

## Programiranje 1, praktični deo, 27.1.2018.

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **P1\_Januar1\_2018\_Tok\_ImePrezime\_BrojIndeksa** gde tok, ime, prezime i broj indeksa treba zameniti svojim podacima. Na primer, za studenta Jovana Marića sa trećeg toka čiji je broj indeksa 205/2016, ime direktorijuma je **P1\_Januar1\_2018\_3\_JovanMaric\_mr16205**. Sve zadatke sačuvati u napravljenom direktorijumu i imenovati ih sa 1.c, 2.c, 3.c, 4.c. Primeri u zadacima koji su obeleženi zvezdicom su eliminacioni.

1. Napisati program koji učitava pozitivan trocifren broj  $n$ , a zatim kreira i na standardni izlaz ispisuje broj koji se dobija spajanjem broja  $n$  i broja koji predstavlja zbir njegovih cifara. U slučaju greške, ispisati  $-1$  na standardni izlaz za grešku.

### Primer 1\*

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
123
IZLAZ:
1236
```

### Primer 2\*

```
INTERAKCIJA SA PROGRAMOM:
987
IZLAZ:
98724
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
-12
IZLAZ ZA GRESKE:
-1
```

2. Napisati program koji učitava reči (niske od najviše 20 karaktera bez belina) do kraja ulaza, a zatim na standardni izlaz ispisuje one reči koje čine neki deo reči *raspolozivost* (bez obzira na veličinu slova). Pretpostaviti da neće biti više od 100 reči. NAPOMENA: *Dozvoljeno je korišćenje funkcije `strstr` deklarisanе u `string.h`.*

### Primer 1\*

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
Ko
ce
da
polozi
IZLAZ:
polozi
```

### Primer 2\*

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
Igras
li
Polo
IZLAZ:
Polo
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
Ziv
zivot
ziv
IZLAZ:
Ziv
ziv
```

3. Napisati program koji transformiše tekst iz datoteke čije se ime navodi kao prvi argument komandne linije i ispisuje ga na standardni izlaz. Transformacija se vrši na sledeći način:
  - ukoliko je tekući karakter ulazne datoteke malo slovo: na izlaz ispisati dva puta odgovarajuće veliko slovo
  - ukoliko je tekući karakter ulazne datoteke veliko slovo: na izlaz ispisati tri puta odgovarajuće malo slovo
  - ukoliko je tekući karakter ulazne datoteke cifra: na izlaz ispisati karakter \*
  - za sve ostale karaktere ulazne datoteke: na izlaz ispisati karakter ?

U slučaju greške, ispisati  $-1$  na standardni izlaz za greške.

### Primer 1\*

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out tekst.txt
TEKST.TXT
Programiranje1
IZLAZ:
pppRR00GGRRAMMIIRRAANNJEE*
```

### Primer 2\*

```
INTERAKCIJA SA PROGRAMOM:
Poziv: ./a.out primer.txt
PRIMER.TXT
Polazete
1. ISPIT
IZLAZ:
ppp00LLAAZZEETTEE*?iisspppiittt
```

### Primer 3

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

4. Definisati strukturu *Igracka* koja sadrži podatke o vrsti igračke u prodavnici: naziv, boju (niske od najviše 20 karaktera) i cenu (realan broj dvostruke tačnosti). Napisati program koji sa standardnog ulaza učitava ceo broj  $n$ , nakon čega se unose podaci o  $n$  igračaka (naziv, boja i cena za svaku redom). Potom se učitava boja željene igračke kao i količina novca kojom raspolaže kupac. Na standardni izlaz ispisati nazive igračaka koje su željene boje i u datom cenovnom rangju. U slučaju da takva igračka ne postoji, ispisati  $-1$  na standardni izlaz za greške. Pretpostaviti da neće biti više od 100 igračaka u prodavnici, kao i da su ulazni podaci ispravni.

### Primer 1\*

```
INTERAKCIJA SA PROGRAMOM:  
ULAZ:  
3  
medvedic plava 599  
lopta zelena 289  
kamion zuta 399  
zelena  
500  
IZLAZ:  
lopta
```

### Primer 2\*

```
INTERAKCIJA SA PROGRAMOM:  
ULAZ:  
4  
lutka ljubicasta 899  
vozic ljubicasta 289  
zirafa zuta 399.99  
medvedic ljubicasta 199.99  
ljubicasta  
450  
IZLAZ:  
vozic  
medvedic
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
ULAZ:  
4  
lutka crvena 1299  
vozic ljubicasta 289  
zirafa zuta 399.99  
medvedic plava 199.99  
zelena  
450  
IZLAZ ZA GRESKE:  
-1
```