

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 asistentko.

## 1. Razglednice

Ana je napisala razglednice Berti, Cilki in Dani. Berta je pisala Ani in Cilki. Cilka je pisala le Dani. Dani je pisala Ani in Emi. Ema in Fanči nista pisali nikomur. Skratka: {"Ana": {"Berta", "Cilka", "Dani"}, "Berta": {"Ana", "Cilka"}, "Cilka": {"Dani"}, "Dani": {"Ana", "Ema"}, "Ema": set(), "Fanči": set()}.

Napiši funkcijo `ignoranti(razglednice)`, ki prejme takšen slovar in vrne slovar, ki pove, kdo komu ni odpisal, čeprav je od njega dobil razglednico. V gornjem primeru vrne {"Cilka": {"Ana", "Berta"}, "Dani": {"Cilka"}, "Ema": {"Dani"}}: Cilka ni odgovorila Ani in Berti, Dani ni odgovorila Cilki in Ema ne Dani.

## 2. Manjkajoče razglednice

Da bi sebi (in vam) zapletle življenje, so se punce dogovorile takole: vsaka mora napisati tri razglednice. Poleg tega pa lahko vsaki piše (največ) ena. Doslej so si pisale takole: {"Ana": {"Berta", "Cilka", "Dani"}, "Berta": {"Ema", "Fanči"}, "Cilka": {"Greta"}}.

Napiši (rekurzivno) funkcijo `manjkajoce_razglednice(razglednice, prvi)`, ki prejme takšen slovar in osebo, pri kateri začnemo opazovanje. Vrniti mora število razglednic, ki jih bo treba še napisati. Če pokličemo `manjkajoce_razglednice(razglednice, "Berta")`, kjer razglednice vsebuje gornji slovar, funkcija vrne 7. Berta je napisala dve razglednici, morala pa bi tri, torej manjka 1. Berta je pisala Emi in Fanči; tidve sta napisali 0 razglednic, morala pa bi 3, torej jih tu manjka še 6. Skupno jih torej manjka 7. Ane, Cilke in Dani tu nismo opazovali, ker nas pot od Berte ne pripelje do njih.

## 3. Sudoku

Sudoku je dolgočasna igra, pri kateri je potrebno v polje dimenzij 9x9 vpisovati števila od 1 do 9, tako da se v nobeni vrstici, nobenem stolpcu in nobenem od kvadratov, označenih na sliki, ne ponovi nobeno število.

Razpored na desni bomo predstavili s takšnim seznamom nizov:

["51...2..6", ".9....7..", ".4...5.8.", ".....9..", "....3....", "..8.....4", ".....4..", "..3.....", ".....9"].

Napiši naslednje funkcije.

- `mozni(plosca, vrstica, stolpec)`, ki vrne množico vseh števil, ki jih je možno postaviti na podano ploščo v podano vrstico in stolpec. Klic `mozni(plosca, 3, 9)` vrne {"1", "2", "3"}, se je na zadnje polje v tretji vrstici možno postaviti 1, 2 in 3; 4 in 5 sta prepovedani, ker sta 4 in 5 že v tej vrstici; 6, 7 in 8 so že v tem kvadratu, 9 (in tudi 6) pa sta že v tem stolpcu.

Pazi: klic `mozni(plosca, 1, 1)` vrne {"3", "5", "7", "8"} - torej tudi 5, ki je sicer res že v tem stolpcu, vrstici in kvadratu, vendar jo je očitno možno dati na to polje.

- `vsi_enkratni(plosca)` vrne množico koordinat polj, na katera je - glede na gornje omejitve - možno postaviti le eno številko. Za gornjo ploščo vrne {(1, 3), (1, 7)} saj je na vsakem od teh dveh polj možna le ena številka (slučajno sta to ravno 7 in 3). Enako množico bi vrnila tudi, če bi bila na polju (1, 3) že vpisana številka 7.
- `pravilna(plosca)` vrne True, če na plošči ni ničesar prepovedanega - dveh enakih števil v isti vrstici, stolpcu ali kvadratu.

5	1			2			6
	9				7		
	4			5		8	
					9		
			3				
		8					4
					4		
		3					
							9

## 4. Razredu Sudoku

Napiši razred Sudoku z metodo `postavi(vrstica, stolpec, stevilo)`. Ta preveri, ali je možno postaviti podano število (ki je niz!) v podano vrstico in stolpec. Če je, ga postavi in vrne True. Če ni, vrne False.

V metodi lahko kličeš funkcijo `mozni(plosca, vrstica, stolpec)`, ki si jo napisal v eni prejšnjih nalog. Še več, pokličeš jo lahko tudi, če je nisi napisal, saj jo bodo testi v tem primeru nagoljufali sami.