Izpit iz predmeta Programiranje 1, 29. januar 2015 ob 12.30. Čas pisanja: 90 minut.

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**.

Pri reševanju nalog je dovoljena vsa literatura na poljubnih medijih.

1. Čokolada

Imamo kvadratno čokolado iz `nxn` koščkov. Kako jo lomimo, pove niz, katerega prvi znak je stran, na kateri jo lomimo, preostali znaki so število odlomljenih stolpcev oz. vrstic:

- `"<3"` pomeni, da odlomimo tri stolpce z leve,
- `">12"` pomeni, da odlomimo 12 stolpcev z desne,
- `"^1"` pomeni, da odlomimo zgornjo vrstico,
- `"v5"` pomeni, da odlomimo spodnjih 5 vrstic.

Če poskušamo odlomiti več, kot je možno (imamo pet vrstic in uporabimo `"v7"`), pač odlomimo, kolikor gre (samo pet vrstic)

Napiši funkcijo `cokolada(n, odlomi)`, ki prejme velikost čokolade (`n`) in seznam lomljenj (`odlomi`, na primer `["<3", ">2", ">1", "v12", "<7"]`. Funkcija naj pove, koliko kvadratkov čokolade je še ostalo. Klic `cokolada(10, ["<3", "v5"])` vrne 35, saj je ostalo 7 stolpcev in 5 vrstic.

2. Združi - razmeči

Napiši funkcijo `zdruzi(s)`, ki prejme seznam `s` ter vrne slovar, katerega ključi so elementi `s`-a, pripadajoče vrednosti pa množice indeksov, kjer se ta element pojavlja. Klic `zdruzi([3, 1, 12, 3, 7, 12])` mora vrniti `{1: {1}, 3: {0, 3}, 7: {4}, 12: {2, 5}}`.

Napiši še funkcijo `razmeci(s)`, ki dela ravno obratno – prejme takšen slovar in vrne seznam.

3. Brez jecljanja

Napiši ***rekurzivno*** funkcijo `brez_jecljanja_rec(s)`, ki prejme seznam `s` in vrne nov seznam, ki ne vsebuje zaporednih ponovitev istega elementa. Klic `brez_jecljanja([1, 6, 3, 3, 1, 1, 1, 1, 2, 3, 5, 5, 1])` naj vrne `[1, 6, 3, 1, 2, 3, 5, 1]`.

Napiši tudi (nerekurzivno) funkcijo `brez_jecljanja_gen(s, e)`, ki počne isto, vendar je napisana tako, da vrne izpeljan seznam (vsa 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 `sopomena(stavek1, stavek2, sopomenke)`, ki pove, ali `stavek1` in `stavek2` pomenita isto. Tako mora, recimo

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

vrniti `True`. Predpostavite lahko, da v stavku ni ločil. Ne vznemirjajte se zaradi sklonov.

5. Picerija

Pek vedno speče štiri različne pice. Ko so pečene, jih da v škatle in da na vrh kupa že pečenih pic: najprej margerito, nato klasiko, potem zelenjavno in potem sire.

Prodajalec vedno proda pico, ki je na vrhu kupa. Kupec nima kaj izbirati; če je na vrhu zelenjavna, dobi zelenjavno.

Napiši razred `Picerija` z naslednjimi metodami

- `speci()` da na kup pic štiri nove pice, kot je opisano zgoraj;
- `prodaj()` vzame s kupa zgornjo pico in vrne njeno ime (`"margerita"`, `"klasika"`, `"zelenjavna"`, `"siri"`). Če trenutno ni nobene pečene pice, naj vrne `None`.
- `zasluzek()` vrne dosedanji zaslužek; z margerito zasluži en evro, s klasiko dva, z zelenjavo enega in s siri tri.

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

- `len(p)` vrne število trenutno pečenih pic,
- `p[3]` ime pice, ki jo bo dobila tretja stranka, ki bo prišla (ob predpostavki, da pek medtem ne bo spekel ničesar,
- `print(p)` izpiše trenutno zalogo pic v obliki, opisani spodaj.

```
p = Picerija()
p.speci()
p.prodaj()
p.prodaj()
p.speci()
print(p)
```

izpiše

```
margerita > klasika > margerita > klasika > zelenjavna > siri
```