

Rešitve shranite v eno samo datoteko s končnico .py in jo oddajte prek Učilnice. Imena funkcij naj bodo takšna, kot jih predpisuje naloga. Rešitev preverite s testnimi primeri. Za rešitev lahko dobite določeno število točk, tudi če ne prestane testov. Rešitev, ki prestane teste, še ni nujno pravilna in tudi posebej okorno napisane rešitve bomo nekoliko kaznovali.

Dovoljena je literatura na poljubnih medijih in ves material, ki je objavljen na Učilnici. Prepovedana je samo komunikacija.

1. Žabji skoki

Napiši funkcijo `zaba(skoki)`, ki kot argument prejme seznam parov smeri skokov in njihovih dolžin, kot rezultat pa vrne množico točk, ki jih je obiskala žaba, če je v začetku stala na koordinatah (0, 0).

```
>>> zaba(["S", 5], ["Z", 2], ["V", 6], ["S", 3])
{(0, 0), (0, 5), (2, 5), (-4, 5), (-4, 8)}
```

Nato napiši funkcijo `zaba_znotraj(skoki, max_x, max_y)`, ki dobi seznam skokov in vrne `True`, če je bila žaba ves čas znotraj pravokotnika med izhodiščem (0, 0) in (max_x, max_y). Sicer naj vrne `False`.

Napiši tudi funkcijo `naj_razdalja(skoki)`, ki vrne razdaljo do najbolj oddaljene točke (torej, najbolj oddaljene od izhodišča). V gornjem primeru bi vrnila $\sqrt{(-4)^2 + 8^2} = \sqrt{80} = 8.944$, saj je najbolj oddaljena točka (-4, 8).

Končno, napiši funkcijo `muhe(koordinate, skoki)`, ki dobi množico s koordinatami muh (npr. {(1, 0), (0, 0), (5, 0), (10, 10)}) in seznam skokov. Žaba ima zelo kratek jezik in poje muho takrat (in natanko takrat) ko skoči nanjo. Funkcija naj vrne število pojedenih muh. Na vsako koordinati je največ ena muha.

2. Naprej nazaj

V tej nalogi imamo žabo, ki vedno skače en korak daleč. Njeno pot opišemo z nizom, kot je "SSVJVZSZJJ".

Če pozorno pogledaš ta niz, vidiš, da je šla žaba najprej po poti SSVJV, nato pa po isti poti nazaj: ZSZJJ. Napiši **rekurzivno** funkcijo `naprej_nazaj(s)`, ki vrne `True`, če niz opisuje pot, ki gre nekam in po tem po isti poti nazaj in `False`, če pot nima te lastnosti.

V pomoč: pot je takšna, če je dolga nič **ali** pa je zadnji element ravno obrat prvega (recimo prvi S in zadnji J ali prvi J in zadnji Z ali prvi V in zadnji Z ali prvi Z in zadnji V) **in** je takšen tudi niz med drugo in predzadnjo črko.

Lahko si pomagaš tudi s slovarjem `obratno = {"S": "J", "J": "S", "V": "Z", "Z": "V"}`. Ni pa treba.

3. Restavracija

Napiši funkcijo `zberi(narocila)`, ki dobi slovar naročil v restavraciji: ključi so imena strank, vrednosti so jedi, ki so jih stranke naročile. Funkcija naj prešteje, koliko katere jedi potrebujemo in to vrne v slovarju, katerega ključi so jedi in vrednosti količina teh jedi.

```
>>> zberi({"Rega": {"mušji kraki", "pajek"}, "Kvakica": {"mušji kraki"}, "Dolgokrak": {"komarjev biftek"}}
{"mušji kraki": 2, "pajek": 1, "komarjev biftek": 1}
```

4. Žaba v razredu

Definirajte razred `Zaba` z metodami `sever(d)`, `jug(d)`, `vzhod(d)` in `zahod(d)`, ki premaknejo žabo `d` korakov severno, južno, vzhodno oz. zahodno. Začetne koordinate žabe naj bodo (0, 0).

Poleg tega naj ima žaba metodo `pozicija()`, ki vrne trenutno pozicijo žabe (npr. (5, 2)) in metodo `skupna_pot()`, ki vrne skupno prehojeno pot (to je, vsota vseh `d`-jev, ki so jih dobile gornje metode).

```
>>> z = Zaba()
>>> z.sever(5)
>>> z.vzhod(2)
>>> z.pozicija()
(2, 5)
>>> z.pot()
7
```