

Računalništvo (fizika): 4. izpit

5. 9. 2018

Naloge rešujte na strežniku Tomo, <https://www.projekt-tomo.si>.

Brskanje po spletni učilnici, nalogah na strežniku Tomo, video zapiskih predavanj in *uradni* dokumentaciji za Python *je dovoljeno*. Brskanje po spletnih straneh, ki vsebujejo rešitve nalog in nasvete o tem, kako se rešuje naloge, *ni dovoljeno*. Komunikacija s komerkoli, razen z asistentom, *ni dovoljena*.

Čas reševanja je 180 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (25 točk)

Računalnik HAL 9000 bi se rad znebil posadke na vesoljski ladji *Discovery One*. Najprej bi se rad znebil osebe, ki bi mu najlažje preprečila opravljanje misije. Podatke o posadki hrani v seznamu slovarjev, kjer vsak slovar predstavlja eno osebo. V slovarju ima zapisano ime (pod ključem `ime`) ter psihološki profil (pod drugimi ključi, ki se lahko med osebami razlikujejo), predstavljen s števili od 1 do 100. Na primer:

```
{ 'ime': "Dr. David Bowman",  
  'inteligenca': 92,  
  'ambicija': 88,  
  'empatija': 87,  
  'vodstvo': 95 }
```

HAL oceni nevarnost osebe s povprečjem vrednosti v psihološkem profilu. Na primer, Dr. Bowman dobi oceno $(92 + 88 + 87 + 95)/4 = 90.5$. HALu pomagajte poiskati najnevarnejšega člana posadke tako, da sestavite funkcijo `nevarni(posadka)`, ki sprejme seznam psiholoških profilov in vrne ime najnevarnejše osebe.

2. naloga (20+10 točk)

Sestavite generator `prestevanka(s)`, ki sprejme seznam $s = [s_0, \dots, s_{n-1}]$ in generira zaporedje

$$s_0, s_1, s_1, s_2, s_2, s_2, s_3, s_3, s_3, \dots$$

Se pravi, k -ti člen seznama s se ponovi $(k + 1)$ -krat.

Dodatna naloga [10 točk]: generator naj deluje tudi, če mu za s podamo poljuben drug iterator, vključno z neskončnim.

3. naloga (30 točk)

Ker Aljaža boli zob, v omrežju svojih znancev išče zobozdravnika. Pomaga si s programom v Pythonu. Osebe predstavi z razredom `Oseba`:

```
class Oseba():  
    def __init__(self, ime, znanci, zobar):  
        self.ime = ime          # ime osebe  
        self.znanci = znanci    # množica znancev  
        self.zobar = zobar      # ali je zobar  
    def __str__(self):  
        return "Oseba({0.ime}, {0.znanci}, {0.zobar})".format(self)
```

Aljažu pomagajte tako, da sestavite funkcijo `poisci_zobarja(o)`, ki sprejme osebo o in vrne ime zobozdravnika. Izmed vseh zobozdravnikov v omrežju znancev osebe o naj funkcija poišče tistega zobozdravnika, ki je osebi o najbližje glede na število vmesnih znancev. Če je takih zobozdravnikov več, naj funkcija vrne enega od njih. Pozor, omrežje znancev lahko vsebuje cikle!

4. naloga (25 točk)

Sestavite funkcijo `max_zmnozek(a, m)`, ki sprejme množico naravnih števil $a = \{a_1, \dots, a_n\}$ in vrne tako podmnožico množice a , da bo njen zmnožek čim večji, a ne bo presegal števila m .

Pri tej nalogi ni pomembno, kako učinkovita je rešitev. Pomembno je le, da najde pravi odgovor.