

Logičko i funkcijsko programiranje

I kolokvijum

2023/2024

Na *Desktop*-u u direktorijumu *Rad* kreirati direktorijum *Ime_Prezime_BrIndeksa_GodinaUpisa* 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) **(3 poena)** Napisati funkciju *sakrij* koja prihvata znak, niz znakova koje treba zameniti i tekst čiji sadržaj se krije na osnovu datog znaka.

```
Main> sakrij '*' "aeiou" "ovo je neki tekst"
"*v* j* n*k* t*kst"
```

2. (Haskell/Scala) **(6 poena)** Napisati funkciju *skriveneReci* koja prihvata „tajni znak“ i tekst, a kao rezultat vraća spisak svih sakrivenih reči koje sadrže više od 1 tajnog znaka. Spisak predstavlja niz torki, gde je prva vrednost sama ta reč, a druga broj pojavljivanja tajnog znaka u njoj.

```
Main> skriveneReci '*' "*v* j* n*k* t*kst"
[ ("*v*", 2), ("n*k*", 2) ]
```

3. (Haskell) **(6 poena)** Napisati funkciju *minesweeper* koja prihvata niz stringova koji predstavlja postavku bombi, gde su bombe predstavljene sa „*“, a prazno polje „-“. Kao rezultat, funkcija vraća matricu koja predstavlja broj bombi koje se nalaze u okolini datog polja. U broj bombi uračunati i bombu koja se nalazi na polju koje se posmatra.

```
Main> minesweeper ["*-","--*","--*"]
[[1,2,2],[1,3,3],[0,2,2]]
```

4. (Haskell) **(8 poena)** Napisati funkciju *emacs* koja će simulirati tekst editor. Editor podržava sledeće komande:

- **h** – pomera kursor za 1 mesto ulevo; ako je kursor na početku teksta, komande ovog tipa nemaju efekta
- **l** – pomera kursor za 1 mesto udesno (slovo L); ako je kursor na kraju teksta, komande ovog tipa nemaju efekta

- `x` – briše znak na kome se kursor nalazi; pri brisanju znaka kursor se ne pomera, osim u slučaju kada se briše poslednji znak i tada se pomera za jedno mesto ulevo
- `r<znak>` – menja znak na kome se kursor nalazi sa znakom koji je prosleđen

Funkcija prihvata početni tekst i spisak komandi koje je potrebno izvršiti. Više komandi može da se definiše i u okviru jedne komande. U slučaju da u nizu komandi postoji neka komanda koja nije validna, funkcija vraća niz sa porukom „Nevalidna komanda. Pokušajte ponovo“. Na početku programa, kursor se nalazi na poslednjem znaku u tekstu. Program se završava nakon izvršavanja svih komandi ili u slučaju kada dođe do greške.

```
Main> emacs "abcde" ["hhxx", "hhrA"]
["abcde", "abe", "Abe"]
```

```
Main> emacs "abcde" ["hChxx", "hhrA"]
["Nevalidna komanda. Pokušajte ponovo"]
```

```
Main> emacs "kolokvijum" ["hhxxx", "hhhhhxxxxxx"]
["kolokvijum", "kolokvi", "vi"]
```

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`

» Zadatak3.hs

```
1  type Lokacija = (Int,Int)
2  type Polje    = String
3  type Tabla    = [Polje]
4
5  prebroj :: Lokacija -> (Int,Int) -> Tabla -> Int
6  prebroj (i,j) (maxLenI,maxLenJ) polja
7      = sum [vrednost (dajEl r k) | r <- indeksiRed, k <- indeksiKol]
8      where
9          uOpsegu maxLen x = x >= 0 && x <= maxLen
10         indeksiRed = filter (uOpsegu maxLenI) [i-1, i, i+1]
11         indeksiKol = filter (uOpsegu maxLenJ) [j-1, j, j+1]
12         dajEl r k = polja !! r !! k
13         vrednost x = if x == '*' then 1 else 0
14
15
16  minesweeper :: Tabla -> [[Int]]
17  minesweeper tabla = [
18      | j <- [0 .. maxJ]]
19      | i <- [0 .. maxI]
20      ]
21  where
22      maxI = length tabla - 1
23      maxJ = length (head tabla) - 1
24      maxLen = (maxI, maxJ)
```

✖ Zadatak4.hs

```
1  type Command = String
2  type Cursor = Int
3
4  errMsg :: String
5  errMsg = "Nevalidna komanda. Pokusajte ponovo"
6
7
8  delete :: Cursor -> String -> String
9  delete c text = (take c text) ++ (drop (c+1) text)
10
11
12  rename :: Cursor -> Char -> String -> String
13  rename c character text
14  | c < length text && c >= 0 = (take c text) ++ [character] ++ (drop (c+1) text)
15  | otherwise                  = text
16
```

✖ Zadatak4.hs

```
17
18  exec :: Cursor -> String -> Command -> (Cursor, String)
19  exec cursor text [] = (cursor, text)
20  exec cursor text (c:cs)
21  | c == 'h' = exec (if cursor - 1 < 0 then 0 else cursor - 1) text cs
22  | c == 'l' = let
23  |           | len = length text
24  |           | cur = if cursor + 1 < len
25  |           |           then cursor + 1
26  |           |           else len
27  |           | in
28  |           |   exec cur text cs
29  | c == 'x' = let
30  |           | len = length text
31  |           | txt = delete cursor text
32  |           | cur = if cursor == len - 1 && cursor /= -1
33  |           |           then cursor - 1
34  |           |           else cursor
35  |           | in
36  |           |   exec cur txt cs
37  | c == 'r' && cs /= [] = exec cursor (rename cursor (head cs) text) (tail cs)
38  | otherwise = (cursor, errMsg)
```

» Zadatak4.hs

```
39
40
41 editor :: String -> Cursor -> [Command] -> [String]
42 editor [] cursor commands = []
43 editor text cursor [] = [text]
44 editor initText initCursor (c:cs)
45     | text /= errMsg = initText : editor text cursor cs
46     | otherwise      = [errMsg]
47     where
48         result = exec initCursor initText c
49         cursor = fst result
50         text   = snd result
51
52
53 emacs :: String -> [Command] -> [String]
54 emacs initText commands = editor initText (length initText - 1) commands
```