

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

1. Delovna sila

Šopi banan prihajajo po tekočem traku. Delavec jih zloga v škatle. Če bi teža banan v škatli, skupaj z naslednjim šopom preseгла 20 kg, preda škatlo naprej in vzame naslednjo, prazno škatlo.

Napišite funkcijo pakiranje(vhod, izhod), ki prejme imeni dveh besedilnih datotek. V prvi so teže zaporednih šopov banan (necela števila, float). V drugo mora funkcija izpisati teže paketov na eno decimalno natančno. Za primer glejte datoteki vhod.txt in izhod.txt, ki se bosta pojavili, ko prvič poženete teste.

Ne pozabite na zadnjo škatlo.

2. Varna naložba

Škatle naložijo na (manjše) tovornjake (pri pakiranju morda ne upoštevajo navodil iz prejšnje naloge). Škatle so vedno naložene štiri nivoje visoko. Naložbo zapišejo v datoteko po nadstropjih. V prvi vrstici datoteke so zapisane teže škatel v prvi vrsti na tleh. Sledi druga vrsta škatel na tleh, nato tretja in tako naprej. Nato ponovijo isto za drugo, tretje in četrto nadstropje. Če je v vsaki vrsti 5 škatel in imamo 8 vrst, bo v datoteki 8x4 vrstic s 5 elementi.

Naložba v tovornjak je varna, če ustreza naslednjim pogojem:

- Skupna teža škatel v posamičnem »stolpcu« štirih škatel, naloženih ena na drugo, je največ 80 kg.
- Razlika med težo na levi in desni strani je največ 10 % celotne teže tovora. Če je število škatel v vrsti liho, srednje ne upoštevajte. (Namig: če je škatel 7, primerjate prve 3 in zadnje 3. Pythonovo indeksiranje...)
- Razlika med težo v sprednjem in zadnjem delu je največ 10 % celotne teže tovora. Če je ena za drugo liho število škatel, srednje ne upoštevate.

Če je celoten tovor težak 4800 kg, sme biti razlika med levo in desno tovoranjaka ter med sprednjo in zadnjo stranjo največ 480 kg.

Napišite funkcijo varnost(ime_datoteke), ki vrne True, če je naložba, opisana v datoteki, varna, in False, če ni.

Nasvet: Klic `help(np.reshape)` pokaže dokumentacijo funkcije `np.reshape`. Prav utegne priti tudi `np.genfromtext`.

Pazi: `-x // 2` je `(-x) // 2` in ne `-(x // 2)`. Zato je `-5 // 2` enako `-3` in ne `-2`.

3. Dolgovi

Napišite funkcijo pregled(narocila, dir_prevozi, porocilo).

- Prvi argument je ime datoteke z naročili banan v obliki .json. Vsebuje slovar, katerega ključi so naročniki, vrednosti pa teža naročenih banan.
- Drugi argument je ime direktorija (imenika), ki vsebuje datoteke .json, v kateri so opisane pošiljke banan. Obliko pogled sam(a). Isti naročnik lahko dobi več dostav. (Imena datotek v direktoriju vrne `os.listdir`.)
- Tretji argument je ime datoteke, v katero moraš (prav tako v obliki json) zapisati slovar, katerega ključi so naročniki, vrednosti pa teža banan, ki jih jim je potrebno še dostaviti. Slovar naj ne vsebuje naročnikov, ki smo jim že dostavili naročeno količino ali celo količino, ki presega naročeno.

4. Vmesno poročilo

Napišite funkcijo porocilo(narocila, dostava), ki prejme slovar z naročili (ključi so naročniki, vrednosti količina naročenih banan) in slovar s skupno količino dostavljenih banan (ključi so naročniki, vrednosti količina dostavljenih banan). Vsak ključ iz drugega slovarja se seveda pojavi tudi v prvem (banane dostavljamo le naročnikom), obratno pa ni nujno (nekaterim naročnikom morda nismo dostavili še ničesar).

Naročnik	Naročeno	Dostava	Razlika
Ana	1000	1000	0
Berta	2000	800	-1200
Cilka	300	0	-300
Dani	1200	1280	80

Funkcija mora vrniti niz (ne izpisovati, vrniti niz!) v obliki, ki jo kaže slika na desni. Vsebina je očitna. Vsi stolpci so široki 10 znakov.