

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 s profesorjem oz. asistentom.

## 1. Preživeli

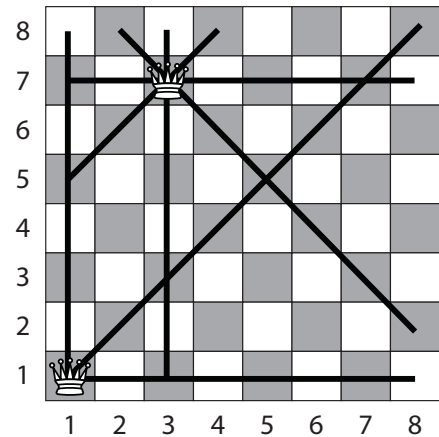
Na šahovnico polagamo figure, recimo takole: [("kmet1", "a2"), ("kmet2", "b2"), ("kmet3", "c2"), ("lovec1", "c1"), ("top1", "a1"), ("konj2", "b2"), ("kraljica", "d1"), ("lovec2", "b2"), ("top2", "c2")]. Pri tem so a2, b2, ... oznake polj. Vsaki figuri, smo, očitno, dali drugo ime, tako da jih ločimo. Opazimo pa še nekaj: na polje b2 smo dali najprej kmet2, nato konj2, nato lovec2. Na koncu se tam nahaja lovec2; figuri kmet2 in konj2 sproti odstranimo.

Napiši funkcijo `prezivelel(postavitev)`, ki prejme seznam, kot je gornji, in vrne množico imen preživelih figur. V gornjem primeru mora vrniti {"kmet1", "lovec1", "top1", "kraljica", "lovec2", "top2"}.

## 2. Prosta polja

Kraljice ogrožajo vsa polja v isti vrstici, stolpcu ali na diagonalah. Napiši funkcijo `prosta_polja(kraljice)`, ki prejme koordinate kraljice (teh je lahko poljubno veliko) in vrne število neogroženih polj. Koordinate bomo poslej opisovali s terkami, da bo lažje programirati.

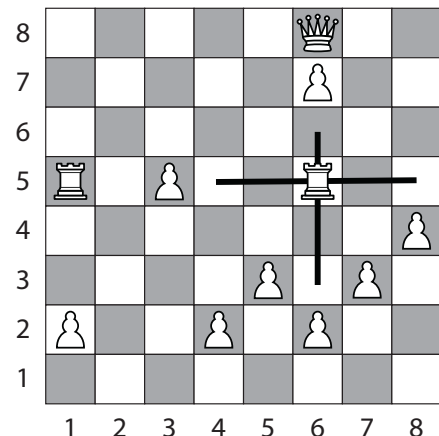
Za sliko na desni klic `prosta_polja([(1, 1), (3, 7)])` vrne 24 – kolikor je neprečrtanih polj.



## 3. Dostopnih polj

Top (AKA trdnjava) se lahko pomika le vodoravno in navpično. Trdnjavi na sliki je v tem trenutku dostopnih 8 polj (vključno s tem, na katerem stoji).

Napiši funkcijo `dostopnih_polj(koordinate, zasedena)`, ki prejme koordinate trdnjave in vseh figur ter vrne število dostopnih polj. Koordinati sta podani kot terka. Za sliko na desni klic `dostopnih_polj((6, 5), [(1, 2), (3, 5), (4, 2), (5, 3), (6, 2), (7, 3), (8, 4), (6, 7), (1, 5), (6, 8)])` torej vrne 8.



## 4. Zaščiteni kmetje

Kmet ščiti kmeta, ki se nahajata eno vrsto višje na levi in na desni. Tako kmet, ki stoji na (4, 2) (glej zadnjo sliko) ščiti kmeta na (5, 3). Kmet na (5, 3) ščiti kmeta na (4, 4) in (6, 4). Kmet (4, 4) ščiti kmeta (3, 5). Tako lahko rečemo tudi, da kmet (4, 2) posredno ščiti kmeta (3, 5).

Napiši funkcijo `sciti_kmeta(x1, y1, x2, y2, kmetje)`, ki prejme koordinate prvega in drugega kmeta ter položaje vseh kmetov (vključno s tema dvema). Vrne `True`, če prvi kmet (posredno ali neposredno) ščiti drugega, in `False`, če ne.

**Pomoč:** Kmet A ščiti kmeta B, če A neposredno ščiti B, ali pa A ščiti nekoga, ki posredno ali neposredno ščiti B.

## 5. Popotni top

Napiši razred `Top`.

- Konstruktor prejme koordinate topa (kot številci).
- Metoda `premik(smer, polj)` prejme smer, ki je lahko "`^`", "`v`", "`<`", "`>`", in razdaljo. Metoda premakne topa.
- Metoda `koordinate()` vrne trenutne koordinate topa (kot terko).
- Metoda `razdalja()` vrne skupno razdaljo, ki jo je prehodil top.

Poleg tega iz razreda `Top` izpelji razred `StarTop`. Značilnost starega topa je, da se ne more dvakrat zapored premakniti za več kot tri polja. Če enemu premiku za štiri polja (ali več) sledi nov tak ukaz, ga ignorira. Naslednji ukaz po tem spet upošteva, saj se je vmes spočil.

