

Vse rešitve shranite v eno samo datoteko s končnico `.py` in jo 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.

Dovoljena je uporaba vseh materialov na Učilnici in druge literature na poljubnih medijih. Prepovedana je vsaka komunikacija.

1. Ne na lihih

Napiši funkcijo `ne_na_lihih(s)`, ki prejme seznam, in vrne množico vseh elementov seznama, ki se nikoli ne pojavijo na lihih mestih (indeksih). Klic `ne_na_lihih([12, 17, 17, 5, 3])` vrne `{12, 3}`. 5 je izpuščena, ker se pojavi samo na lihem mestu, 17 pa se sicer na sodem, vendar tudi na lihem.

Namig: malo razmisli, preden se zaprogramiraš. Naloga je v resnici trivialna.

2. Intervali

Napiši funkcijo `intervali(s)`, ki prejme seznam števil in vrne seznam parov, ki predstavljajo začetke in konce intervalov naraščajočih zaporednih števil v tem seznamu. Tako mora klic `intervali([4, 5, 6, 7, 15, 21, 22, 23])` vrniti seznam `[(4, 7), (15, 15), (21, 23)]`.

Napiši še funkcijo `razpisi(ints)`, ki dela ravno obratno – prejme, recimo `[(4, 7), (15, 15), (21, 23)]` in vrne `[4, 5, 6, 7, 15, 21, 22, 23]`.

3. Indeksi

Napiši **rekurzivno** funkcijo `indeksi_rec(s, e)`, ki prejme seznam in vrednost ter vrne seznam indeksov, na katerih se pojavi ta vrednost. `indeksi_rec([6, 2, 4, 6, 6, 3], 6)` vrne `[0, 3, 4]`.

Napiši tudi (nerekurzivno) funkcijo `indeksi_gen(s, e)`, ki izračuna enak rezultat, vendar z izpeljanim seznamom. Vsaka funkcija je torej le en sam `return`.

4. Sopomenke

Sopomenke lahko predstavimo z množico besed, kot, recimo `{"fant", "deček", "pob"}`. Takšne množice lahko zberemo v seznam, recimo `[{"fant", "deček", "pob"}, {"cesta", "pot", "kolovoz", "makadam"}, {"kis", "jesih"}]`.

Napišite funkcijo `predelaj(stavek1, sopomenke)`, ki dobi stavek in sopomenke, ter vrne stavek, v katerem so vse besede, ki imajo sopomenke, zamenjane z naključno izbrano sopomenko. Tako lahko

```
predelaj("fant in dekleta sta vzela pot pod noge",
        [{"fant", "deček", "pob"}, {"cesta", "pot", "kolovoz", "makadam"},
        {"kis", "jesih"}, {"noge", "tace"}, {"dekle", "punca", "fajla"}])
```

vrne, recimo `"pob in punca sta vzela kolovoz pod tace"`.

5. Picerija

V neki piceriji pek peče pice na zalogo. Napišite razred `PicerijaNaZalogo` z metodami

- `speci(vrsta)`, ki je namenjena temu, da jo kličemo, ko pek speče pico podane vrste (`vrsta` bo vedno `"margerita"`, `"klasika"`, `"zelenjavna"`, ali `"siri"`). Metoda mora nekako zabeležiti, da je zdaj na voljo ena pica te vrste več.
- `prodaj(vrsta)` pokličemo, ko prodamo pico podane vrste, torej imamo zdaj eno manj. Če pice te vrste ni na zalogi, funkcija ne naredi ničesar.
- `zasluzek()` vrne dosedanji zaslužek; z `margerito` zasluži en evro, s `klasiko` dva, z `zelenjavo` enega in s `siri` tri.

Poleg tega poskrbite za izpis in dolžino. Če je `p` neka picerija, naj

- `len(p)` vrne število trenutno pečenih pic,
- `print(p)` izpiše seznam vseh vrst pic, ki so trenutno pečene, urejen po abecednem redu, ločene z vejicami (glej primer v testih). Tudi če je neke pice več, naj jo izpiše le enkrat.