

Programske paradigme (R-smer), praktični deo, JUN1

Na *Desktop*-u se nalazi arhiva **ppr_jun1_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_jun1_2024_JovanMaric_mr22205**. Svaki zadatak sačuvati u odgovarajućem poddirektorijumu.

1. **25% [Python]** U datoteci `laureati.json` nalazi se spisak dobitnika nobelove nagrade u `json` formatu. Dobitnici mogu biti ili osobe pojedinačno ili organizacije (poput *Crvenog krsta*). Na prvom nivou se nalazi istoimeno polje `laureati` sa nizovnom vrednošću kojom su opisani dobitnici nagrade. Treba zapaziti da se istorijski (iako retko) dešavalo da osoba ili organizacija osvajaju nagradu više od jednog puta. Po dobitniku se čuvaju naredna polja:

- `ime` – označava ime dobitnika ili naziv organizacije
- `prezime` – postoji samo u slučaju kada dobitnik nije neka organizacija
- `nagrade` – čija vrednost je niz sa elementima koji opisuju nagrade:
 - `godina`
 - `kategorija`
 - `udeoNagrade`

Napisati program kojem se kao prvi argument komandne linije prosleđuje naziv kategorije i to neki od narednih: ekonomija, fizika, hemija, knjizevnost, medicina ili mir. Program zatim treba da sve dobitnike iz te kategorije upiše u datoteku `laureati_NAZIV_KATEGORIJE.json` sortirano prema prezimenu i imenu (organizacije nakon osoba).

Napomena: za potrebe testiranja možete uporediti sopstveni rezultat sa .json datotekama iz direktorijuma 1-test ili pokrenuti skriptu ./1-test/test.sh.

2. **20% [Haskell]** Predrag je uložio vreme kako bi redukovao format iz prethodnog zadatka tako da podaci o pojedinačnoj nagradi budu dati u obliku uređene četvorki (`id`, `godina`, `kategorija`, `udeo`). Značenja tih polja su:

- `id` – identifikacioni broj osobe/organizacije kojoj je dodeljena nagrada
- `godina` – predstavlja godinu u kojoj je dodeljena nagrada
- `kategorija` – kategoriju iz koje je dodeljena nagrada
- `udeo` – predstavlja broj osoba/organizacija koje su podelile nagradu te godine

Pomozite Predragu tako što ćete napisati program koji je u stanju da čita te podatke. Potrebno je napisati funkciju `info :: Int -> [(Int, Int, String, Int)] -> [(Int, String, Int)]` koja za prosleđeni identifikacioni broj i listu uređenih četvorki vraća listu uređenih trojki (`godina`, `kategorija`, `udeo`) koje predstavljaju informacije o nagradama te osobe ili organizacije.

Primer 1

```
Poziv: ghci 2.hs
ULAZ:
info 6 [(6, 1911, "hemija", 1), (937, 2016, "knjizevnost", 1), (6, 1903, "fizika", 4)]
IZLAZ:
[(1911,"hemija",1),(1903,"fizika",4)]
```

Primer 2

```
Poziv: ghci 2.hs
ULAZ:
info 16 [(217, 1962, "mir", 1)]
IZLAZ:
[]
```

Primer 3

```
Poziv: ghci 2.hs
ULAZ:
info 633 [(712, 1994, "ekonomija", 3), (66, 1972, "fizika", 3), (937, 2016, "knjizevnost", 1), (633, 1961, "knjizevnost", 1)]
IZLAZ:
[(1961,"knjizevnost",1)]
```

NASTAVAK NA SLEDEĆOJ STRANI!

3. **20% [Haskell]** Dekadni broj $n > 2, n = \overline{c_1c_2...c_k}$ je *faktorion* ukoliko je ($n = \sum_{i=1}^k c_i!$). Potrebno je definisati funkciju `faktorion :: Int -> Bool` koja izvršava proveru da li je dati broj *faktorion*. Definirati i nularnu funkciju `prva2 :: [Int]` koja vraća listu prva dva takva broja.

Primer 1

```
|| Poziv: ghci 3.hs
|| ULaz:
||   faktorion 145
|| Izlaz:
||   True
```

Primer 2

```
|| Poziv: ghci 3.hs
|| ULaz:
||   faktorion 4321
|| Izlaz:
||   False
```

Primer 3

```
|| Poziv: ghci 3.hs
|| ULaz:
||   prva2
|| Izlaz:
||   [145,40585]
```

4. **20% [Prolog]** Napisati program koji dodeljuje različite vrednosti različitim karakterima tako da izraz:

PET
-DVA

TRI

bude zadovoljen. Minimizovati vrednost broja TRI i ispisati rezultat (potpisano) na sličan način kao u tekstu iznad.

5. **20% [Python]** Dati su apoeni od *10 centi*, *50 centi*, *1 dolara*, *10 dolara* i *100 dolara*. Napisati program kojem se kao prvi argument prosleđuje brojeva vrednost X , a na standardni izlaz se ispisuju sve moguće kombinacije u kojima se svaka od navedenih novčanih vrednosti pojavljuje barem jednom, a u zbiru su jednaki vrednosti broja X .

Primer 1

```
|| Poziv: python3 5.py 112
|| Izlaz:
||   10 centi: 5, 50 centi: 1, 1 dolar: 1, 10 dolara: 1, 100 dolara: 1
```

Primer 2

```
|| Poziv: python3 5.py 111
|| Izlaz:
||   nema resenja!
```

Primer 3

```
|| Poziv: python3 5.py 113.5
|| Izlaz:
||   10 centi: 5, 50 centi: 1, 1 dolar: 2, 10 dolara: 1, 100 dolara: 1
||   10 centi: 5, 50 centi: 3, 1 dolar: 1, 10 dolara: 1, 100 dolara: 1
||   10 centi: 10, 50 centi: 2, 1 dolar: 1, 10 dolara: 1, 100 dolara: 1
||   10 centi: 15, 50 centi: 1, 1 dolar: 1, 10 dolara: 1, 100 dolara: 1
```

Primer 4

```
|| Poziv: python3 5.py 111.6
|| Izlaz:
||   10 centi: 1, 50 centi: 1, 1 dolar: 1, 10 dolara: 1, 100 dolara: 1
```