

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

## 1. Statistika

Klicni center beleži podatke o številu klicev v numpyjevo tabelo z  $24 * 60$  elementi. Element z indeksom  $i$  pove število klicev v  $i$ -ti minuti dneva. Napišite naslednje funkcije:

- `po_urah(a)` prejme tabelo, kot jo opisujemo zgoraj in vrne tabelo s 24 elementi, ki vsebujejo število klicev v posamezni uri dneva
- `naj_ura(a)` vrne uro z največ klici (npr. 5, če je največ klicev med peto in šesto uro).
- `brez_klicev(a)` naj vrne število minut, ko ni bilo klicev (torej število ničelnih elementov podane tabele `a`).

Pri ocenjevanju bom upošteval tudi vašo spretnost uporabe knjižnice `numpy`.

## 2. Izpis

Napišite funkcijo `izpis(a)`, ki prejme tabelo s 24 elementi, ki predstavljajo število klicev po urah in vrne niz, oblikovan natančno tako (do presledka enako!), kot kaže slika.

Prvi številki povesta uro dneva; začetek je poravnano desno, konec levo. Sledi število klicev v tej uri in "histogram", pri čemer vsak znak # predstavlja (dopolnjenih 10) klicev; pri 66, na primer, imamo 6 znakov #.

```
0 - 1    20 ##
1 - 2    34 ###
2 - 3    66 #####
3 - 4    82 #####
4 - 5   114 #####
5 - 6   125 #####
6 - 7   204 #####
7 - 8   272 #####
8 - 9   364 #####
9 - 10  453 #####
10 - 11 522 #####
(in tako naprej do)
23 - 24  36 ###
```

## 3. Pravilnost

Napišite funkcijo `preveri(ime_datoteke)`, ki prejme ime datoteke, ki vsebuje besedilo, kakršnega vrne prejšnja funkcija. Funkcija vrne `True`, če je izpis pravilen in `False` če ni.

Predpostaviti smete, da datoteka vsebuje pravilno število (24) vrstic in da so oblikovane pravilno. Preveriti pa mora, da so pravilni začetki in konci (torej, da si ure sledijo v vrstnem redu 0 – 1, 1 – 2, 2 – 3 in tako naprej) ter da se število znakov # ujema s številom klicev (deljenim z 10).

## 4. Operaterji

V datoteki v formatu `csv` so shranjeni prejeti klici – za vsakega vemo, kateri operater ga je sprejel, koliko minut je trajal, kdaj se je začel in za kakšno vrsto klica je šlo (zadnja dva podatka sta nepomembna).

Napišite funkcijo `obremenitve(ime_datoteke)`, ki vrne slovar, katerega ključi so imena operaterjev, vrednosti pa število minut, ki jih je operater preživel na klicih. Za gornji primer vrne `{"Ana": 13, "Berta": 15, "Cilka": 9, "Dani": 5, "cbgadbbb": 5}`.

```
operater,dolzina,zacetek,tip
Ana,10,123,I
Berta,2,453,I
Cilka,5,134,O
Berta,10,500,T
Ana,3,135,I
Dani,5,245,T
Berta,3,573,I
Cilka,4,262,I
cbgadbbb,5,157,T
```

Napišite tudi funkcijo `naj_obremenjeni(ime_datoteke)`, ki vrne ime najbolj obremenjenega glede na skupno število minut; v gornjem primeru vrne "Berta". Če je najbolj obremenjenih več, lahko vrne poljubnega med njimi.

## 5. Hierarhija

Mogoče je klicnih centrov več, postavljeni pa so v neko hierarhijo. Osnovna sta Koper in Maribor. Celje je v hierarhiji podrejeno Mariboru. Ptuj tudi. Ormož je podrejen Ptujju. Ljutomer tudi Ptujju. Laško Celju. Postojna Kopru. Vse doslej naštetu bi lahko shranili v takšen seznam: `[("Celje", "Maribor"), ("Ptuj", "Maribor"), ("Ormož", "Ptuj"), ("Ljutomer", "Ptuj"), ("Laško", "Celje"), ("Postojna", "Koper")]`. Krajev je seveda veliko več in hierarhija je lahko poljubno globoka.

Napiši funkcijo `centri(odvisnosti)`, ki prejme seznam, kot je gornji in vrne slovar obrnjenih parov: ključi so kraji, vrednosti pa množice krajev, ki so mu podrejeni. V gornjem primeru mora vrniti `{"Maribor": {"Celje", "Ptuj"}, "Ptuj": {"Ormož", "Ljutomer"}, "Celje": {"Laško"}, "Koper": {"Postojna"}}`. Kraji, ki jim ni podrejen noben kraj, naj se ne pojavijo kot ključi.

Pomoč: to ni nujno naloga iz rekurzije.