

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. Najpogostejše polje

Podan je razred Robot. Ta se v začetku nahaja na koordinatah (0, 0) in je obrnjen na desno. Koordinatni sistem je takšen kot pri matematiki: koordinata y narašča navzgor. Robot ima naslednje metode:

- `naprej(d)` gre za `d` naprej v podani smeri;
- `desno()` se obrne za 90 stopinj v desno;
- `levo()` se obrne za 90 stopinj levo;
- `coordinate()` vrne trenutne koordinate (x in y)

Napišite funkcijo `najpogostejse_polje(navodila)`, ki prejme navodila za robota v obliki niza, na primer "7D3221LL", kar bi pomenilo, da skoči za 7 polj naprej, se obrne desno, nato skoči za 3 polja naprej, za 2 polji naprej, za 2 polji naprej, za 1 polje naprej, nato se obrne levo in še enkrat levo. Funkcija vrne koordinati polja, na katerega je robot najpogosteje prišel. (Robot "pride" tudi na začetno polje. Ko se robot obrne, se to ne šteje, kot da je ponovno prišel na to polje. Polja, prek katerih gre, vendar se na njih ne ustavi, ne štejejo.)

## 2. Navodila

Napiši funkcijo `navodila(zaporedje)`, ki prejme zaporedje, kot je, na primer "7D3221LL" in vrne seznam [7, "D", 3221, "L", "L"]. Tole je podobno, a ravno prav različno, da si ne morete pomagati s prejšnjo nalogo. :) Razlika je v tem, da je tu potrebno več zaporednih števk šteti kot eno število.

## 3. Menjave

Napiši funkcijo `zamenjano(s, menjave)`, ki prejme seznam `s` in slovar `menjave`. Vrne naj nov seznam, v katerem so vsi elementi seznama, ki nastopajo kot ključi v slovarju, zamenjani s pripadajočimi vrednostmi. Elemente, ki se ne pojavijo v slovarju, pusti pri miru.

Klic `zamenjano(["Ana", "Ana", "Berta", "Ana", "Cilka"], {"Ana": "Peter", "Berta": "Ana"})` vrne ["Peter", "Peter", "Ana", "Peter", "Cilka"].

Funkcija `zamenjano` ne sme spremeniti podanega seznama `s`.

Poleg tega napiši podobno funkcijo `zamenjaj(s, menjave)`, ki pa ne vrne ničesar temveč ustrezno spremeni podani seznam `s`.

## 4. Sprazni

Napiši **rekurzivno** funkcijo `sprazni(s)`, ki prejme seznam, katerega elementi so `None` in seznamami, katerih elementi so `None` in seznamami in tako naprej. Funkcija naj vrne prav tak seznam, a brez `None`ov. Če pokličemo `sprazni([None, None, [None], [None, None, []], None])`, dobimo `[[], [[]]]`.

Pomoč: funkcija sestavlja seznam. Če vidi `None`, ga preskoči. Če vidi seznam, ga doda v nastajajoči seznam ... a spraznjenega.

## 5. Čarovnik

Napiši razred `Carovnik`, katerega **konstruktor** kot argument dobi niz, ki pove, kaj čarovnik zna in koliko zaračuna za to, na primer `alkemist = Carovnik("svinec -> zlato", 10)`. Niz bo vedno vseboval dve enobesedni stvari, ločeni z `->`.

Razred ima metodo `caraj(s)`, ki prejme seznam in vrne nov seznam, v katerem so vsi elementi prečarani v nove. Tako bi klic `alkemist(["svinec", "les", "zlato", "svinec"])` vrnil ["zlato", "les", "zlato", "zlato"]. Alkemist bi za to zaračunal 20 evrov. (Poceni!)

Poleg tega ima razred metodo `zasluzek`, ki pove, koliko je čarovnik doslej zaslužil. V gornjem primeru bi klic `alkemist.zasluzek()` torej vrnil 20.