

Programske paradigme (R-smer), praktični deo, JAN1

Na *Desktop*-u se nalazi arhiva `ppr_jan1_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_jan1_2024_JovanMaric_mr22205`. Svaki zadatak sačuvati u odgovarajućem poddirektorijumu.

1. **20% [Python]** Napisati program koji oponaša *Unix* komandu `grep`. Najpre se sa standardnog ulaza unosi putanja do fajla kojeg je potrebno pretražiti, a zatim se, u zasebnom redu unosi neka reč. Na standardni izlaz potrebno je ispisati broj linije u datom fajlu, u kojoj se ta reč pojavljuje, praćenu dvotačkom i zatim tekstualnim sadržajem te linije. U slučaju da datoteka ne sadrži zadatu reč ispisati poruku (videti primer 3). Program napisati bez korišćenja petlji. *Napomena:* za potrebe testiranja možete koristiti tekstualne datoteke iz direktorijuma 1.

Primer 1

```
Poziv: python3 1.py
Ulaz:
mr-hyde.txt
table
Izlaz:
7:'I rose accordingly from table'
```

Primer 2

```
Poziv: python3 1.py
Ulaz:
sherlock.txt
instant
Izlaz:
20:'The instant that I had crossed'
34:'I shall not keep you waiting an instant'
70:'for an instant'
```

Primer 3

```
Poziv: python3 1.py
Ulaz:
sherlock.txt
asdf
Izlaz:
Rec 'asdf' nije pronadjena!
```

2. **25% [Haskell]** Podaci o tačkama u ravni su zadati kao `type Point = (Int, Int)`, a svi kvadranti su definisani kao `data Quadrant = Q1 | Q2 | Q3 | Q4 deriving Show`.

- (a) Definisati tip `QuadrantPoints` kao uređenu četvorku lista tačaka. U nastavku zadatka će se u *i*-toj koordinati objekata tog tipa skladištiti tačke koje pripadaju *i*-tom kvadrantu. Definisati funkciju `get :: Quadrant -> QuadrantPoints -> [Point]` koja pribavlja sve tačke iz kvadranta koji se prosleđuje kao prvi argument. Definisati funkciju `set :: [Point] -> Quadrant -> QuadrantPoints -> QuadrantPoints` kojom se može postaviti skup tačaka u odgovarajući kvadrant.

Primer 1

```
Poziv: ghci 2.hs
Ulaz:
qp = ([], [], [], [(1, -1)]) :: QuadrantPoints
qp
Izlaz:
([], [], [], [(1,-1)])
```

Primer 2

```
Poziv: ghci 2.hs
Ulaz:
qp = ([], [], [], [(1, 1)])
get Q1 qp
get Q4 qp
Izlaz:
[]
[(1,1)]
```

Primer 3

```
Poziv: ghci 2.hs
Ulaz:
qp = ([], [], [], [(1, -1)])
set [(1, 1)] Q1 qp
Izlaz:
([(1,1)], [], [], [(1,-1)])
```

- (b) Definisati funkcije `xAxis :: Point -> Bool` i `yAxis :: Point -> Bool` koje odgovaraju na pitanje da li se tačka prosleđena kao prvi argument nalazi na odgovarajućoj osi, a zatim i `onAxis :: Point -> Bool` koja odgovara na pitanje da li se tačka nalazi na bilo kojoj osi.

Primer 1

```
Poziv: ghci 2.hs
Ulaz:
onAxis (0, -2)
onAxis (1, -1)
onAxis (2, 0)
Izlaz:
False
False
True
```

Primer 2

```
Poziv: ghci 2.hs
Ulaz:
yAxis (-1, 1)
yAxis (0, 1)
yAxis (1, 1)
Izlaz:
False
True
False
```

Primer 3

```
Poziv: ghci 2.hs
Ulaz:
onAxis (0, -2)
onAxis (1, -1)
onAxis (2, 0)
Izlaz:
True
False
True
```

NASTAVAK NA SLEDEĆOJ STRANI!

- (c) Definirati funkciju `getQuadrant :: Point -> Maybe Quadrant` koja za datu tačku određuje kojem kvadrantu pripada. Definirati funkciju `pointsFromRect` koja pravi listu svih tačaka koje imaju celobrojne koordinate, a koje pripadaju pravougaoniku sa temenima iz opsega koji se zadaje argumentima: $X_{min}, Y_{min}, X_{max}, Y_{max}$. Definirati funkciju `groupByQuadrants :: [Point] -> QuadrantPoints` koja za datu listu tačaka, ne menjajući njihov međusobni poredak, grupiše tačke po kvadrantima. Tačke sa osu ne pripadaju nijednom kvadrantu.

