

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.

Nadmorske višine točk na pravokotni mreži lahko shranimo v terko terk. Tako lahko višine na desni predstavimo s

```
teren = ((6, 17, 18, 19, 21, 21),
         (8, 12, 3, 19, 23, 22),
         (14, 14, 13, 19, 20, 21),
         (15, 16, 12, 19, 23, 23),
         (9, 5, 11, 11, 25, 24),
         (8, 6, 8, 22, 22, 22),
         (8, 6, 8, 22, 22, 22))
```

Višina točke s koordinatama x, y je očitno zapisana v `teren[y][x]`.

	0	1	2	3	4	5
0	6	← 17	← 18	← 19	← 21	21
	↑		↓		↑	↑
1	8	← 12	→ 3	← 19	← 23	→ 22
	↑	↑	↑		↓	↓
2	14	14	→ 13	← 19	← 20	← 21
	↑	↑	↓		↑	↑
3	15	← 16	→ 12	← 19	← 23	23
	↓	↓	↓	↓	↑	↑
4	9	→ 5	← 11	11	← 25	→ 24
	↓	↑	↓	↑	↓	↓
5	8	→ 6	← 8	← 22	22	22
6	8	→ 6	← 8	← 22	22	22

1. Vrhovi

Napiši funkcijo `koordinata_sosedov(x, y, teren)`, ki prejme koordinati neke točke in teren (potrebuje ga le, da razbere širino in dolžino mreže). Funkcija vrne množico koordinat vseh sosednjih štirih točk (ali manj, če je točka na robu ali v kotu). Klic `koordinata_sosedov(4, 2, teren)` vrne `{(3, 2), (5, 2), (4, 1), (4, 3)}` in klic `koordinata_sosedov(5, 2, teren)` vrne `{(5, 1), (5, 3), (4, 2)}` (ker je koordinata x že na robu). Ta funkcija bo prišla prav pri dveh drugih funkcijah.

Napiši funkcijo `vrhovi(teren)`, ki vrne množico koordinat vrhov. Točka je vrh, če ne leži na robu in je višja od vseh sosedov. Ni sliki jih prepoznaš po tem, da iz njih kažejo štiri puščice. V gornjem primeru so to `{(1, 3), (4, 1), (4, 4)}`.

2. Najnižja

Lenuh gre vedno le navzdol. Napiši funkcijo `najnizja(x, y, teren)`, ki vrne višino najnižje točke, ki jo lahko doseže lenuh, če začne pot na koordinatah x, y . Če začne na 3, 3, gre lahko levo ali dol. Če gre dol, bo dosegel le 11, če levo, pa lahko nadaljuje dol in levo (ali pa levo, dol, dol, levo, gor) ter doseže 5. Funkcija torej vrne 5. (Pazi, lenuh ne gre v najnižjo možno točko v danem trenutku. V gornjem primeru gre iz 19 na 12, ne 11.) Namig: rekurzija.

3. Višinski metri

Kolesar pa šteje "višince". Če začne na 1, 3 in gre po poti "`>>>^<<<^v`", bo vozil prek točk z višinami 16, 12, 19, 23, 20, 19, 13, 3, 13. Vsota vzponov je $(19-12) + (23-19) + (13-3) = 7 + 4 + 10 = 21$. Napiši funkcijo `visinski_metri(pot, x, y, teren)`, ki vrne skupno višino vzponov. Predpostaviti smeš, da pot nikoli ne pelje izven meja mreže.

4. Žretje

Koordinate polj na šahovnici so označene s črko in številko. "c6" pomeni tretji stolpec in šesto vrstico. V neki igri, ki se igra na šahovnici, vendar nima zveze s šahom, igralca premikata figure. Kadar nekdo premakne figuro na polje, kjer že stoji druga figura, slednjo odstrani s šahovnice.

Napiši funkcijo `zretje(figure, poteze)`, ki prejme terko s polji, na katerih stojijo figure, in poteze v obliki seznama terk (odkod, kam). Funkcija vrne število figur, ki po teh potezah ostanejo na šahovnici. `zretje(("a2", "b1"), [{"a2", "a3"}, {"b1", "a3"}])`, vrne 1: prvi igralec premakne figuro iz a2 na a3, drugi pa z b1 na a3, s čimer odstrani prvo figuro in na koncu je na igralni plošči le še ena figura.

5. Figure

V testih je podan razred `Figura`, ki predstavlja figuro na šahovnici. Tega ne spreminjaj, pač pa iz njega izpelji razreda `Trdnjava` in `Lovec`. Definiraj jima nov konstruktor in metodo `premik` z enakimi argumenti kot podedovani metodi.

`Trdnjava` (`top`) se premika vodoravno ali navpično. Metoda `premik` prejme smer, ki je lahko "`|`" ali "`-`" in razdaljo. Metoda spremeni x ali y za podano razdaljo; pozitivna razdalja poveča x oz. y , negativna ga zmanjša.

`Lovec` (`tekač`, `laufar`) se premika diagonalno. Metoda `premik` prejme smer `\` ali `/` in razdaljo. Pozitivna razdalja vedno poveča y in negativna ga zmanjša. Sprememba x je odvisna od smeri: pri smeri `/` pozitivna razdalja poveča x , negativna ga zmanjša. Pri smeri `\` pozitivna razdalja zmanjša x , negativna ga poveča.

Poleg tega, **ne da bi spremenil(a) ali sestavila novo metodo** opis, poskrbi, da bo metoda `opis` za `trdnjava` na, recimo, koordinatah 3, 5 vrnila "`Trdnjava na c5`" (in ne "`Figura na c5`"), za `lovca` na 3, 5 pa "`Lovec na c5`".