

## Logičko i funkcijsko programiranje

### I kolokvijum

Na *Desktop*-u u direktorijumu *Rad* kreirati direktorijum *ImePrezime\_BrIndeksa* i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Nazivi fajlova **moraju** biti oblika *Zadatak1.hs*, *Zadatak2.hs*, *Zadatak3.hs* i *Zadatak4.hs*. Ukoliko je predato rešenje u programskom jeziku *Scala*, ekstenzija odgovarajućeg rešenja treba da bude *scala*, dok je naziv objekta isti kao naziv ulaznog fajla.

1. (Haskell/Scala) Napisati funkciju **brojSamoglasnika** koja broji koliko zadata reč ima samoglasnika.

```
Main> brojSamoglasnika "haskell"
2
```

2. (Haskell/Scala) Napisati funkciju **parniPreNeparnih** koja od zadate liste  $l$  formira novu listu  $l'$  u koju se pakuju redom prvo svi elementi koji se nalaza na parnim pozicijama liste  $l$ , a zatim i elemenati koji se nalaze na neparnim pozicijama liste  $l$ .

```
Main> parniPreNeparnih [0..10]
[0,2,4,6,8,10,1,3,5,7,9]
Main> parniPreNeparnih "haskell"
"hselakl"
```

3. (Haskell) Napisati funkciju **nadjiKoZadovoljava** koja za dati niz funkcija  $a \rightarrow \text{Bool}$  i niz tipa  $a$ , vraća niz u kome je  $i$ -ti element niz koji odgovara elementima datog niza za koje je vrednost  $i$ -te funkcije `True`.

```
Main> nadjiKoZadovoljava [(>2), (<5), odd, even] [1..10]
[[3,4,5,6,7,8,9,10], [1,2,3,4], [1,3,5,7,9], [2,4,6,8,10]]
```

4. (Haskell) Aritmetički izraz može biti konstanta, promenljiva ili zbir/razlika/proizvod aritmetičkih izraza. Nazivi promenljivih su stringovi. Napisati funkcije :

- a) **vrednostIzraza** kojom se za zadatu listu vrednosti promenljivih i izraz izračunava vrednost aritmetičkog izraza.

```
Main> vrednostIzraza [("a",2), ("a1",3)] (Sab (Konst 10.0) (Sab (Prom
"a") (Prom "a1")))
15.0
```

- b) **vrednostDefinicije** kojom se za zadatu listu uređenih parova promenljivih i aritmetičkih izraza

koji im odgovaraju, preračunavaju vrednosti promenljivih. Aritmetički izrazi koji odgovaraju promenljivama mogu sadržati i promenljive. Podrazumevati da nema rekurzivnih zavisnosti (npr.  $x = y, y = x$ ) i da je moguće izračunati vrednost za svaku promenljivu.

```
Main> vrednostDefinicije [("y", (Sab (Prom "x") (Konst 2))), ("x",  
  (Konst 2)), ("z", (Sab (Prom "y") (Prom "x")))]  
  
[("y", 4.0), ("x", 2.0), ("z", 6.0)]
```

#### Korisne ugrađene funkcije :

- Prebacivanje stringa u broj - `read p :: Int`
- Glava liste - `head lista`
- Rep liste - `tail lista`
- Prebacivanje broja u string - `show n`
- Ispis stringa na ekran - `putStr string`