

R smer, Programske paradigme 2025, SEP1

Na *Desktop-u* se nalazi arhiva **ppr_sep1_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_sep1_2025_J**. Svaki zadatak sačuvati u odgovarajućem poddirektorijumu.

1. **[Python (20%)]** Napisati program koji prolazi kroz fajl i traži zadatu reč. Putanja do fajla i reč unose se u zasebnom redu. Za svaku liniju u kojoj se tražena reč nalazi na izlaz ispisati: redni broj linije u kojoj se reč pojavljuje, indeks početka reči u liniji i sadržaj linije razdvojene dvotačkom. U slučaju da datoteka ne sadrži zadatu reč ili da fajl sa zadatim imenom ne postoji ispisati poruku (videti primere 3 i 4). Program napisati **bez korišćenja petlji i rekurzije**.

Primer 1*

```
Poziv: python3 1.py
Ulaz:
mr-hyde.txt
table
IzLAZ:
7:24:I rose accordingly from table
```

Primer 2*

```
Poziv: python3 1.py
Ulaz:
sherlock.txt
instant
IzLAZ:
20:4:The instant that I had crossed
34:32:I shall not keep you waiting an instant
70:7:for an instant
```

Primer 3*

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

Primer 4*

```
Poziv: python3 1.py
Ulaz:
nepostojeciFajl.txt
table
IzLAZ:
Fajl sa ovim imenom ne postoji!
```

2. **[Haskell (20%)]** Ana uči izdvajanje cifara broja i formiranje novih brojeva od njih. Definisati sledeće funkcije koji Ani pomažu da proveri svoje znanje:

- **parNepar :: Int -> (Int, Int)** - formira par brojeva (BrojP, BrojN) na osnovu datog celog broja N: BrojP koji predstavlja broj formiran od parnih cifara broja N i BrojN koji predstavlja broj formiran od neparnih cifara broja N čitajući cifre broja N sa desna na levo. U slušaju da u zapisu broja ne postoje odgovarajuće cifre, rezultat je broj 0.
- **listaParNepar :: [Int] -> [(Int, Int, Int)]** - od liste celih brojeva L formira listu trojki čiji su elementi oblika (N, BrojP, BrojN), gde je N element liste L, a BrojP i BrojN odgovarajući brojevi formirani pozivom funkcije parNepar.
- **saberiNeparne :: [Int] -> Int** i **saberiParne :: [Int] -> Int** - koji redom sabiraju samo neparne, tj. parne brojeve dobijene pozivom funkcije listaParNepar.

Primer 1*

```
Poziv: ghci 2.hs
Ulaz: parNepar 123
IzLAZ:
(2, 13)
```

Primer 2*

```
Poziv: ghci 2.hs
Ulaz: parNepar (-9)
IzLAZ:
(0, 9)
```

Primer 3*

```
Poziv: ghci 2.hs
Ulaz: listaParNepar [-9, -12, 2000]
IzLAZ:
[(-9, 0, 9), (-12, 2, 1), (2000, 2000, 0)]
```

Primer 4*

```
Poziv: ghci 2.hs
Ulaz: saberiParne [123, 10507]
IzLAZ:
2
```

Primer 5*

```
Poziv: ghci 2.hs
Ulaz: saberiNeparne [123, 10507]
IzLAZ:
170
```

3. **[Haskell (25%)]** Napisati biblioteku za pomoć studentskoj službi u arhiviranju i vođenju evidencije o studentima. Definirati tipove **StepenStudija** (osnovne, master, doktorske), **Student** (karakterisan brojem indeksa, imenom, prezimenom i stepenom studija) i **Rezultat** (svaki student ima opcioni rezultat predstavljen brojem poena). Obezbediti da se student može ispisati na standardni izlaz i porediti po jednakosti sa ostalim studentima po broju indeksa. Pretpostaviti da su sve unete vrednosti validne.

Primer 1*

```
Poziv: ghci 3.hs
Ulaz: print ( MkStudent "123/22" "Pera" "Peric" OsnovneStudije)
IzLAZ:
123/22 : Pera Peric : Osnovne studije
```

Primer 2*

```
Poziv: ghci 3.hs
Ulaz: stud1 = MkStudent "123/22" "Pera" "Peric" OsnovneStudije
stud2 = stud1{indeks = "321/22"}
stud3 = stud2{indeks = "123/22"}
stud2 == stud1
IzLAZ:
False
```

Primer 3*

```
Poziv: ghci 3.hs
Ulaz: stud1 = MkStudent "123/22" "Pera" "Peric" OsnovneStudije
stud2 = stud1{indeks = "321/22"}
stud3 = stud2{indeks = "123/22"}
stud3 == stud1
IzLAZ:
True
```

Primer 4*

```
Poziv: ghci 3.hs
Ulaz:
stud1 = MkStudent "123/22" "Pera" "Peric" OsnovneStudije
rezultatIspita = (stud1, Nothing)
rezultatIspita
IzLAZ:
(123/22 : Pera Peric : Osnovne studije,Nothing)
```

4. **[Prolog (15%)]** Napisati program koji dodeljuje različite cifre različitim karakterima tako da izraz:

TEN + TEN + NINE + EIGHT + THREE = FORTY bude zadovoljen.

Rezultat ispisati tako da se jasno vidi koja promenljiva je dobila koju vrednost.

5. **[Python (20%)]** Čistačica Mica sređuje i čisti kuće i stanove. Da bi sredila i počistila jedan stan potrebno joj je 1 sat, dok joj je za kuću potrebno 1.5 sati. Prilikom čišćenja, Mica potroši neku količinu deterdženta, 120ml po stanu, odnosno 100ml po kući. Mica zaradi 1000 dinara po svakom stanu, odnosno 1500 dinara po kući. Ukoliko Mica radi 40 sati nedeljno i ima 5l deterdženta na raspolaganju, koliko stanova i kuća je potrebno da očisti kako bi imala najveću zaradu? Sve rezultate ispisati na standardni izlaz.