

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. Bingo

Igra Bingo poteka tako, da ima vsak igralec listek z nekaj številkami. Voditelj žreba številke. Če se izžrebana številka nahaja na igralčevem listku, jo ta prečrta.

Napiši funkcijo `bingo(listki, vrstni_red)`, ki prejme listke, ki jih imajo igralci (seznam seznamov števil) in vrstni red, v katerem bodo žrebane številke (recimo, da smo izumili časovni stroj). Funkcija naj pove, po koliko izžrebanih številkah bodo na enem od listkov prečrtane vse številke.

Funkcija naj ne spreminja podanih seznamov!

## 2. Nogavice brez para

Ker smo moški, vemo, barvno slepi in ne ločimo med črnimi in (domnevno!) temno modrimi nogavicami, se lahko najdemo tako, da na vsako nogavico prišijemo številko. Obe nogavici iz para imata enako številko.

Recimo, da iz pranja dobimo nogavice [15, 12, 17, 18, 12, 4, 15, 4, 7]. Nekaj nogavic je kot vedno seveda brez para: tokrat so to nogavice 17, 18 in 7. Nato pride naslednja žehta; v njej so, recimo, [3, 4, 3, 5, 17, 7]. Nogavice, ki se jim je našel par (v tem primeru 17 in 7), pokombiniramo.

Napiši funkcijo `brez_parov(s, t)`, ki prejme dva takšna seznama in vrne dve številki, ki povesta, koliko nogavic iz vsake žehte je še vedno brez para. V gornjem primeru vrne (1, 2), saj je brez para ena nogavica iz prve žehte (7) in dve iz druge (4 in 5).

## 3. Seznam iz parov

Nekateri računalniški jeziki nimajo seznamov, temveč le pare – v Pythonu so to terke z dvema elementoma. Seznam predstavimo tako, da naredimo par, ki vsebuje prvi element in ostanek. Ostanek je spet par, ki vsebuje prvi element ostanka in ostanek ostanka. Namesto ["Ana", "Berta", "Cilka", "Dani", "Ema"] tako pišemo ("Ana", ("Berta", ("Cilka", ("Dani", ("Ema", None))))). Prazen seznam predstavimo z None (ne s prazno terko!). Zadnji element zadnje terke je, ko vidimo v gornjem primeru, tudi None, kar pomeni, da je ostanek (ostanka ostanka ostanka) prazen seznam.

Napiši nerekurzivno funkcijo `v_seznam(pari)`, ki prejme seznam v obliki parov in vrne običajni seznam.

Napiši tudi rekurzivno funkcijo `v_seznam_rek(pari)`, ki prejme seznam v obliki parov in vrne običajni seznam.

Nasvet: rekurzivna funkcija mora vrniti seznam, ki vsebuje prvi element in vse ostale, pretvorjene v seznam.

## 4. Pari iz seznama

Zdaj pa še obratna vaja: napiši funkcijo `v_pare(s)`, ki spremeni seznam v pare.

Nasvet: funkcijo smeš napisati rekurzivno ali ne. Rekurzivna ti bo najbrž lažja, ker te bo pri nerekurzivni nekaj zelo motilo in se morda ne boš spomnil rešitve.

## 5. Sledilnik

Napiši razred `Sledilnik`, katerega konstruktor prejme slovar, katerega ključi so imena oseb, pripadajoče vrednosti pa povejo, kje posamezna oseba živi. Te podatke si shrani v poljubni obliki, ki sem vam zdi primerna.

Razred naj ima naslednje metode:

- `kje_zivi(self, oseba)` vrne kraj, v katerem trenutno živi podana oseba;
- `prebivalci(self, kraj)` vrne množico prebivalcev podanega kraja;
- `preseli(self, oseba, kraj)` preseli podano osebo v podani kraj;
- `preseli_vse(self, odkod, kam)` preseli vse prebivalce kraja odkod v kraj kam;
- `selitev(self)` vrne število vseh selitev. Če so se iz nekega kraja preselijo tri osebe, so to tri selitve.