

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 json 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. Primer datoteke je na sliki (prelom vrstic je lahko tudi drugačen!).

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, "eaaudgef": 5}`.

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

Pomagajte si s funkcijo `json.load(datoteka)`. Funkcija prejme odprto datoteko, ne niza z imenom!

5. Brez klicev

Napišite funkcijo `obdobje_brez(a)`, ki prejme takšen argument kot funkcije iz prve naloge, in vrne začetek in konec najdaljšega obdobja brez klicev. Če so vsi elementih na indeksih od, na primer, 150 do (vključno) 180 enaki 0 in je to tudi najdaljše zaporedje ničel, mora funkcija vrniti (150, 180).

Da bo reševanje lažje, so v testih tudi trije primeri s tabelami, ki nimajo $24 * 60$ temveč le 13 števil.

```
[{"operater": "Ana", "dolzina": 10, "zacetek": 123, "tip": "I"},
 {"operater": "Berta", "dolzina": 2, "zacetek": 453, "tip": "I"},
 {"operater": "Cilka", "dolzina": 5, "zacetek": 134, "tip": "O"},
 {"operater": "Berta", "dolzina": 10, "zacetek": 500, "tip": "T"},
 {"operater": "Ana", "dolzina": 3, "zacetek": 135, "tip": "I"},
 {"operater": "Dani", "dolzina": 5, "zacetek": 245, "tip": "T"},
 {"operater": "Berta", "dolzina": 3, "zacetek": 573, "tip": "I"},
 {"operater": "Cilka", "dolzina": 4, "zacetek": 262, "tip": "I"},
 {"operater": "eaaudgef", "dolzina": 5, "zacetek": 157, "tip": "T"}]
```