

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

## 1. Skupna cena

Napišite funkcijo `cena(zasedenost, cene)`, ki prejme numpyjevo matriko `zasedenost` tipa `bool`, ki ima šest stolpcev in toliko vrstic, kolikor vrst ima letalo ter vsebuje `True` na mestih, ki ustrezajo zasedenim sedežem in `False` na mestih, ki ustrezajo prostim. Poleg tega dobi Pythonov seznam `cene`, ki vsebuje cene vozovnic za sedeže v prvih treh stolpcih, na primer `[200, 175, 193]`, če sedeži ob oknu (stolpec 0) stanejo 200, sedeži na sredini (stolpec 1) 175 in sedeži ob hodniku (stolpec 2) 193 evrov. Cene sedežev v drugih treh stolpcih so zrcalno enake tem (na primer 193, 175 in 200).

Funkcija mora vrniti vsoto cen vseh prodanih vozovnic. Funkcija mora biti napisana v numpyju. Točkovanje bo upoštevalo eleganco rešitve.

## 2. Število letov

Imamo tabelo, v kateri so pari poletov in pristankov, na primer `[(360, 420), (400, 580), (485, 1300), (490, 600), (600, 800), (650, 720), (700, 800), (780, 800), (930, 1380), (950, 1380), (950, 1000), (1100, 1200)]`. Časi so v minutah od polnoči. Tako prvi par, `(360, 420)` pomeni da neko letalo vzleti ob 6. uri zjutraj in pristane ob 7. uri. Vsa letala vzletajo in pristajajo na istem letališču, saj gre za turistične prelete.

Nek turist bi rad v istem dnevu naredil čim več poletov. Pri tem pa mora med pristankom nekega leta in odletom naslednjega miniti vsaj 60 minut, da gre vmes po sendvič. Napišite funkcijo `stevilo_letov(prvi, leti)`, ki prejme prvi let, na katerega bo šel in seznam vseh letov, ter vrne največje število poletov, ki jih lahko opravi v enem dnevu.

- Za gornje podatke klic `stevilo_letov((360, 420), leti)` vrne 5, saj lahko gre na lete `(360, 420)`, `(490, 600)`, `(700, 800)`, `(950, 1000)`, `(1100, 1200)`.
- Klic `stevilo_letov((600, 800), leti)` vrne 3, saj bo ujel samo `(600, 800)`, `(950, 1000)`, `(1100, 1200)`.

## 3. Razpored

Neka letalska družba sedežev ne dodeljuje vnaprej, temveč lahko potniki zgolj izrazijo željo. Potnike nato vkrcavajo po seznamu. Vsak potnik se usede na zeleni sedež, če je ta že zaseden, pa se pomika nazaj po vrstah, dokler ne naleti na vrsto, v kateri je sedež v zelenem stolpcu prazen, ter ga zasede. Če so, na primer, že zasedeni sedeži 12C, 13C in 14C, se bo potnik, ki bi želel sedeti na 12C, usedel na 15C. Letalska družba uporablja dolga letala, ki imajo 130 vrst, obenem pa ji ne gre prav dobro (čudno, čudno), zato smete predpostaviti, da se bo po tem postopku našel sedež za vsakogar.

Napišite funkcijo `razpored(seznam)`, ki dobi seznam s pari `(ime_potnika, zeleni_sedez)` in vrne nov seznam, ki je enak temu, le da so sedeži zamenjani z sedeži, ki jih bodo potniki dejansko dobili.

Pomoč: če je `c` črka A, B, C, D, E ali F, je `ord(c) - 65` enak 0, 1, 2, 3, 4 oz. 5. Mogoče vam pride prav.

## 4. Ravnotežje

Napišite funkcijo `ravnotezje(ime_datoteke)`, ki prejme ime datoteke s podatki, ločenimi z vejico. V prvi vrstici so zapisana imena stolpcev. Eden od njih se imenuje `sedez`; sedeži so zapisani v običajni obliki, npr. 12F ali 1C ali 128E. Funkcija mora vrniti razliko med številom potnikov, ki sedijo na desni (stolpci D, E, F) in levi (stolpci A, B, C) strani letala. Če, recimo, na desni sedi 7 potnikov več, vrne 7; če je 7 potnikov več na levi, vrne -7.

## 5. Leti

Napišite funkcijo `vozni_redi(potniki, ime_datoteke)`.

Argument `potniki` je seznam z imeni potnikov, številkami letov in pari (ura, minuta) za odhod in prihod. Primer podatkov je, recimo `[("Ana Argon", "LH2832", (12, 10), (13, 20)), ("Berta Bor", "U0391", (15, 5), (20, 30)), ("Cilka Cankar", "LH192", (7, 0), (12, 30))]`. Funkcija mora v datoteko s podanim imenom zapisati tabelico z imeni, leti in časi, kot jo kaže slika na desni.

Ana Argon	LH2832	12:10-13:20
Berta Bor	U0391	15:05-20:30
Cilka Cankar	LH192	7:00-12:30

Točno obliko - število presledkov - razberite iz testov.