

# resitev

January 28, 2024

## 0.1 Izpeljani prenašalci

Tokratna domača naloga je [enaka kot pred dvema tednoma](#), vendar morate funkcije `osebe`, `dejavnosti`, `dejavnosti_osebe`, `udelezenci`, `v_stiku`, `v_karanteno` z uporabo tega, kar smo se učili danes. Torej v eni vrstici - le sestavite in vrnete množico, ki jo zahteva naloga (oziroma pri `v_stiku`, sestavite množico in preverite, ali je takšna in takšna).

Tudi preostali dve funkciji je možno brez težav napisati v eni vrstici, vendar je za to potrebno poznati stvari, ki se jih nismo in ne bomo učili.

Vaše funkcije lahko kličejo druge funkcije, ki ste jih napisali v okviru te naloge. Pač pa tokrat izjemoma ne smete pisati dodatnih funkcij (ker ni potrebno in ker bi se s tem lahko kvečjemu izognili pravi nalogi).

Seveda si lahko pomagata s svojo ali tudi z [mojo rešitvijo](#) naloge izpred dveh tednov.

Naloga je res lahka in skrajni čas je, da vam začnem dajati težje. V bistvu morate le znati pravilno zapisati, kar smo pisali na predavanjih.

## 0.2 Rešitev

Praktično vsi študenti so nalogo rešili tako.

```
[1]: def osebe(obiski):  
    return {oseba for oseba, _ in obiski}  
  
def dejavnosti(obiski):  
    return {dejavnost for _, dejavnost in obiski}  
  
def dejavnosti_osebe(oseba, obiski):  
    return {dejavnost for oseba1, dejavnost in obiski if oseba1 == oseba}  
  
def udelezenci(dejavnost, obiski):  
    return {oseba for oseba, dejavnost1 in obiski if dejavnost1 == dejavnost}  
  
def v_stiku(oseba1, oseba2, obiski):  
    return bool(dejavnosti_osebe(oseba1, obiski) & dejavnosti_osebe(oseba2,   
↪obiski))  
  
def v_karanteno(oseba, obiski):
```

```
    return {oseba1 for oseba1 in osebe(obiski) if v_stiku(oseba1, oseba,   
↪obiski) and oseba1 != oseba}
```

In o tem se nimamo kaj pogovarjati.