Programske paradigme, primer ispita

1. 20% **[Python]** U datoteci `diplomirani.json` nalaze se podaci o studentima koji su završili Matematički fakultet u sledećem formatu (`P` je oznaka za prosek, `S` oznaka za smer):

```
[{"Student" : "Jovan Peric", "P" : 10.00, "GodStudiranja" : 4, "S" : "r" }, ...]
```

Napisati program koji među diplomiranim studentima pronalazi najboljeg kandidata za saradnika u nastavi na Matematičkom fakultetu. Sa standardnog ulaza se u zasebnim redovima učitavaju oznaka smera sa kog se traži kandidat i minimalna prosečna ocena, redom. Na standardni izlaz ispisati ime i prezime najboljeg kandidata sa traženog smera koji ispunjava uslov za saradnika. U slučaju iste prosečne ocene, prioritet ima student koji je kraće studirao, a u slučaju da ima više takvih studenata, ispisati ih sve sa napomenom da treba obaviti dodatni intervju. Ako ne postoji student koji zadovoljava kriterijum za izbor, ispisati poruku `Nema odgovarajućih kandidata`. Pretpostaviti da su podaci ispravni.

Primer 1*

```
DIPLOMIRANI.JSON
 [{"Student":"Luka Peric", "P":  10.00, "GodStudiranja":  4, "S":"r" },
 {"Student":"Marko Mitic", "P":  8.59, "GodStudiranja":  5, "S":"n" },
 {"Student":"Ana Mikic", "P":  9.78, "GodStudiranja":  4, "S":"r" },
 {"Student":"Mina Antic", "P":  10.00, "GodStudiranja":  4, "S":"v" }]
 r
 9.5
Izlaz:
 Luka Peric
```

Primer 2

```
DIPLOMIRANI.JSON
 [{"Student":"Jovan Peric", "P":  9.5, "GodStudiranja":  5, "S":"v" },
 {"Student":"Milica Mikic", "P":  9.2, "GodStudiranja":  4, "S":"v" },
 {"Student":"Milena Antic", "P":  9.5, "GodStudiranja":  4, "S":"v" }]
 v
 9.0
Izlaz:
 Milena Antic
```

Primer 3

```
DIPLOMIRANI.JSON
 [{"Student":"Jovan Peric", "P":  9.5, "GodStudiranja":  5, "S":"v" },
 {"Student":"Marko Mitic", "P":  9.5, "GodStudiranja":  5, "S":"v" }]
 v
 9.0
Izlaz:
 Potreban je dodatni intervju:
 Jovan Peric
 Marko Mitic
```

Primer 4

```
DIPLOMIRANI.JSON
 [{"Student":"Jovan Peric", "P":  8.5, "GodStudiranja":  5, "S":"m" },
 {"Student":"Marko Mitic", "P":  9.5, "GodStudiranja":  5, "S":"v" }]
 m
 9.0
Izlaz:
 Nema odgovarajućih kandidata.
```

2. 25% **[Haskell]** Podaci o poenima osvojenim na takmičanju `MATF21` su zadati kao `type Poeni = [Double]`, a podaci o osvojenim mestima kao `type Mesta = [[Char]]`.

   a) Definisati funkciju `rezultati :: Poeni -> Mesta` koja formira listu sa nazivom osvojenih mesta na takmičenju `MATF21` na osnovu broja poena. Ukoliko je `x` osvojen broj poena, takmičar je zauzeo `prvo` mesto za $x \geq 91$, `drugo` za $x \in [81, 91)$, `trece` za $x \in [71, 81)$, a lošiji rezultat se označava sa `pohvala`. Poeni za mesta su fiksni i ne mora da postoji svako mesto na takmičenju. Pretpostaviti da je broj osvojenih poena iz segmenta $[0, 100]$.

Primer 1*

```
Poziv: rezultati [91,68,78.5]
Izlaz:
 ["prvo","pohvala","trece"]
```

Primer 2

```
Poziv: rezultati [51,88.5,45,90.2]
Izlaz:
 ["pohvala","drugo","pohvala","drugo"]
```

Primer 3

```
Poziv: rezultati [70.5,68.5]
Izlaz:
 ["pohvala","pohvala"]
```

   b) Definisati funkciju `takmicar :: [[Char]] -> Poeni -> [([Char], [Char])]` koja na osnovu liste imena takmičara i poena koje su ostvarili na takmičenju formira listu parova oblika `(ime, mesto)`. Pretpostaviti da su u datim listama na istim pozicijama podaci koje treba upariti.

