

Vse rešitve shranite v eno samo datoteko s končnico `.py` in jo oddajte prek Učilnice. Vse funkcije naj imajo takšna imena, kot jih predpisuje naloga. Za rešitev naloge lahko dobite določeno število točk, **tudi če ne prestane testov**. Funkcija, ki prestane vse teste, **še ni nujno pravilna**.

Dovoljena je vsa literatura na poljubnih medijih, ves material, ki je objavljen na Učilnici, vključno s programi. Izpit morate pisati na fakultetnih računalnikih. Študenti s predolgimi vratovi in podobnimi hibami bodo morali zapustiti izpit. Hujše kršitve, na primer uporabo telefonov in podobno, bomo prijavili disciplinski komisiji.

Študentom **Kognitivne znanosti** je potrebno za 100 % rešiti štiri naloge. Naloga iz snovi, ki se vas ne tiče, je zadnja, vendar jo lahko, če želite, rešite; točke se vam bodo upoštevale.

1. Vstavi teže

Imamo seznam imen; ženska imena se vedno končajo s črko "a", moška pa nikoli. Vsi moški so nam zaupali svoje teže. Napišite funkcijo `vstavi_teze(osebe, teze)`, ki prejme tadva seznama in imena moških zamenja z njihovimi težami (po vrsti), imena žensk pa pusti pri miru. Funkcija naj ne vrača ničesar – spreminjati mora podani seznam.

Primer

```
>>> imena = ["Adam", "Eva", "Kajn", "Abel"]
>>> teze = [87, 86, 75]
>>> vstavi_teze(imena, teze)
[87, "Eva", 87, 75]
```

2. Trgovanje

Nekdo je za desko ali steklenico pripravljen dati pašteto. Nekdo drug je pripravljen za knjigo ali vilice dati svečo, desko ali papir. Nekdo tretji je za ribo ali pašteto pripravljen dati tipkovnico ali zaslon. To zapišemo kot

```
ponudbe = [{"deska", "steklenica"}, {"pašteta"}],
           [{"knjiga", "vilice"}, {"sveča", "deska", "papir"}],
           [{"riba", "pašteta"}, {"tipkovnica", "zaslon"}]
```

Če je tako, lahko zamenjamo vilice za desko, desko za pašteto in pašteto za zaslon (`["vilice", "deska", "pašteta", "zaslon"]`). Zamenjamo lahko tudi, recimo, desko za ribo. Ne moremo pa zamenjati vilic za papir, papirja za pašteto in paštete za tipkovnico (`["vilice", "papir", "pašteta", "tipkovnica"]`), ker na vsem božjem svetu ni nikogar, ki bi menjal papir za pašteto.

- Napišite funkcijo `obstaja(dam, dobim, ponudbe)`, ki pove (`True` ali `False`), ali v seznamu `ponudbe` obstaja kdo, ki bi za stvar `dam` dal stvar `dobim`.
- Nato napišite funkcijo `menjave(zaporedje_stvari, ponudbe)`, ki pove, ali je možno podano zaporedje menjav.

```
>>> obstaja("pašteta", "zaslon", ponudbe)
True
>>> menjave(["deska", "pašteta", "zaslon"], ponudbe)
True
>>> menjave(["vilice", "papir", "pašteta", "tipkovnica"], ponudbe)
False
```

3. Izmenjavanje

Napiši rekurzivno funkcijo `izmenjavanje(imena)`, ki prejme seznam imen in vrne `True`, če se v njem izmenjujejo ženska in moška imena (`["Ana", "Benjamin", "Cilka", "Daniel"]`) in `False`, če se ne (`["Ana", "Benjamin", "Cilka", "Dora", "Erik"]`). Seznam se lahko začne z moškim ali z žensko.

4. Figura

R	L	L	L
R	R	L	L
F	R	F	F
F	R	L	F

Figuro na začetku postavimo na neko polje, obrnjeno v neko smer. Če na polju piše L, se figura obrne za 90 stopinj v levo, če R za 90 stopinj v desno, če F, se ne obrne. Nato naredi korak naprej. Na naslednjem polju se zgodba ponovi.

Slika kaže gibanje figure, ki smo jo postavili na igralno ploščo na polje (3, 1) obrnjeno navzgor. Obrnila se je za 90 stopinj levo in naredila korak. Nato se je spet obrnila v levo in naredila korak. Potem se ni obrnila in je le naredila korak. Nato se je spet obrnila levo in naredila korak. Nato se ni obrnila nikamor, naredila korak in padla čez rob.

Koordinate merimo od leve proti desni od zgoraj navzdol; polje (0, 0) je zgoraj levo. Smer figure zapišemo z U, R, D, L (up, right, down, left).

- Napišite funkcijo `spremeni_smer(trenutna_smer, obrat)`, ki kot argument dobi trenutno smer in tisto, kar piše na polju, vrne pa novo smer. Če pokličemo, recimo, `spremeni_smer("R", "L")`, mora funkcija vrniti "U": če figuro, ki v začetku gleda desno, obrnemo proti levi, bo gledala gor.
- Napišite funkcijo `korak(x, y, smer)`, ki prejme koordinate figure in smer gibanja ter vrne nove koordinate. Če pokličemo `korak(5, 8, "U")`, mora vrniti 5, 7.
- Napišite funkcijo `potuj(x, y, smer, plosca)`, ki prejme začetne koordinate in smer ter igralno ploščo, vrne pa pot, ki jo prepotuje figura. Plošča je podana kot seznam nizov (glej spodnji primer). Pot se konča, ko figura pade čez rob plošče ali pa stopi na polje, na katerem je že bila. Pot, ki jo vrne, mora biti seznam terk, ki vsebuje tudi prvo in zadnje polje. Če pokličemo `potuj(3, 1, U, ["RLLL", "RLL", "FRFF", "FRLF"])`, mora vrniti [(3, 1), (2, 1), (2, 2), (2, 3), (3, 3), (4, 3)] – primer z gornje slike.

5. Knjižica

Definirajte razred `Knjizica`, v katerem bo knjižnica beležila, katere knjige ima nekdo sposojene, in računala zamudnino. Objekt `Knjizica` se nanaša na posameznega člana knjižnice.

- Konstruktor naj nima argumentov in naj le pripravi, kar mora.
- `izposodi(naslov, datum)` zabeleži, da si je lastnik te knjižice na podani datum izposodil knjigo s podanim naslovom.
- `izposojeno()` vrne *množico* z naslovi vseh knjig, ki jih ima trenutno izposojene lastnik `Knjizice`.
- `vrni(naslov)` pokličemo, ko član vrne določeno knjigo.
- `zamudnina(danasnji_datum)` izračuna zamudnino.

Z datumi in zamudnino je takole. Datum je podan kot številka – konkretno, gre za število dni, odkar je knjižnica odprta. Če je datum, recimo, 42, to pomeni 42. dan po odprtju knjižnice.

Knjige si je možno izposoditi za 21 dni. Če si nekdo izposodi knjigo na 10. dan, potem na 31. dan še ne plača zamudnine. Kasneje znaša zamudnina en evro za vsak dan za vsako knjigo. Recimo, da si nekdo izposodi dve knjigi, eno na 10. dan in eno na 15. dan. Prvo knjigo ima lahko do 31. dneva in drugo do 36. dneva. Funkcija `zamudnina(40)` (torej "danes je 40. dan, koliko zamudnine je dolžan) mora vrniti 13, to je, 9 + 4 – devet za prvo knjigo in 4 za drugo.