeric_mi14231**. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa **1.c**, **2.c** i **3.c**. Ne ispisivati niske **Ulaz:** i **Izlaz:**.

1. Kao argumenti komandne linije zadaju se celi brojevi a i b . Sa standardnog ulaza se učitava prirodan broj n i ime datoteke u kojoj se nalazi n celih brojeva. Napisati program koji u datoteku **izlaz.txt** ispisuje koliko je brojeva iz datoteke manje od aritmetičke sredine brojeva a i b . U slučaju greške na standardni izlaz za greške ispisati -1. Pretpostaviti da je sadržaj datoteke ispravan. Maksimalna dužina naziva datoteke je 20 karaktera.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 3 brojevi.txt
Ulaz:
3
  brojevi.txt
BROJEVI.TXT
-5.6
2.6
2.2
IZLAZ.TXT
2
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 2
STANDARDNI IZLAZ ZA GRESKE
-1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 2 2
Ulaz:
3
  dat.txt
DAT.TXT
-1
0
2
IZLAZ.TXT
2
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out -1 1
Ulaz:
1
  brojevi.txt
BROJEVI.TXT
1
IZLAZ.TXT
0
```

2. Sa standardnog ulaza se zadaje pozitivan ceo broj n , a zatim informacije o n studenata. U jednom redu ulaza se nalazi indeks studenta a zatim jedna njegova ocena (broj u intervalu $[6, 10]$). Na standardni izlaz ispisati indeks i prosečnu ocenu, zaokruženu na dve decimale, za svakog studenta o kome su dobijene informacije. Informacije o indeksu i proseku ispisivati u redosledu unosa, ne ispisivati duple vrednosti indeksa. U slučaju greške na standardni izlaz za greške ispisati -1. **Ne postoji pretpostavka o maksimalnoj vrednosti broja n i zadatak se mora uraditi koristeći dinamičku alokaciju, u protivnom se rešenje boduje sa 0 poena.**

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
5
  mi16123 6
  mi16234 8
  mi16123 10
  mi16234 8
  mi16103 6
IzLaz:
mi16123 8.00
mi16234 8.00
mi16103 6.00
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
-2
STANDARDNI IZLAZ ZA GRESKE
-1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
3
  mi16123 6
  mi16123 7
  mi16123 8
IzLaz:
mi16123 7.00
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
3
  mi16123 6
  mi16123 7
  mi16123 4
STANDARDNI IZLAZ ZA GRESKE
-1
```

3. Napisati rekurzivnu funkciju **int noviBroj(int x, int n)** koja iz celog broja x izbacuje prvih n parnih cifara sa desne strane. U slučaju da broj x ima manje od n parnih cifara, funkcija vraća -1. Napisati program koji sa standardnog ulaza učitava ceo broj x i pozitivan ceo broj n i testira funkciju *noviBroj*. U slučaju neispravnih ulaza na standardni izlaz za greške ispisati -1. Napomena: Vrednost -1 dobijenu kao povratnu vrednost funkcije ispisati na standardni izlaz. **Funkcija *noviBroj* mora biti rekurzivna sa navedenim prototipom, u protivnom se rešenje boduje sa 0 poena.**

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
123456 2
IzLaz:
1235
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
123456 5
IzLaz:
-1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
12111 1
IzLaz:
1111
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
1234 0
STANDARDNI IZLAZ ZA GRESKE
-1
```

Programiranje 2, I smer, kolokvijum, 2. grupa, 22.04.2017.

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **InicijaliAsistenta_ImeIPrezime_BrojIndeksa**. Na primer, **AZ_PeraPeric_mi14231**. Sve zadatke sačuvati u ovom direktorijumu. Zadatke imenovati sa **1.c**, **2.c** i **3.c**. Ne ispisivati niske **Ulaz:** i **Izlaz:**.

1. Kao argumenti komandne linije zadaju se pozitivan ceo broj n i ime datoteke u kojoj se nalazi n realnih brojeva. Sa standardnog ulaza se učitavaju dva prirodna broja a i b . Napisati program koji u datoteku **izlaz.txt** ispisuje koliko je brojeva iz datoteke veće od zbira kvadrata brojeva a i b . U slučaju greške na standardni izlaz za greške ispisati -1. Pretpostaviti da je sadržaj datoteke ispravan. Maksimalna dužina naziva datoteka je 20 karaktera.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 3 brojevi.txt
Ulaz:
 2 3
  brojevi.txt
BROJEVI.TXT
 2
-5
 15
IZLAZ.TXT
 1
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 2
STANDARDNI IZLAZ ZA GRESKE
-1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 3 dat.txt
Ulaz:
 1 3
  dat.txt
DAT.TXT
 5
 115
 11
IZLAZ.TXT
 2
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out 3 dat.txt
Ulaz:
-2 -3
  dat.txt
DAT.TXT
 5
-2
-121
IZLAZ.TXT
 0
```

2. Sa standardnog ulaza se zadaje pozitivan ceo broj n , a zatim informacije o n artikala u magacinu. Jedan artikal se zadaje preko svog naziva (dužine najviše 50 karaktera) i broja komada. Na standardni izlaz ispisati ukupnu količinu svakog artikla u redosledu unosa. Ne ispisivati duple nazive artikala. U slučaju greške na standardni izlaz za greške ispisati -1. **Ne postoji pretpostavka o maksimalnoj vrednosti broja n i zadatak se mora uraditi koristeći dinamičku alokaciju, u protivnom se rešenje boduje sa 0 poena.**

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 5
  olovka 50
  sveska 20
  olovka 30
  gumica 100
  olovka 5
Izlaz:
  olovka 85
  sveska 20
  gumica 100
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
-2
STANDARDNI IZLAZ ZA GRESKE
-1
```

Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 3
  olovka 10
  olovka 20
  olovka 30
Izlaz:
  olovka 60
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 3
  olovka 15
  gumica 11
  sveska -15
STANDARDNI IZLAZ ZA GRESKE
-1
```

3. Napisati rekurzivnu funkciju **int izmeniNiz(int niz[], int n)** koja razmenjuje vrednosti i -tog i $(n - 1 - i)$ -tog elementa celobrojnog niza ukoliko su i -ti i $n - 1 - i$ -ti elementi različite parnosti ($i = 0.. \frac{n}{2}$). Napisati program koji sa standardnog ulaza učitava pozitivan ceo broj n a zatim n elemenata niza i na standardni izlaz ispisuje niz izmenjen funkcijom *izmeniNiz*. U slučaju neispravnih ulaza na standardni izlaz za greške ispisati -1. **Funkcija *noviBroj* mora biti rekurzivna sa navedenim prototipom, u protivnom se rešenje boduje sa 0 poena.**

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 4
 2 3 4 6
Izlaz:
 2 4 3 6
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 5
 1 2 3 4 5
Izlaz:
 1 2 3 4 5
```

Primer 6

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
 4 2 3 5 8 1
Izlaz:
 1 2 3 5 8 4
```

Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Ulaz:
-2
STANDARDNI IZLAZ ZA GRESKE
-1
```