Primer 1*

```
Poziv: takmicar ["Ana","Jovana","Aleksa"] [51,78.50,92]
Izlaz:
 [("Ana","pohvala"),("Jovana","trece"),("Aleksa","prvo")]
```

Primer 2

```
Poziv: studenti ["Milos","Milica","Jovan","Ana"] [45,100,95.50,78]
Izlaz:
 [("Milos","pohvala"),("Milica","prvo"),("Jovan","prvo"),("Ana","trece")]
```

   c) Definisati funkciju `najbolji :: [([Char], [Char])] -> [[Char]]` koja na osnovu rezultata takmičenja datih u obliku liste parova `(ime, mesto)` formira listu imena takmičara koji su bili najbolji. Ukoliko su svi dobili pohvalu, treba formirati praznu listu.

Primer 1*

```
Poziv: najbolji [("Marko","trece"),("Jana","pohvala"),("Jovan","drugo")]
Izlaz:
 ["Jovan"]
```

Primer 2

```
Poziv: najbolji [("Ana","prvo"),("Mina","drugo"),("Aleksa","prvo")]
Izlaz:
 ["Ana","Aleksa"]
```

3. 20% **[Haskell]** Marija i Jana su osmislile igru `Kockica`. Jana zamisli broj od 1 do 6, a zatim Marija pet puta baca pravilnu kocku (kocka sa šest strana, na čijim stranama je zapisan po jedan broj od 1 do 6 bez ponavljanja) i beleži redom koje je brojeve dobila bacanjem kocke. Jana je pobedila ukoliko se broj koji je zamislila pojavio bar jednom pri bacanju čiji je redni broj iste parnosti kao zamišljen broj, u suprotnom, Marija je pobednik. Napisati funkcije koje određuju pobednika u igri `Kockica`.

   a) `bacanja :: Int -> [Int] -> [Int]` - za zamišljen broj `n` i listu dobijenih brojeva pri bacanju kocke pet puta pravi listu rednih brojeva bacanja pri kojima se dobio zamišljen broj `n`. Redni broj prvog bacanja kocke je 1. Pretpostaviti da su argumenti ispravni.

Primer 1*
```
Poziv: bacanja 2 [1,1,2,6,6]
Izlaz:
 [3]
```

Primer 2
```
Poziv: bacanja 5 [5,6,1,5,3]
Izlaz:
 [1,4]
```

Primer 3
```
Poziv: bacanja 6 [1,2,1,5,4]
Izlaz:
 []
```

   b) `kockica :: Int -> [Int] -> [Char]` - za zamišljen broj `n` i listu dobijenih brojeva pri bacanju kocke pet puta određuje da li je Jana ili Marija pobedik u igri `Kockica`. Pretpostaviti da su argumenti ispravni i da uvek Jana zamišlja broj, a Marija baca kocku.

Primer 1*
```
Poziv: kockica 2 [1,1,2,6,6]
Izlaz:
 "Marija"
```

Primer 2
```
Poziv: kockica 5 [5,6,1,5,3]
Izlaz:
 "Jana"
```

Primer 3
```
Poziv: kockica 6 [1,2,1,5,4]
Izlaz:
 "Marija"
```

4. 20% **[Prolog]** Četiri komšinice svakog meseca organizuju druženje i tom prilikom svaka od njih donese nešto od jela ili pića, pri čemu se unapred dogovore da svaka donese različito. Važi sledeće:

   a) Petrović će doneti voćnu tortu
   b) Ni Branković, ni Kristina neće doneti čokoladni kolač
   c) Ramona, koja se ne preziva Davidović, će doneti limunadu
   d) Marija će doneti vino
   e) Ana se preziva Aleksić

   Rešenje zagonetke je lista struktura koje jednoznačno određuju svaku komšinicu (ime, prezime, piće ili jelo koje donosi na druženje).

   1) Napisati predikat `druzenje(Lista)` koji rešava datu zagonetku i promenljivu `Lista` unifikuje sa rešenjem zagonetke.
   2) Napisati predikat `odgovori(X,Y)` u kom se promenljiva `X` unifikuje sa prezimenom komšinice koja će doneti čokoladni kolač, a promenljiva `Y` sa imenom komšinice koja se preziva Davidović.

5. 15% **[Prolog]** Napisati predikat `sat` koji ređa različite vrednosti iz intervala `[1,14]` u obliku peščanog sata:

```
x x x x
 x   x
  x x
 x   x
x x x x
```

   tako da brojevi na svakoj horizontali budu poređani u strogo opadajućem redosledu sa leva na desno i da suma brojeva na najnižoj horizontali bude jednaka 20. Predikat treba da ispisuje pronađene kombinacije u obliku prikazanog peščanog sata.