

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.

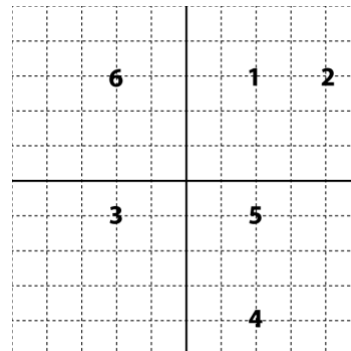
Dovoljena je uporaba vseh materialov na Učilnici in druge literature na poljubnih medijih. Prepovedan je dostop do vseh drugih spletnih strani in vsaka oblika komunikacije, razen z asistentom.

## 1. Zakladi

Neusmiljeni gusarji so se ves teden vozili po morju in na mimoidočih ladjah kradli toaletni papir. Potem so šli na nek otok, zakopali papir na različne lokacije in zdaj je potrebno sestaviti navodila za iskanje – saj poznate, dva koraka desno (kar zapišemo z ">2"), tri korake gor ("^3") in tako naprej.

Začnemo na koordinatah (0, 0). Možne smeri so <, >, ^ in v.

Napiši funkcijo zakladi(koordinate), ki prejme seznam koordinat zakladov in vrne niz s potjo, ki gre po vrsti prek njih. Vsakič najprej povemo število korakov v vodoravni, nato v navpični smeri. Kadar je število korakov v kako smer enako 0, jih izpustimo. Klic zakladi([(2, 3), (4, 3), (-2, -1), (2, -4), (2, -1), (-2, 3)]) (to so koordinate zakladov na sliki), vrne ">2 ^3 >2 <6 v4 >4 v3 ^3 <4 ^4".



## 2. Hoardanje

Pridejo časi, ko hodimo v štacuno manjkrat in takrat je dobro splanirati, kaj kupiti. Imamo recepte, pravzaprav spiske sestavin. V takšni obliki.

```
recepti = {
  "Palačinke": {"Jajca", 4}, {"Mleko", 1}, {"Moka", 0.25},
  "Kruh": {"Moka", 1}, {"Kvas", 1},
  "Pohani zrezki": {"Zrezek", 4}, {"Jajca", 1},
  "Mlečni riž": {"Riž", 0.125}, {"Mleko", 0.5},
}
```

Recimo, da se odločimo, da bomo trikrat jedli palačinke in dvakrat mlečni riž, se pravi

```
jedilnik = {"Palačinke", 3}, {"Mlečni riž", 2}
```

Napiši funkcijo spisek(jedilnik, recepti), ki prejme jedilnik in recepte v gornji obliki, vrne pa množico sestavin in količin, ki jih je potrebno kupiti. Če sta recepti in jedilnik takšna, kot smo določili zgoraj, klic funkcije vrne {"Jajca", 12}, {"Mleko", 4}, {"Moka", 0.75}, {"Riž", 0.25}. Jajc, recimo, potrebujemo 12 (palačinke delamo trikrat in vsakič porabimo štiri jajca), mleka pa potrebujemo štiri litre, namreč trikrat en liter za palačinke in še dvakrat po pol za mlečni riž. Poleg tega pa seveda še ustrezne količine moke in riža.

Rešitev mora biti seveda takšna, da bi delovala tudi pri drugih receptih, ne le teh iz testov.

## 3. Športne aktivnosti

Robin je živel na robu občine, gozd, po katerem je imel navado trenirati tek in beg, pa je bil onstran občinske meje. Da ne bi plačeval kazni, je ostal doma in se uril v lokostrelstvu. Na lesen plot je s kredo narisal daljico in s precejšnje razdalje **izmenično** streljal v eno in drugo krajišče daljice. Za evidenco je v datoteko zabeležil koordinate strelcev (primer na desni).

Ponoči je dež spral s kredo narisano daljico. Nas pa vseeno zanima: kako dolga je bila daljica?

