

1. Enkratni

Napiši funkcijo `enkratni(s)`, ki dobi neprazen seznam seznamov števil. Števila znotraj posamičnega seznama se ne ponavljajo. Funkcija vrne seznam, ki vsebuje največ števil, ki jih ne vsebuje noben drug seznam. Če je takšnih seznamov več, vrne prvega med njimi.

Klic `enkratni([[1, 2, 3], [4, 1, 5], [2, 8], [3, 1, 2, 8]])` vrne `[4, 1, 5]`, saj ta seznam vsebuje dve števili (4 in 5), ki jih ne vsebuje noben drug seznam.

2. Vsota

Napiši funkcijo `vsota(s)`, ki prejme seznam, katerega elementi so cela števila in seznamami števil in seznamov. Primer takšnega argumenta je `[[1, 2], [[3, 4], 5, 113, [12]], [[[4]]], [-2, 1]]`. Funkcija vrne vsoto vseh števil; v tem primeru 143.

Če želimo preveriti, ali je `x int` (ali `list`), uporabimo `if isinstance(x, int)` (ali `isinstance(x, list)`).

3. Imena

Napiši funkcijo `imena(zelena, prepovedana)`, ki prejme seznam zelenih imen in množico prepovedanih imen.

Če nobeno od zelenih imen ni prepovedano, funkcija vrne kar podani seznam `zelena`. Sicer pa vrne seznam imen, v katerem je vsakemu imenu dodan presledek in številka v oklepaju. Vsa imena imajo pripeto isto številko. Številka mora biti najnižje pozitivno celo število, pri katerem nobeno od tako dopoljenih imen ni prepovedano.

Če pokličemo `imena(["ana", "berta", "cilka"], {"berta", "ana (4)", "berta (2)", "cilka (2)", "ana (1)", "dani (7)"})`, funkcija vrne seznam `["ana (3)", "berta (3)", "cilka (3)"]`. Številke ne more uporabiti, ker je prepovedano `ana (1)`, številke 2 pa, ker sta prepovedani `berta (2)` in `cilka (2)`.

4. Intervali

Napiši funkcijo `intervali(s)`, ki prejme neprazen niz z opisom nekaterih intervalov, na primer `"1-3, 5-9, 10-19, 21-21"`. Intervali so ločeni z vejico in presledkom ter sestavljeni iz dveh celih števil, ločenih z minusom, pri čemer je gornja meja večja ali enaka spodnji.

Funkcija mora vrniti niz, v katerem vsakemu intervalu sledita presledek in oklepaj, v katerem je zapisana širina intervala. Na koncu doda še `" => "` in nato vsoto dolžin vseh intervalov.

Klic `intervali("1-3, 5-9, 10-19, 21-21")` vrne `"1-3 (2), 5-9 (4), 10-19 (9), 21-21 (0) => 15"`.

5. Unikado

Unikado je podoben pikadu. Tekmovalci eden za drugim mečejo puščice v tarčo. Ko vsak vrže po eno puščico, mečejo naslednjo. Število krogov je nepomembno.

Tarča je razdeljena na polja. Vsako polje na tarči prinaša določeno število točk; na tarči ni dveh polj z enakim številom točk. Tekmovalec ima toliko točk, kolikor je **vsota polj, v katera so zapičene njegove puščice**. Posebnost unikada pa je: **če tekmovalec zadane polje, v katerem je že od prej kakšna druga puščica, se ta, prejšnja puščica odstrani**.

Napiši razred `Unikado` z naslednjimi metodami:

- Konstruktor prejme število tekmovalcev
- Metodo `met(self, tocke)` pokličemo ob vsakem metu; argument pove število točk v polju, ki ga je zadel tekmovalec. Če je tekmovalec zgrešil tarčo (se zgodi), je `tocke` enak `None`.
- Metoda `stanje(self)` vrne seznam, ki vsebuje toliko elementov, kolikor je tekmovalcev, pri čemer elementi predstavljajo število točk, ki jih ima tekmovalec.
- Metoda `zmagovalec(self)` vrne zaporedno številko tekmovalca, ki ima največ točk. Tekmovalce štejemo od 1 naprej. Če si eden ali več tekmovalcev deli isto število točk, vrne `None`.