Izpit iz predmeta Programiranje 1, 31. januar 2024 **Čas pisanja: ~~60 minut~~ → 90 minut**

Dovoljena je uporaba literature (Učilnica in vse, kar prinesete s seboj), prepovedana pa je vsakršna komunikacija.

Kot v prvih domačih nalogah bodo ovire predstavljene s trojkami `(x0, x1, y)`, pri čemer se ovira razprostira od stolpca `x0` do vključno `x1`, šteti pa začnemo z 1. Zemljevid na desni bi opisali z `[(4, 7, 1), (1, 3, 2), (10, 14, 2), (4, 7, 4), (10, 14, 4), (4, 7, 6), (1, 3, 7), (6, 7, 7), (11, 15, 7)]`.

```
    1
  123456789012345
1 ...<-->........
2 <->......<--->.
3 ...............
4 ...<-->..<--->.
5 ...............
6 ...<-->........
7 <->..<>...<--->
```

1. Nabava

Zemljevidi vsebujejo ovire različnih dolžin. Gornji zemljevid, recimo, ima 1 oviro dolžine 2, 2 oviri dolžine 3, 3 ovire dolžine 4 in 3 ovire dolžine 5. Nekega dne se MOL odloči spremeniti razpored ovir na neki kolesarski poti, zato bo morda potrebno nabaviti nekaj novih ovir določenih dolžin (kar bo sicer strošek, vendar MOL takrat, ko gre za varnost kolesarjev, ne varčuje!)

Napiši funkcijo `nabava(stari, novi)`, ki prejme dva seznama ovir in vrne slovar, katerega ključi so dolžine ovir, ki jih bo potrebno dokupiti, ker med obstoječimi ovirami ni (dovolj) ovir takšne dolžine. Pripadajoče vrednosti bodo število potrebnih novih ovir. Če je potrebno dokupiti 3 ovire dolžine 5 in sedem ovir dolžine 1, funkcija vrne `{5: 3, 1: 7}`.

2. Rekonstrukcija

Napiši funkcijo `rekonstrukcija(kocke)`, ki dobi seznam polj, ki jih pokrivajo ovire. Seznam je podan v obliki parov `(y, x)` (pazi: najprej y, potem x, saj to olajša nalogo!), ki pa niso nujno urejeni. Funkcija mora vrniti seznam dejanskih ovir v obliki običajnih trojk. Vrnjeni seznam mora biti urejen po vrsticah in znotraj vrstic po stolpcih.

`rekonstrukcija([(2, 3), (1, 1), (2, 2), (2, 4), (1, 2), (3, 4)])` vrne `[(1, 2, 1), (2, 4, 2), (4, 4, 3)]`.

3. Spet nov format

Napiši funkcijo `dekodiraj_vrstico(vrstica)`, ki dobi eno vrstico zemljevida v obliki, kot ga vidiš na vrhu izpita. Vrne seznam začetkov in koncev ovir: `dekodiraj_vrstico("..<-->...<---->..<>")` vrne `[(3, 6), (10, 15), (18, 19)]`. Ovir dolžine 1 ni. Je Angelca rekla, da so čisto brez zveze.

Napiši funkcijo `preberi(ime_datoteke)`, ki prejme ime datoteke, v kateri je shranjen zemljevid v takšni obliki. Vrne naj seznam ovir (urejen po vrsticah, znotraj po stolpcih). Za zemljevid z gornje slike mora vrniti seznam iz besedila ob njej.

4. Sodobna umetnost

```
      T
j     l          z       B  A
w i oo  pp       s       gg n
c r uu vvv       x y     qq mm
aaa bbbbbb       ttt ee fffff
ddddddddd        hhhhhhhhhhh
.............................
```

V sklopu konference Velocity je potekala tudi umetniška razstava z inštalacijo iz ovir. Naložili so jih, kot kaže slika na desni: na zelo široki oviri d stojita oviri a in b. Na a stojita c in r... in tako naprej. Takšno postavitev predstavimo s slovarjem `{".": "dh", "d": "ab", "h": "tef", "a": "cr", "b": "uv", "t": "xy", "f": "qm", "c": "w", "r": "i", "u": "o", "v": "p", "x": "s", "y": "", "q": "g", "m": "n", "w": "j", "o": "l", "s": "z", "g": "B", "n": "A", "l": "T"}`. Ovira je *vrh*, če na njej ni nobene druge ovire. V primeru na sliki so vrhovi j, i, T, p, z, y, e, B, A.

Napiši funkcijo `vrhovi(skladovnica, ovira, visina)`, ki prejme takšen slovar, neko oviro in neko višino. Vrniti mora množico imen vrhov, ki so nad to oviro in so za vsaj `visina` višji od nje. Funkcija naj bo seveda splošna in naj ne deluje le za ovire v tej umetnini. Seveda pa predpostavimo, da je ime ovire vedno podano z eno samo črko. Klic `vrhovi(skladovnica, "a", 3)` vrne `{"j"}`, in `vrhovi(skladovnica, "a", 2)` vrne `{"j", "i"}`.

Namig: višina je lahko tudi negativna. V tem primeru funkcija vrne vse vrhove nad oviro. (To ti bo v pomoč!)

5. Potapljanje ovir

Napiši razred `Ovire`.

- Konstruktor sprejme množico ovir, podanih s trojkami `(x0, x1, y)`.
- Metoda `strel(x, y)` prejme koordinate nekega polja. Če je tam ovira, vrne `True`, sicer `False`.
- Metoda `zadetkov()` vrne število strelov, ki so zadeli kako oviro.
- Metoda `vse_ovire()` vrne ovire, ki še obstajajo. Ovira, ki je trikrat zadeta (lahko trikrat v isto točko, lahko v različne), se razblini (v celoti, ne le tam, kjer je bila zadeta).
- Metoda `zmaga()` vrne `True`, če ni ostala nobena ovira več. Sicer pa `False`.