

R smer, Programske paradigme 2025, SEP2

Na *Desktop-u* se nalazi arhiva **ppr_sep2_2025_ImePrezime_AlasNalog.zip** koju po raspakivanju treba preimenovati korišćenjem svojih podataka. Na primer, za studenta Jovana Marića čiji je broj indeksa 205/2022, ime direktorijuma je **ppr_sep2_2024_JovanaMarića**. Svaki zadatak sačuvati u odgovarajućem poddirektorijumu.

1. **[Python (20%)]** Napisati program koji prolazi kroz fajl i traži zadatu reč. Kao argument komandne linije se unosi putanja do fajla kojeg je potrebno pretražiti. Ukoliko korisnik unese previše ili premalo argumenata komandne linije, ispisuje se poruka o grešci i program se prekida. Inače, unosi se neka reč. Na standardni izlaz potrebno je ispisati broj linije u datom fajlu, u kojoj se ta reč pojavljuje, praćenu crticom i zatim tekstualnim sadržajem te linije. U slučaju da datoteka ne sadrži zadatu reč ispisati poruku (videti primer 3). Program napisati **bez korišćenja petlji i rekurzije**.

Primer 1*

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

Primer 2*

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

Primer 4*

```
Poziv: python3 1.py
IZLAZ:
Nedovoljno argumenata!
```

2. [Haskell (25%)]

a) Definisati funkciju `armstrongovBroj :: Int -> Bool` koja određuje da li je dati dekadni broj Armstrongov. Dekadni broj je Armstrongov ako je jednak zbiru svojih cifara podignutih na n-ti stepen, pri čemu je n broj cifara datog broja. Zatim napisati funkcije `izdvojArmstrongove :: [Int] -> [Int]` koja dobija listu `lst` na ulazu i vraća listu elemenata koji su Armstrongovi, `sadrziArmstrong :: [Int] -> Bool`, koja proverava da li u prosleđenoj listi ima Armstrongovih brojeva i `sviArmstrongovi :: [Int] -> Bool` koja proverava da li su svi elementi liste Armstrongovi brojevi.

Primer 1*

```
Poziv: armstrongovBroj 371
IZLAZ:
True
```

Primer 2*

```
Poziv: sviArmstrongovi [1..9]
IZLAZ:
True
```

Primer 3*

```
Poziv: armstrongovBroj 372
IZLAZ:
False
```

Primer 4*

```
Poziv: izdvojArmstrongove [0, 1, 15, 371, 372, 9474]
IZLAZ:
[0, 1, 371, 9474]
```

Primer 5*

```
Poziv: sadrziArmstrong [0, 372, 373]
IZLAZ:
True
```

Primer 6*

```
Poziv: sadrziArmstrong [372, 373, 374]
IZLAZ:
False
```

b) Napisati funkciju `sifruj :: String -> Char -> Maybe String`, koja na osnovu prosleđenog karaktera modifikuje nisku na sledeće načine:

- ukoliko je prosleđen karakter "s", sva velika slova u nisci se smanjuju,
- ukoliko je prosleđen karakter "u", sva mala slova se uvećavaju,
- ukoliko je prosleđen karakter "b", potrebno je obrisati cifre iz niske,
- inače, vratiti `Nothing`.

Napomena: dozvoljeno je koristiti modul `Data.Char`.

Primer 1*

```
Poziv: ghci 2.hs
Ulaz: sifruj "SManJI FoNt" 's'
IzLAZ:
Just "smanji font"
```

Primer 2*

```
Poziv: ghci 2.hs
Ulaz: sifruj "SManJI FoNt" 'u'
IzLAZ:
Just "SMANJI FONT"
```

Primer 3*

```
Poziv: ghci 2.hs
Ulaz: sifruj "SManJI FoNt, 1a2b34!" 'b'
IzLAZ:
Just "SManJI FoNt, ab!"
```

Primer 4*

```
Poziv: ghci 2.hs
Ulaz: sifruj "SManJI FoNt, 1a2b34!" 'n'
IzLAZ:
Nothing
```

3. **[Haskell (20%)]** Napisati biblioteku funkcija koja pomaže asistentima da vode evidenciju o studentima na kursu:

- Napisati funkciju `polozili :: [(String, Int)] -> [String]` koja čuva rezultate ispita. Svaki element liste čini jedan par: ime studenta i broj bodova koje je on ostvario na ispitu. Vratiti spisak imena svih studenata koji su položili ispit. Napisati funkciju `najbolji :: [(String, Int)] -> String` koji nalazi ime studenta koji je najbolje uradio ispit.
- Napisati funkciju `ocena :: (String, Int) -> Int` koja za studenta i broj poena na ispitu vraća ocenu koju je student dobio, kao i funkciju `prosek :: [(String, Int)] -> Float` koja računa prosečnu ocenu svih studenata koji su **položili ispit**.

Primer 1*

```
Poziv: ghci 3.hs
Ulaz: polozili [("Ana", 35), ("Marko", 55), ("Milica", 98)]
IzLAZ:
["Marko", "Milica"]
```

Primer 2*

```
Poziv: ghci 3.hs
Ulaz: najbolji [("Ana", 35), ("Marko", 55), ("Milica", 98)]
IzLAZ:
"Milica"
```

Primer 3*

```
Poziv: ghci 3.hs
Ulaz: ocena ("Milica", 98)
IzLAZ:
10
```

Primer 4*

```
Poziv: ghci 3.hs
Ulaz: prosek [("Marko", 55), ("Ana", 35), ("Milica", 68)]
IzLAZ:
6.5
```

4. **[Prolog (15%)]** Posle naporene sezone, na završni masters u snukeru plasiralo se najboljih 6 igrača. Njihova imena su Džad, Mark, Roni, Džon, Ding i Kajren, a prezimena Vilson, Higin, Junghui, O'Saliven, Tramp i Vilijams. Njihovi rangovi su od 1 do 6. Pored toga, važi i sledeće:

- Ding se preziva Junghui, a Džad je prvi na rang listi.
- Džon ima duplo gori rejting od Vilsona, koji se nalazi jedno mesto iza Trampa na rang listi.
- Roni ima za 1 slabiji rejting od Higin, čiji je rejting za 1 slabiji od Marka.
- O'Saliven je na rang listi 3 mesta ispod Kajrena.

Rešenje zagonetke je lista struktura koje jednoznačno određuju svakog igrača (ime, prezime, rang).

1) Napisati predikat `snuker(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 igrača koje se preziva O'Saliven, a promenljiva Y njegovim rangom.

5. **[Prolog (20%)]** Napisati program pronalazi vrednosti A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S (svako slovo predstavlja različit broj) koje su poredane u heksagon na sledeći način:

A, B, C
D, E, F, G
H, I, J, K, L
M, N, O, P
Q, R, S

tako da zbir vrednosti duž svake horizontalne i dijagonalne linije bude 38 ($A+B+C = D+E+F+G = \dots = Q+R+S = 38$, $A+D+H = B+E+I+M = \dots = L+P+S = 38$, $C+G+L = B+F+K+P = \dots = H+M+Q = 38$). Sve rezultate čitko ispisati na standardni izlaz.