

Podan je slovar postaje, katerega ključi so imena postaj Biciklja, vrednosti pa slovarji z zemljepisnimi koordinatami postajališč, kapaciteto (število stojal) in trenutnim številom koles na postaji. (Glej slovar v testih.)

Podana je tudi funkcija razdalje(postaja1, postaja2), ki prejme imeni postaj in vrne razdaljo med njima.

1. Branje podatkov

Napišite funkcijo prenesi(), ki prebere podatke z naslova <https://ucilnica.fri.uni-lj.si/mod/resource/view.php?id=59478> (naslov je shranjen v spremenljivki URL v testih), jih predela v slovar v enaki obliki kot podani slovar postaje, in ga vrne. (Opomba: podatki so kopija resničnih, sproti osveževanih podatkov z naslova <https://api.icdecaux.com/vls/v3/stations?apiKey=frfk0jbxefqgnigez09tw4jvk37wyf823b5i1i&contract=ljubljana>, ki je med izpitom nedosegljiv.)

2. Najbližje 3

Avtor izpita ni uporabnik Biciklja (enkrat je bilo dovolj), ve pa, da na določeni postaji včasih ne moreš dobiti ali oddati kolesa, zato se je potrebno ozreti po bližnjih postajah. Napišite funkcijo najblizje3(postaja), ki za podano postajo poišče najbližje tri postaje in vrne seznam (treh) parov (razdalja, ime). Seznam naj bo urejen po razdaljah.

Klic najblizje3("ŽIVALSKI VRT") vrne [(0.73687236710574, 'VIŠKO POLJE'), (0.9639016462683163, 'TEHNOLOŠKI PARK'), (1.0760703913614853, 'SOSESKA NOVO BRDO')].

3. Tabela postaj

Napišite funkcijo zapisi(ime_datoteke), ki v podano datoteko zapiše imena postaj (na 40 mest), ki ji sledi število razpoložljivih koles in število mest. Zapis mora biti urejen po abecedi imen postaj (šumniki pa bodo na koncu; uporabite običajno urejanje v Pythonu) in v obliki, katere izsek kaže okvirček.

KONGRESNI TRG-ŠUBIČEVA ULICA	9/20
KOPALIŠČE ILIRIJA	0/20
KOPALIŠČE KOLEZIJA	19/20
KOPRSKA ULICA	3/8
KOSEŠKI BAJER	14/20

4. Prevozi

Imamo matriko v numpyju, ki predstavlja podatke o prehodih koles med postajami v nekem časovnem obdobju: element z indeksoma [i][j] pove, koliko koles je bilo prevzetih na i-ti postaji in vrnjenih na j-to. Matrika na sliki nam pove, da so na postaji 1 prevzeli 3 kolesa, ki so jih vrnili na postajo 1; 7 koles, ki so jih vrnili na postajo 2; in 5 koles, ki so jih vrnili na 4.

```
[ [1, 2, 1, 0, 3, 0],
  [0, 3, 7, 0, 5, 0],
  [0, 1, 0, 2, 3, 0],
  [0, 3, 0, 0, 4, 2],
  [0, 1, 5, 0, 0, 3],
  [0, 0, 5, 1, 3, 0] ]
```

- Skupno so s postaje 1 odpeljali 15 koles, pripeljali pa so jih (kje to vidimo?) 10. Na postaji 1 je pet koles manj, kot jih je bilo prej. Napišite funkcijo spremembe(prevozi), ki prejme takšno matriko in vrne matriko s toliko elementi, kolikor je postaj. Vsak element pove, koliko koles več (pozitivna števila) oz. manj (negativna) kot prej je na postaji. Za matriko s slike vrne [-6, -5, 12, -6, 9, -4].
- S postaje 1 na postajo 4 so pripeljali 5 koles, s postaje 4 na 1 pa samo 1 kolo. Razlika je torej 4. S postaje 4 na 5 so jih pripeljali 3, obratno prav tako 3. Razlika je 0. Največja razlika je med postajama 1 in 2: v eno smer so jih pripeljali 1, v drugo 7, razlika je 6. Napišite funkcijo asimetrija(prevozi), ki vrne par postaj z največjo razliko. V gornjem primeru mora vrniti [1, 2] (ali [2, 1]). Če je parov z največjo razliko več, smete vrniti poljubnega.

Zaželeno je, da nalogo rešite s spretno uporabo numpyja – v vrstici ali dveh, brez kakšnih zank v Pythonu.

5. Vračanje

Upravljalci Biciklja morajo prevažati kolesa nazaj. Napišite funkcijo vračanje(razlike), ki prejme razlike, kakršne vrača funkcija spremembe, in vrne matriko, ki pove, koliko koles je potrebno prepeljati s katere postaje na katero, da bo na vsaki postaji spet toliko koles, kot jih je bilo v začetku. Pravilnih rešitev je neskončno mnogo, funkcija naj vrne eno od njih.

Funkcija vračanje dela torej ravno nasprotno kot funkcija spremembe: za podane spremembe vrne matriko, ki bi dala takšne spremembe (le z negativnim predznakom). Test deluje tako, da si izmisli razlike in pokliče vašo funkcijo

```
razlike = np.array([-6, -5, 12, -6, 9, -4])
nazaj = vračanje(razlike)
```

nato pa preveri, da so vsi elementi nazaj pozitivni in da matrika razlike + spremembe(nazaj) vsebuje same ničle. Pri tem predpostavlja, da ste pravilno rešili četrto nalogo.

To niti ni naloga iz numpyja. Napisati bo potrebno napisati nekaj zank; numpy najbrž ne bo v posebno pomoč.