Napiši funkcijo strelj(ime\_datoteke), ki prejme ime datoteke s koordinatami strelcev in vrne ocenjene dolžino daljice. Predpostavimo lahko, da je bila x (oz. y) koordinata posamičnega oglišča približno enaka poprečju x-koordinat strelcev v to oglišče. Tako je koordinata x prvega oglišča v gornjem primeru enaka poprečju števil 10.12, 10.18, 9.55 in 9.18, koordinata x drugega oglišča pa poprečju števil 3.14, 2.99, 3.085 in 3.123. Za y koordinate pa vzamemo druga števila v vrstici.

10.12	2.7
3.14	42
10.18	2.95
2.99	41.872
9.55	2.75
3.085	42.4
9.18	2.72
3.123	41.8

**OBRNI LIST!**

#### 4. PeF pa prazna

Recimo, da ima PeF tri stavbe. Prva je prazna. V drugi sta dve prazni sobi. V tretji pa so štiri sobe. V prvi so tri prazne mize, v drugi ena miza, na kateri je prazen zvezek, preostali sobi pa sta prazni.



V Pythonu bi to predstavili tako:

```
[ [ ], [ [ ], [ ] ], [ [ ], [ ], [ ], [ ] ], [ [ ], [ ], [ ], [ ] ] ]
```

Koliko praznih stvari imamo? Devet. 1 prazna stavba, 2 prazni sobi, 3 prazne mize, 1 prazen zvezek in še 2 prazni sobi.

Napiši funkcijo `praznih(s)`, ki prejme takšen trapast seznam in vrne število praznih seznamov (ki so lahko znotraj seznamov znotraj seznamov in tako naprej). Klic `praznih([[], [], []], [[[]], [], []], [[[]]], [], [ ] ])` torej vrne 9.

Razmisli tole: klic `praznih([ ])` vrne 0, ker je prazna samo ena stvar (namreč svet :). Sicer pa imamo seznam. In ta vsebuje toliko praznih reči, kolikor praznih reči vsebujejo njegovi elementi. Zdaj sem pa že vse povedal. :)

#### 5. Sledilnik

Napiši razred `Sledilnik` z metodami `zbolel(drzava)` in `ozdravel(drzava)`, s katerimi povemo, da je v tej državi zbolel oz. ozdravel en študent. Poleg tega ima metodi `zbolelih(drzava)` in `ozdravelih(drzava)`, ki vrnejo skupno število vseh zbolelih in ozdravelih v podani državi. Poleg tega pa še metodo `bolnih(drzava)`, ki vrne število vseh, ki so zboleli, a še niso ozdraveli.

Postorivši vse to, dodaj še metodo `doloci_kontinent(drzava, kontinent)`, s katero povemo, da je podana država na podanem kontinentu (razred pa si to na nek način zapomni). Potem spremeni metode `zbolelih(drzava)`, `ozdravelih(drzava)` in `bolnih(drzava)` tako, da bomo lahko namesto države podatli kontinent in dobili število zbolelih, ozdravelih ali bolnih na podanem kontinentu.

V testih piše (med drugim) to:

```
s = Sledilnik()
s.zbolel("Kitajska")
s.zbolel("Kitajska")
s.zbolel("Kitajska")
s.zbolel("Italija")
s.zbolel("Slovenija")
s.zbolel("Kitajska")
s.zbolel("Italija")
s.zbolel("Kr ena ladja")

s.doloci_kontinent("Italija", "Evropa")
s.doloci_kontinent("Slovenija", "Evropa")
s.doloci_kontinent("Kitajska", "Azija")

s.ozdravel("Kitajska")
s.ozdravel("Kitajska")
s.ozdravel("Italija")

self.assertEqual(s.zbolelih("Kitajska"), 4)
self.assertEqual(s.ozdravelih("Kitajska"), 2)
self.assertEqual(s.bolnih("Kitajska"), 2)

self.assertEqual(s.zbolelih("Paragvaj"), 0)

self.assertEqual(s.zbolelih("Evropa"), 3)
self.assertEqual(s.ozdravelih("Evropa"), 1)
self.assertEqual(s.bolnih("Evropa"), 2)

self.assertEqual(s.zbolelih("Azija"), 4)
self.assertEqual(s.ozdravelih("Azija"), 2)
self.assertEqual(s.bolnih("Azija"), 2)
```