Domácí projekty 8

Na začátek trocha práce se seznamy. Některé z projektů a zvláštíten poslední potřebují trochu přemýšlení; nebudeš-li vědět jak dál, zeptej se na internetu nebo na sraze ostatních a řešte společně!

- o. Udělej si seznam domácích zvířat. Budeš ho potřebovat v dalších úlohách. Domácí zvířata známe tato: "pes", "kočka", "králík", "had".
- 1. Napiš funkci, která vrací jména domácích zvířat (zadaných argumentem), která jsou kratší než 5 písmen.
- 2. Napiš funkci, která vrací jména domácích zvířat (zadaných argumentem), která začínají na k.
- 3. Napiš funkci, která dostane slovo a zjistí, jestli je v seznamu domácích zvířat. "Zjistí" znamená, že funkce vrátí True nebo False.
- 4. Napiš funkci, která dostane dva seznamy jmen zvířat a vrátí tři seznamy:
 - (a) zvířata, která jsou v obou seznamech (sjednocení množin: první ∪ druhá),
 - (b) zvířata, která jsou jen v prvním seznamu (rozdíl množin: první druhá),
 - (c) zvířata, která jsou jen ve druhém seznamu (rozdíl množin: druhá první).

Napiš (a pusť) k této funkci testy, aby sis ověřila, že funguje správně.

- 5. Napiš program, který seřadí seznam domácích zvířat podle abecedy.
- 6. Had byl pyšný na to, že je v abecedě první. Dokud nepřiletěla "andulka". Abys hada uklidnila, vytvoř funkci, která zvířata seřadí podle abecedy, ale bude ignorovat první písmeno (t.j. vrátí ["had", "pes", "andulka", "kočka", "králík"]). Postup:
 - Máš seznam hodnot, které chceš seřadit podle nějakého klíče. Klíč se dá z každé hodnoty vypočítat.
 - Vytvoř seznam dvojic (klíč, hodnota).
 - Seřaď tento seznam dvojic dvojice se řadí nejdřív podle prvního prvku, pak druhého atd.
 - Nakonec vytvoř ze seznamu dvojic opět jen seznam hodnot.

Proč má zrovna had takovéhle výsadní postavení, zjistíš později.

Cvičení na analýzu reálných dat

- 7. Napiš tyto funkce. Každá z nich dostane jako argument řetězec s rodným číslem a nějak ho zanalyzuje:
 - (a) Je ve správném formátu: 6 číslice, lomítko, 4 číslice? (vrací True nebo False)
 - (b) Je dělitelné jedenácti? (vrací True nebo False)
 - (c) Jaké datum narození je v čísle zakódováno? (vrací trojici čísel den, měsíc, rok)
 - (d) Jaké pohlaví je v čísle zakódováno? (vrací 'muž' nebo 'žena')

Napiš i testy, abys ověřila, že funkce fungují správně.

Pro účely úkolu stačí, když bude program umět zpracovat čísla vydávaná od roku 1985. Reálná rodná čísla můžou být složitější :)

8. Napiš program který se uživatele zeptá na rodné číslo a vypíše výsledky.

Jedna klasická programovací úloha, která nejspíš pořádně potrápí tvé logické myšlení. Je nepovinná, nemáš-li na ni aspoň pár hodin, tak ji přeskoč.

Napiš funkci, která převede římské číslice na číslo (int).
 Tip: Nejdřív napiš k této funkci testy, aby sis ověřila, že (a co) (ne)funguje správně.

Projekty I0-16 závisí jeden na druhém, řeš je postupně. Až to uděláš, můžeš si zahrát hru! Tahle sekce není jednoduchá (a poslední dva projekty jsou obzvláště náročné). Můžeš zkusit spojit síly s ostatními účastnicemi kurzu!

10. Napiš funkci, která dostane seznam souřadnic (párů čísel menších než 10, která určují sloupec a řádek) a vypíše je jako mapu: mřížku 10×10, kde na políčka která jsou v seznamu napíše X, jinde tečku. Například:

Jak na to?

. X .

- (a) Udělej tabulku (seznam seznamů) se samými tečkami, něco jako: [['.', '.', '.'], ['.', '.'], ['.', '.']].
- (b) Na příslušných místech nahraď tečky X-ky.
- (c) Tabulku vypiš pomocí dvou cyklů for zanořených do sebe.
- 11. Napiš funkci pohyb, která dostane seznam souřadnic a světovou stranu ("s", "j", "v" nebo "z") a přidá k seznamu poslední bod "posunutý" v daném směru. Např.:

```
souradnice = [(0, 0)]
pohyb(souradnice, 'v')
print(souradnice)  # → [(0, 0), (1, 0)]
pohyb(souradnice, 'v')
print(souradnice)  # → [(0, 0), (1, 0), (2, 0)]
pohyb(souradnice, 'j')
print(souradnice)  # → [(0, 0), (1, 0), (2, 0), (2, 1)]
pohyb(souradnice, 's')
print(souradnice)  # → [(0, 0), (1, 0), (2, 0), (2, 1), (2, 0)]
```

Funkce by neměla nic vracet. Jen mění už existující seznam.

Nezapomeň na testy.

12. Napiš cyklus, který se bude ptát uživatele na světovou stranu, podle ní zavolá pohyb, a následně vykreslí seznam jako mapu. Pak se opět se zeptá na stranu atd.

Začínej se seznamem [(0, 0), (1, 0), (2, 0)].

- 13. Doplň funkci pohyb tak, aby při pohybu umazala první bod ze seznamu souřadnic. Výsledný seznam tak bude mít stejnou délku, jako před voláním. Uprav testy.
- 14. Doplň funkci pohyb tak, aby zamezila:
 - pohybu ven z mapy,
 - pohybu na políčko, které už v seznamu je.

Vhodná výjimka pro tyto situace je ValueError('Game over'). Doplň i testy.

15. Přidej do hry hadí potravu. Tady jsou pravidla pro vegetariánského hada, ale můžeš si je změnit podle chuti:

Seznam ovoce obsahuje na začátku jedno ovoce na políčku, na kterém není had (například: [(2, 3)] znamená jedno ovoce na pozici (2, 3)). Když had sežere ovoce, vyroste ("nesmaže" se mu ocas, tedy neprovede se to, cos přidala v projektu 13), a pokud na mapě zrovna není další ovoce, na náhodném místě (kde není had) vyroste ovoce nové.

Každých 30 tahů vyroste nové ovoce samo od sebe.

Na mapě se toto tajemné ovoce zobrazuje jako otazník (?).

16. Hadí hřiště může mít libovolné rozměry větší než 4×1. Třeba 20×20 nebo 10×30.

A nakonec projekt na přemýšlení.