

Programske paradigme (R-smer), praktični deo, SEP1

Na *Desktop*-u se nalazi arhiva `ppr_sep1_2024_ImePrezime_AlasNalog.zip` čiji direktorijum po raspakivanju u `~/Desktop` treba preimenovati korišćenjem svojih podataka. Na primer, za studenta Jovana Marića čiji je broj indeksa 205/2022, ime direktorijuma je `ppr_sep1_2024_JovanMaric_mr22205`. Svaki zadatak sačuvati u odgovarajućem poddirektorijumu.

1. **20% [Python]** Napisati program kojem se sa standardnog ulaza unosi putanja do fajla kojeg je potrebno ispitati. Na standardni izlaz potrebno je ispisati broj linije u datom fajlu, praćenu dvotačkom i zatim tekstualnim sadržajem te linije, za svaku liniju u kojoj se pojavljuje znak uzvika. Program napisati bez korišćenja petlji. *Napomena:* za potrebe testiranja možete koristiti tekstualne datoteke iz direktorijuma 1.

Primer 1

```
Poziv: python3 1.py
ULAZ:
horizont.txt
IZLAZ:
9:'Pogledajte ovu neverovatnu lepotu!'
20:'Ovo je bilo fantastično! Jedva čekam da se vratim i ponovim sve ponovo!'
```

Primer 2

```
Poziv: python3 1.py
ULAZ:
selo.txt
IZLAZ:
```

Primer 3

```
Poziv: python3 1.py
ULAZ:
muzej.txt
IZLAZ:
18:'Ovo je bio jedan od najboljih dana u mom životu!'
```

2. **20% [Haskell]** Data je lista uređenih trojki. Svaka trojka liste ima šablonski tip (a, b, c) za koji važi ograničenje $\text{Ord } a \Rightarrow \text{Ord } b \Rightarrow \text{Ord } c$. Napisati funkciju `sortiraj` kojoj se kao prvi argument prosleđuje pozitivan broj i koji predstavlja redni broj koordinate po kojoj je potrebno sortirati listu poštujući relaciju (\leq) uspostavljenu u šablonskom tipu. Pritom je potrebno očuvati relativan poredak jednakih elemenata u listi u odnosu na kriterijum poređenja, tj. sortiranje treba da bude *stabilno*.

Primer 1

```
Poziv: ghci 2.hs
ULAZ:
l1 = [(0, 30, "G"), (10,20,"E"), (0,10,"F"), (0,20,"F")]
l2 = sortiraj 2 l1
l2
l3 = sortiraj 1 l2
l3
sortiraj 3 l3
IZLAZ:
[(0,10,"F"),(10,20,"E"),(0,20,"F"),(0,30,"G")]
[(0,10,"F"),(0,20,"F"),(0,30,"G"),(10,20,"E")]
[(10,20,"E"),(0,10,"F"),(0,20,"F"),(0,30,"G")]
```

Primer 2

```
Poziv: ghci 2.hs
ULAZ:
sortiraj 5 [("b", "g", "t"), ("a", "q", "w"),
("y", "h", "n")]
sortiraj 1 [(1, 2, 3), (0, 3, 3)]
sortiraj 2 [(1, 2, 3), (0, 3, 3)]
sortiraj 3 [(1, 2, 3), (0, 3, 3)]
sortiraj 4 [(1, 2, 3), (0, 3, 3)]
IZLAZ:
[("b", "g", "t"), ("a", "q", "w"), ("y", "h", "n")]
[(0,3,3),(1,2,3)]
[(1,2,3),(0,3,3)]
[(1,2,3),(0,3,3)]
[(1,2,3),(0,3,3)]
```

3. **20% [Haskell]** Napisati funkciju `ciklicno :: Int -> [Int] -> [Int]` kojoj se prosleđuje ceo broj n i lista koja sadrži dekadne cifre. Rezultat je prosleđena lista ciklično pomerena za n mesta ulevo. Zatim napisati funkciju `brojOdListe :: [Int] -> Int` koja vraća broj određen ciframa koje se nalaze u listi čitajući ih sa kraja ka početku liste. Pretpostaviti da je ulaz korektan, tj. da su elementi liste zaista cifre.

Primer 1

```
Poziv: ghci 3.hs
ULAZ:
ciklicno (2) [1..5]
ciklicno (3) [0..4]
ciklicno (-1) [2..6]
IZLAZ:
[3,4,5,1,2]
[3,4,0,1,2]
[6,2,3,4,5]
```

Primer 2

```
Poziv: ghci 3.hs
ULAZ:
brojOdListe [2..9]
brojOdListe [3, 6, 9]
brojOdListe [7, 0, 0]
IZLAZ:
98765432
963
7
```

Primer 3

```
Poziv: ghci 3.hs
ULAZ:
l = ciklicno (-2) [0..9]
brojOdListe l
IZLAZ:
7654321098
```

4. **20% [Prolog]** Napisati program koji rešava sledeću zagonetku. Programeri se zovu Džejms, Margaret, Trejsi, Džon i Linus, a prezivaju Hamilton, Ču, Gozling, Torvalds i Karmak. Svi imaju različit broj godina iskustva od jedne do šest. Važi sledeće:
- (a) Džejms se zaposlio tačno godinu dana pre programerke koja se preziva Ču, koja se zaposlila tačno godinu dana pre programerke Margaret
 - (b) Džon radi barem duplo duže od programerke koja se preziva Hamilton
 - (c) Linus se preziva Torvalds
 - (d) Programer koji se preziva Gozling ima tri godine više iskustva od Trejsi

Rešenje zagonetke je lista struktura koje jednoznačno određuju svaku osobu (ime, prezime, godineIskustva).

- 1) Napisati predikat `programeri(Lista)` koji rešava zagonetku i promenljivu `Lista` unifikuje sa rešenjem zagonetke.
- 2) Napisati predikat `odgovori(X,Y)` u kom se promenljiva `X` unifikuje sa imenom programerke koja se preziva Ču, a promenljiva `Y` sa prezimenom programerke koja se zove Margaret.

5. **15% [Prolog]** Napisati program koji pronalazi sve vrednosti promenljivih X , Y i Z za koje važi da je $X + Z > 30$, $Y * Z < 1000$ i $X * Z + Y^3 = 10600$ pri čemu promenljive pripadaju narednim domenima $X \in \{13, 26, 39, \dots, 650\}$, $Y \in \{2, 4, 6, \dots, 46\}$ i $Z \in \{0, 10, 20, \dots, 800\}$. Ukoliko ima više rešenja, ispisati ono sa minimalnom vrednošću X .