

Nekatere naloge zahtevajo branje datotek. Datoteke se bodo pojavile, ko prvič požnete teste.

1. Analiza sedežnega reda

Na sliki je zaslonski posnetek visoko specializirane programske opreme, ki jo letalske družbe uporabljajo za razporejanje potnikov. Na primeru s slike, na sedežu 2D sedi potnica z imenom Cilka. Programska oprema izvozi podatke v datoteko z vsebino, kot jo kaže okvirček pod sliko (gre seveda zgolj za primer, v splošnem so imena in razpored lahko tudi drugačni).

Napišite naslednje funkcije:

- `popularnost(ime_dat)` prejme ime datoteke z razporedom in vrne seznam ali tabelo v numpy s šestimi elementi, ki ustrezajo šestim stolpcem (A-F). Vrednosti povedo, koliko mest v posameznem stolpcu je zasedenih. Za gornji primer vrne `[6, 5, 4, 5, 5, 8]`.
- `naj_stolpec(ime_dat)` vrne najpopularnejši stolpec. V gornjem primeru vrne "F". Če si mesto najpopularnejšega deli več stolpcev, sme vrniti poljubnega.
- `po_imenih(ime_dat)` vrne slovar, katerega ključi so imena potnikov, vrednosti pa število potnikov s tem imenom. V gornjem primeru vrne `{ 'Ana': 8, 'Berta': 7, 'Cilka': 6, 'Dani': 2, 'Fanči': 7, 'Greta': 3 }`.
- `naj_stolpec_ime(ime_dat, ime)` vrne najpopularnejši stolpec med potniki s podanim imenom. Klic `naj_stolpec("razpored.csv", "Fanči")` vrne "C", ker največ potnikov z imenom Fanči sedi v stolpcu C.
- `prazne_vrste(ime_dat)` vrne seznam ali tabelo s številkami praznih vrst. V gornjem primeru vrne `[10, 11]`.

Namiga: Letalske družbe številčijo vrste tako, da ima prva vrsta številko 1, ne 0. "ABCDEF"[2] vrne "C".

	A	B	C	D	E	F
1	Ana	Berta	Fanči		Dani	Cilka
2		Dani		Cilka		Ana
3	Berta			Ana	Fanči	
4	Ana		Greta		Berta	Fanči
5		Greta		Fanči	Ana	Cilka
6	Ana		Fanči		Cilka	Berta
7		Greta		Fanči		
8	Ana		Fanči	Cilka		Berta
9						Berta
10						
11						
12	Ana	Berta				Cilka

```
Ana,Berta,Fanči,,Dani,Cilka
,Dani,,Cilka,,Ana
Berta,,,Ana,Fanči,
Ana,,Greta,,Berta,Fanči
,Greta,,Fanči,Ana,Cilka
Ana,,Fanči,,Cilka,Berta
,Greta,,Fanči,,
Ana,,Fanči,Cilka,,Berta
,,,,Berta
,,,,,
,,,,,
,,,,,
Ana,Berta,,,Cilka
```

2. Prehranska problematika

Napišite funkcijo `posebne_zelje(zelje)`, ki prejme slovar, katerega ključi so oznake posebnih želja glede hrane, na primer VEG za vegetarijansko, GLU za brezglutensko in tako naprej. Pripadajoče vrednosti so sezname sedežev, kjer sedijo potniki, ki jim je potrebno postreči takšno hrano.

```
zelje = {
    "GLU": ["2B", "4F"],
    "LAK": ["1A", "2B", "5F", "8A"],
    "ORE": ["2B", "1A"],
    "VEG": ["2B", "4C", "5B", "7B", "5E"]}
```

Funkcija `posebne_zelje` mora vrniti število potnikov s posebnimi željami. Ne spreglejte, da lahko potnik prosi tudi za brezglutenski vegetarijanski obrok brez laktoze.

3. Navodila za stevardese

Napišite funkcijo `navodila(ime_razpored, zelje, ime_navodila)`, ki prejme ime datoteke z razporedom (kot v 1. nalogi) in slovar želja, tretji argument pa je ime datoteke, ki jo mora ustvariti ta funkcija in vanjo zapisati navodila za stevardese v obliki, ki jo kaže okvirček na desni.

Za polne točke naj bodo tako imena kot kode želja (GLU, LAK ...) urejene po abecedi. Osebe z enakim imenom uredite po sedežih, katerih oznake – da bo lažje – obravnavajte kot nize: 14A naj bo pred 5A, ker je niz "14A" po abecedi prej kot "5A".

```
Ana (1A): LAK, ORE
Ana (5E): VEG
Ana (8A): LAK
Cilka (5F): LAK
Dani (2B): GLU, LAK, ORE, VEG
Fanči (4F): GLU
Greta (4C): VEG
Greta (5B): VEG
Greta (7B): VEG
```

4. Prestopi

Imamo slovar `leti`. Ključi so kode letov, vrednosti pa sezname letov, na katere je mogoče (neposredno) prestopati s tega leta. Primer je v testih.

Napišite funkcijo `mozni_leti(let, prestopanj)`, ki prejme kodo nekega leta in največje dovoljeno število prestopanj (lahko tudi 0). Vrniti mora množico kod vseh letov (vključno s podanim), na katere lahko pride potnik, ki se vkrci na podani let in prestopa največ tolikokrat, kolikor določa argument.

Lažja različica (za manj točk): napišite funkcijo `mozni_leti(let)`, ki vrne isto, vendar brez omejitve števila prestopanj.