# Uvod v programiranje, 2008/09

## 2. kolokvij, bolonjska matematika

**1.** Med vajami iz računalništva bi se rad na asistentovem računalniku malo pohecal z besedilom in ga izpisal tako, da se črke ponovijo večkrat zaporedoma. V ta namen sestavi funkcijo za čudno množitev črk. Funkcija naj sprejme izvorni niz ter vrne "pokvarjenega", kjer se vsaka črka ponovi naključno od 1- do 6-krat. Znake, ki niso črke, naj funkcija pusti pri miru. Ni nujno, da je funkcija namenjena zgolj nagajanju asistentu ...

```
>>> veckratneCrke('Naš učitelj je najboljši.')
'NNNNaaššššš uuuuuučččiiiiittttteeellllljjjjjj jeeee naaaajjbooollllljjjjjššiii.'
```

Ko asistent opazi, da se njegov računalnik čudno obnaša, te seveda takoj osumi. Najprej se izgovarjaš, da je najbrž kriva pokvarjena tipkovnica, ko pa ti zagrozi z ukorom, se ponudiš, da boš napisal funkcijo, ki bo besedilo spravila nazaj v red. Sestavi torej še funkcijo, ki bo iz danega niza sestavila in vrnila nov niz, v katerem se **črke** ne bodo ponavljale.

```
>>> brezPonovitev('NNaššš uuučččiiiiittttteeellllljjjjjj jee naaajjboollljjšši...')
'Naš učitelj je najboljši...'
```

**2.** Tajnica na policijski postaji v Velenju s poročil o prometnih prekrških vsak dan pretipka kup imen. Podobno mora storiti tajnica v Kungoti. Občini bi zdaj radi združili svoja seznama imen in našli tiste ljudi, ki so grešili v obeh občinah. Ker se tajnice občasno zmotijo, smatramo, da dve imeni (eno iz Velenja in drugo iz Kungote) opisujeta isto osebo, če se bodisi povsem ujemata bodisi razlikujeta v enem samem znaku. Primer sta imeni 'Peter Novak' in 'Petar Novak'. Imen 'Petr' in 'Peter', po drugi strani, ne upoštevamo kot enakih. Velike in male črke strogo ločujemo. Sestavi funkcijo, ki sprejme dve množici imen in izpiše tista imena, ki se pojavijo v obeh množicah. Pri tem je dovoljeno razlikovanje v enem znaku, kot je bilo pravkar opisano. Če se neko ime v obeh seznamih pojavi v isti obliki, ga preprosto izpiši. Če se ime pojavi v dveh oblikah, izpiši v isti vrsti obe obliki, ločeni s poševnico.

```
>>> imena(
{'Tadej Nose', 'Gregor Mankoč', 'Peter Šubelj', 'Lovro Kašmerl', 'Eva Janež'},
{'Gregor Mankoč', 'Eva Hanež', 'Tina Weiss', 'Gregor Nose', 'Lovro KAšmerl'})
Eva Janež/Eva Hanež
Gregor Mankoč
Lovro Kašmerl/Lovro KAšmerl
```

**3.** Sestavi funkcijo, ki za parametra dobi slovar in celo število *n*. Vrednosti v slovarju so seznami. Funkcija naj iz slovarja odstrani vse tiste ključe, ki imajo za vrednost seznam, katerega dolžina ni deljiva z *n*, ostale vrednosti (tiste, katerih dolžina je deljiva z *n*) pa naj stisne tako, da sešteje po *n* zaporednih elementov seznama. Funkcija naj ne ustvarja novega slovarja, pač pa samo spremeni tistega, ki ga dobi za parameter. Pazi: v zanki, ki teče po slovarju, se dolžina slovarja ne sme spremeniti.

```
>>> s = {'a': [1, 2, 3, 4, 5, 6], 8: [], 'x': [8, 4, 2, 1], 't': [5], 0: [7, 4, 3]}
>>> slovar(s, 1)
>>> s
{'a': [1, 2, 3, 4, 5, 6], 8: [], 't': [5], 0: [7, 4, 3], 'x': [8, 4, 2, 1]}
>>> s = {'a': [1, 2, 3, 4, 5, 6], 8: [], 'x': [8, 4, 2, 1], 't': [5], 0: [7, 4, 3]}
>>> slovar(s, 2)
>>> s
{'a': [3, 7, 11], 8: [], 'x': [12, 3]}
>>> s = {'a': [1, 2, 3, 4, 5, 6], 8: [], 'x': [8, 4, 2, 1], 't': [5], 0: [7, 4, 3]}
>>> slovar(s, 3)
>>> s
{'a': [6, 15], 8: [], 0: [14]}
```