Primer 1

```
Poziv: ghci 2.hs
ULAZ:
  getQuadrant (1, 1)
  getQuadrant (-1, -1)
IZLAZ:
  Just Q1
  Just Q3
```

Primer 2

```
Poziv: ghci 2.hs
ULAZ:
  pointsFromRect (-1) (-1) 1 1
IZLAZ:
  [(-1,-1),(-1,0),(-1,1),(0,-1),
   (0,0),(0,1),(1,-1),(1,0),(1,1)]
```

Primer 3

```
Poziv: ghci 2.hs
ULAZ:
  pts = pointsFromRect (-2) (-2) 1 1
  groupByQuadrants pts
IZLAZ:
  [(1,1),(-2,1),(-1,1)],
  [(-2,-2),(-2,-1),(-1,-2),
   (-1,-1)],[(1,-2),(1,-1)]
```

3. **20% [Haskell]** Napisati funkciju `moduloListe :: Int -> [Int] -> [Int]` kojoj se prosleđuje cifra k i lista nenegativnih celih brojeva. Rezultat je lista celobrojnih ostataka pri deljenju sa k . Zatim napisati funkciju `brojOdListe :: [Int] -> Int` koja vraća broj određen ciframa koje se nalaze u listi čitajući ih sa početka ka kraju liste. Pretpostaviti da je ulaz korektan, tj. da je k zaista cifra, a data lista sa nenegativnim brojevima.

Primer 1

```
Poziv: ghci 3.hs
ULAZ:
  moduloListe 2 [1, 2, 3]
  moduloListe 3 [0..5]
  moduloListe 5 [2..8]
IZLAZ:
  [1,0,1]
  [0,1,2,0,1,2]
  [2,3,4,0,1,2,3]
```

Primer 2

```
Poziv: ghci 3.hs
ULAZ:
  brojOdListe [1..9]
  brojOdListe [0, 1, 2, 3]
  brojOdListe [9, 8, 7]
IZLAZ:
  123456789
  123
  987
```

Primer 3

```
Poziv: ghci 3.hs
ULAZ:
  l = moduloListe 9 [5..20]
  brojOdListe l
IZLAZ:
  5678012345678012
```

4. **20% [Prolog]** Nekoliko učenika trećeg razreda osnovne škole došlo je na ideju da se zajednički pripremaju za kontrolne vežbe iz različitih predmeta. Svaki od ukupno četiri učenika ove grupe ima svoj omiljeni predmet za koji je zadužen. Tako su se okupili da ne postoji predmet koji je omiljen više nego jednom učeniku. Važi sledeće:

- (a) Branković će spremati istoriju
- (b) Ni Milenković, ni Jelena neće spremati srpski jezik
- (c) Jelisaveta, koja se ne preziva Blagojević, će spremati geografiju
- (d) Borko će spremati matematiku
- (e) Dunja se preziva Plazinić

Rešenje zagonetke je lista struktura koje jednoznačno određuju svakog učenika (ime, prezime, predmet).

- 1) Napisati predikat `ucenje(Lista)` koji rešava zagonetku i promenljivu `Lista` unifikuje sa rešenjem zagonetke.
- 2) Napisati predikat `odgovori(X,Y)` u kom se promenljiva `X` unifikuje sa prezimenom učenika koji će se spremati za srpski, a promenljiva `Y` sa imenom učenika koji se preziva Blagojević.

5. **15% [Prolog]** Porodična firma bavi se ručnom izradom keramičkih šolja i ukrasnih posuda. Za proizvodnju jedne posude potrebno je 50 minuta, dok je za šolju dovoljno 35 minuta. Masa gline za jednu posudu iznosi 550g, dok je za šolju potrebno 300g. Zarada koja se ostvari prilikom prodaje jedne posude je 1300 dinara, a prilikom prodaje jedne šolje je 700 dinara. Firma planira svojih narednih 10 radnih dana sa standardnim osmočasovnim radnim vremenom. Ukoliko se zna da firma trenutno raspolaže sa 10kg gline, koliko komada tanjira, a koliko šolja treba da proizvedu kako bi se ostvarila maksimalna zarada? Pretpostaviti da će svi proizvodi biti prodati.