

Rešitev oddajte prek Učilnice. Za rešitev naloge lahko dobite določeno število točk, **tudi če ne prestane testov**. Funkcija, ki prestane vse teste, **še ni nujno pravilna**. Upošteva se tudi kvaliteta rešitve.

Dovoljena je uporaba vseh materialov na Učilnici in druge literature na poljubnih medijih. Prepovedan je dostop do vseh drugih spletnih strani in vsaka oblika komunikacije, razen s profesorjem oz. asistentom.

## 1. Briši ponovitve

Napiši funkcijo `brisi_ponovitve(s)`, ki prejme seznam pozitivnih celih števil in ga popravi takole: v seznamu pusti največ eno enko, največ dve dvojki, največ tri trojke, štiri štirice in tako naprej (do poljubno velikih števil). Vse nadaljnje ponovitve pobriše. Funkcija ne vrne ničesar, temveč spreminja podani seznam.

```
>>> a = [1, 3, 4, 1, 3, 2, 2, 3, 5, 3, 2, 3, 4]
>>> brisi_ponovitve(a)
>>> a
[1, 3, 4, 3, 2, 2, 3, 5, 4]
```

## 2. Največji v vseh

Napiši funkcijo `najvecji_v_vseh(s)`, ki dobi seznam seznamov števil, `s`. Vrniti mora največje število, ki se pojavi v vseh (pod)seznamih. Če sezname nimajo nobenega skupnega elementa, funkcija vrne `None`. Klic `najvecji_v_vseh([[5, 1, 2, 3], [3, 1, 8], [42, 5, 3, 1]])` vrne 3. V vseh treh seznamih se namreč pojavita 1 in 3; 3 je večja.

Pričajak dolge sezname! Če je tvoja rešitev počasna in testi zmrznejo, jih prekini, sicer bo računalnik vedno počasnejši.

## 3. Vrstni red

Napiši funkcijo `vrstni_red(ime_datoteke)`, ki prejme ime datoteke z rezultati nekega tekmovanja v naslednji obliki:

```
Ana Anžič: 5:12
Berta Bertolin: 4:48
Cilka Centrih: 5:05
Dani Dolinar: 10:12
Ema Evelina Estrih: 4:45
```

Funkcija vrne seznam imen tekmovalcev, urejen po časih od manjših proti večjim. Če imata dva tekmovalca enak čas, ju uredi po abecedi. Za gornji primer vrne `["Ema Evelina Estrih", "Berta Bertolin", "Cilka Centrih", "Ana Anžič", "Dani Dolinar"]`.

## 4. Preveri vsoto

Napiši rekurzivno funkcijo `preveri_vsoto(s, n)`, ki prejme seznam števil in *domnevno* vsoto teh števil. Funkcija vrne `True`, če je *domnevna* vsota točna in `False`, če ni.

Rekurzivna naj bo prav ta funkcija s prav temi argumenti. Ne piši pomožnih funkcij. Duhovitosti v smislu rekurzivne funkcije, ki izračunajo vsoto, ne veljajo.

## 5. Čakalnica

Napiši razred `Čakalnica` z naslednjimi metodami:

- `prihod(ime, cas)` zabeleži, da je ob podanem času v čakalnico vstopila oseba s podanim imenom. Predpostaviti smeš, da se metoda pokliče v trenutku, ko oseba vstopi. To pomeni, da bo čas pri vsakem klicu večji od časa pri predhodnem klicu;
- `cakajocih()` vrne število oseb v čakalnici;
- `naslednji(cas)` vrne ime naslednje osebe, ki je na vrsti. Osebe pridejo na vrsto v takšnem vrstnem redu, v kakršnem so prihajale. Če je čakalnica trenutno prazna, vrne `None`. Metoda kot argument prejme trenutni čas; potrebuje ga zaradi naslednjih dveh metod;
- `skupni_cas_cakanja()` vrne skupni čas čakanja vseh oseb, ki so že bile na vrsti (ne pa tistih, ki so še v čakalnici);
- `povprečni_cas_cakanja()` vrne povprečni čas čakanja za vse osebe, ki so že bile na vrsti (ali 0, če ni bila na vrsti še nobena oseba).

Konstruktor napiši, če meniš, da ga potrebuješ. Argumentov pa ne sme imeti.

Čas je vedno podan oz. vrnjen v urah. Če nekdo pride ob 10:45, je `cas` enak 10.75 (10 in tri četrt). Če je skupni čas čakanja uro in pol, metoda `skupni_cas_cakanja` vrne 1.5 (ne 1.30).