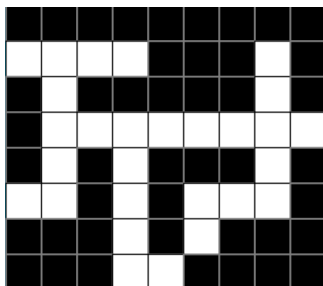


## Zadanie skúšky z programovania 6.2.2017

### Kamery v Kocúrkove

Na mape Kocúrkova môžete vidieť bielou farbou označené ulice:



Mestská rada chce do mesta umiestniť kamery, ktoré budú sledovať ulice všetkými smermi ľubovoľne ďaleko, ale len vo vodorovných a zvislých smeroch (kamera cez domy nevidí). Zrejme majú v pláne rozmiestniť kamery tak, aby mohli monitorovať všetky ulice mesta. Preto pre nich pripravíme aplikáciu, pomocou ktorej budú môcť experimentovať s rozmiestňovaním rôzneho počtu kamier.

Táto aplikácia by mala fungovať takto:

- na začiatku je mapa prázdna - nie je tu žiadna kamera, len ulice a domy;
- používateľ našej aplikácie môže na ľubovoľné miesta mapy umiestňovať nové kamery (každá kamera má svoj číselný identifikátor), prípadne už predtým umiestnenú kameru presunúť na ľubovoľné iné miesto, alebo ju z mapy odstrániť;
- používateľ si hocikedy môže nechať zobraziť mapu aj s vyznačenými monitorovanými políčkami mapy;
- tiež dostáva informáciu o tom, koľko políčok mapy je už monitorovaných a koľko ešte nie.

Textový súbor popisuje hraciu plochu ako štvorcovú sieť, v ktorej sú všetky riadky rovnako dlhé. Každý riadok súboru popisuje jeden riadok štvorcovej siete: obsahuje postupnosť celých čísel (oddelených medzerami), pričom prvé číslo v postupnosti označuje počet plných štvorčekov (domov) na začiatku riadku, za tým nasleduje počet prázdnych políčok (ktoré označujú ulice), každé ďalšie číslo v riadku označuje buď počet plných alebo prázdnych políčok. Zrejme súčet všetkých čísel v každom riadku musí byť rovnaký a rovná sa šírke hracej plochy. Počítajte s tým, že medzi týmito číslami môžu byť aj nuly.

Riešenie zapíšete do triedy **Kocurkovo** s týmito metódami:

```
class Kocurkovo:

    def __init__(self, meno_suboru):
        ...

    def __str__(self):
        return ''

    def kamera(self, cislo, pozicia):
        return 0

    def kontrola(self):
        return 0, 0

    def zoznam(self):
        return ()
```

kde

- metóda `__str__()` vráti znakový reťazec, ktorý reprezentuje kompletnú mapu: v tejto mape je políčko s domom vyjadrené znakom 'M', monitorované políčko (aspoň jednou kamerou) znakom '+', a nemonitorovanú ulicu znakom '.'; medzi riadkami mapy je znak '\n';
- metóda `kamera(cislo, pozicia)` umiestni kameru s daným číslom (číslo kamery je ľubovoľné celé číslo) na zadanú pozíciu (dvojjicu čísel pre riadok a stĺpec, pričom číslujeme od 0); ak už bola predtým táto kamera v ploche, tak sa takto premiestni na túto novú pozíciu; metóda vráti počet políčok, ktoré táto kamera monitoruje (bez ohľadu na iné kamery); ak je umiestnená na políčko mimo ulice (teda dom, ktorý je na mape označený ako 'M'), vráti hodnotu 0; ak je umiestnená mimo súradníc mapy, je takto z mapy odstránená a metóda vráti hodnotu `None`
- metóda `kontrola()` vráti dvojjicu (typ `tuple`): celkový počet monitorovaných políčok a počet políčok, ktoré žiadna kamera nemonitoruje;
- metóda `zoznam()` vráti n-ticu (typ `tuple`) informácií o kamerách; každá kamera je v n-tici reprezentovaná dvojjicu (číslo kamery, pozícia), kde pozícia je dvojjica (riadok, stĺpec).

Napr. pre súbor (popisuje mapu z úvodu):

```
9
0 4 3 1 1
1 1 5 1 1
1 8
1 1 1 1 3 1 1
0 2 1 1 1 3 1
3 1 1 1 3
3 2 4
```

vypíše:

```
MMMMMMMMMM
...MMM.M
M.MMMMM.M
M.....
M.M.MMM.M
..M.M...M
MMM.M.MMM
MMM..MMMM
kontrola = 0 28
kamera 12
MMMMMMMMMM
...MMM.M
M.MMMMM.M
M+++++++
M.M+MMM.M
..M+M...M
MMM+M.MMM
MMM+.MMMM
kontrola = 12 16
kamera 7
MMMMMMMMMM
...MMM+M
M.MMMMM+M
M+++++++
M.M+MMM+M
..M+M+++M
MMM+M.MMM
MMM+.MMMM
kontrola = 18 10
zoznam: (1, (3, 3)) (3, (5, 7))
```

takýto test:

```
if __name__ == '__main__':
    k = Kocurkovo('subor1.txt')
    print(k)
    print('kontrola =', *k.kontrola())
    print('kamera', k.kamera(1, (3, 3)))
    print(k)
    print('kontrola =', *k.kontrola())
    print('kamera', k.kamera(3, (5, 7)))
    print(k)
    print('kontrola =', *k.kontrola())
    print('zoznam:', *k.zoznam())
```

Z úlohového servera L.I.S.T. si stiahnite kostru programu `skuska.py`. Pozrite si testovacie dáta v súboroch `'subor1.txt'`, `'subor2.txt'`, `'subor3.txt'`, ..., ktoré bude používať testovač.

Aby ste mohli spúšťať skúškové testy, program uložte do súboru `skuska.py`.

Riešenie (len súbor `skuska.py` bez dátových súborov) odovzdajte na úlohový server

<http://capek.ii.fmph.uniba.sk/list>.

Praktická časť končí o 11:00 a skúška ďalej pokračuje od 12:00 vyhodnotením v kancelárii **m162**.