

# I smer, Programske paradigme 2025, Primer

## 1. [Haskell (50p)]

Napisati biblioteku funkcija koje pružaju korisniku podršku za rad sa prostim brojevima. Implementirati:

- (a) (8p) Funkciju `delioci n`, čiji je tip: `delioci :: Int -> [Int]` koja vraća rastuće sortiranu listu pravih delilaca broja `n`.

Primer 1\*

```
|| Poziv: delioci 2
|| Izlaz:
|| []
```

Primer 2\*

```
|| Poziv: delioci 20
|| Izlaz:
|| [2,4,5,10]
```

Primer 3

```
|| Poziv: delioci 15
|| Izlaz:
|| [3,5]
```

- (b) (8p) Funkciju `prost n`, čiji je tip: `prost :: Int -> Bool` koja vraća `True` ukoliko je `n` prost, a `False` inače.

Primer 4\*

```
|| Poziv: prost 17
|| Izlaz:
|| True
```

Primer 6

```
|| Poziv: prost 9929
|| Izlaz:
|| True
```

Primer 5\*

```
|| Poziv: prost 4
|| Izlaz:
|| False
```

Primer 7

```
|| Poziv: prost 1
|| Izlaz:
|| False
```

- (c) (8p) Funkciju `generisiProste n`, čiji je tip: `generisiProste :: Int -> [Int]` koja generiše proste brojeve u intervalu `[2..n]`.

Primer 7\*

```
|| Poziv: generisiProste 10
|| Izlaz:
|| [2,3,5,7]
```

Primer 8\*

```
|| Poziv: generisiProste 6
|| Izlaz:
|| [2,3,5]
```

Primer 9

```
|| Poziv: generisiProste 1
|| Izlaz:
|| []
```

- (d) (2p) Kreirati tip podataka `Broj` koji se može konstruisati putem konstruktora `MkBroj` i sadrži dva atributa `tip` (`String`) i `vrednost` (`Int`) (npr. `(prost, 19)` i `(slozen, 10)`), kao i funkcije `tip` i `vrednost` koje omogućavaju pristup poljima `tip` i `vrednost`. Za kreirani tip instancirati klasu `Show` (podrazumevano) i klasu `Ord` tako da se brojevi porede po vrednosti.
- (e) (12p) Funkciju `sumaProstih a b`, čiji je tip: `sumaProstih :: [Broj] -> Int` koja vraća sumu prostih brojeva liste.

Primer 10\*

```
|| Poziv: sumaProstih [(MkBroj "prost" 2), (MkBroj "prost" 7), (MkBroj "slozen" 6), (MkBroj "prost" 7)]
|| Izlaz:
|| 16
```

Primer 12

```
|| Poziv: sumaProstih [(MkBroj "slozen" 14), (MkBroj "slozen" 1389)]
|| Izlaz:
|| 0
```

Primer 11\*

```
|| Poziv: sumaProstih [(MkBroj "slozen" 8), (MkBroj "slozen" 10), (MkBroj "prost" 11)]
|| Izlaz:
|| 11
```

- (f) (12p) Funkciju `faktori n`, čiji je tip: `faktori :: Broj -> [Int]` koja vraća listu uređenih parova koji predstavljaju proste faktore broja `n` kao i broj njihovih ponavljanja.

Primer 13\*

```
|| Poziv: faktori (MkBroj slozen 350)
|| Izlaz:
|| [(2,1),(5,2),(7,1)]
```

Primer 14\*

```
|| Poziv: faktori (MkBroj slozen 49)
|| Izlaz:
|| [(7,2)]
```

Primer 15

```
|| Poziv: faktori (MkBroj prost 11)
|| Izlaz:
|| []
```

2. **[Scala konkurentno (25p)]** Auto-servis *QuickFix* pruža usluge servisiranja automobila. U datoteci *automobili.txt* nalaze se podaci o automobilima koje je potrebno servisirati u sledećem formatu:

```
1  registarski_broj_vozila tip_servisa
```

Sa standardnog ulaza učitati broj dostupnih mehaničara. Posao svakog mehaničara izvršava se u posebnoj niti. Mehaničar uzima jedan po jedan automobil iz reda automobila koje je potrebno servisirati i započinje servis. Servisiranje može biti redovno održavanje (sa trajanjem koje je slučajan broj od 2 do 5 sekundi) ili popravka (sa trajanjem koje je slučajan broj od 5 do 10 sekundi). Nakon što je servis završen, mehaničar beleži podatak o trajanju servisiranja automobila. Kada svi mehaničari završe sa servisiranjem, ispisati ukupan broj automobila koji su prošli redovno održavanje i ukupan broj automobila koji su popravljeni, kao i prosečno vreme servisiranja.

Primer formata izlaza:

```
1  Redovno održavanje: 15 vozila
2  Popravka: 5 vozila
3  Prosecno vreme servisiranja: 4.2 sekunde
```

3. **[Prolog (25p)]** Ana je na rođendan pozvala četiri osobe koje su se unapred dogovorile da joj za poklon svako donese različitu stvar. Važi sledeće:

- a) Osoba koja se preziva Ivanović, će doneti narukvicu
- b) Osoba koja se preziva Marković se zove Marija i ima plavu kosu
- c) Ni Stefan, ni Aleksić neće doneti cveće
- d) Milan će doneti knjigu. Osoba koja se preziva Janković, a ne zove se Jana, će doneti tortu.

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

- 1) Napisati predikat `rodjendan(Lista)` koji rešava datu zagonetku i promenljivu `List`a unifikuje sa njenim rešenjem.
- 2) Napisati predikat `odgovori(X,Y)` u kom se promenljiva `X` unifikuje imenom osobe koja će doneti cveće, a promenljiva `Y` nazivom poklona koji će doneti osoba koja se preziva Marković.