

# C2184 Úvod do programování v Pythonu

## Závěrečný test – 19. 12. 2019

---

### Pokyny:

- Během testu je povoleno používat libovolné materiály vč. internetu a vlastního notebooku. Zakázána je pouze komunikace (online i fyzická) s jiným člověkem (kromě učitele).
- Na řešení testu jsou vyhrazeny maximálně 2 hodiny.
- Řešení je nutné vyplnit do připravených šablon s doctesty. Při řešení si můžete vytvářet pomocné funkce.
- Řešení (archiv s vyplněnými šablonami) odevzdejte včas do připravené odevzdávací skříně.
- Dejte si pozor na rozdíl mezi “funkce vrátí” (`return`), “funkce vypíše” (`print`), “funkce vyhodí výjimku” (`raise`). Taky pozor na rozdíl mezi “funkce vezme parametr”, “načte ze vstupu” (`input`), “načte ze souboru”.
- Testy si můžete spustit z příkazové řádky pomocí:  

```
python3 -m doctest moje_reseni.py -v
```

---

### Zadání:

**1. (16 bodů)** *Narcistické číslo* je číslo, které je rovné součtu svých cifer umocněných na počet cifer. Např. číslo 8208 je narcistické, protože  $8^4 + 2^4 + 0^4 + 8^4 = 8208$ .

Napište definici funkce `je_narcisticke`, která vezme přirozené číslo `n` a vrátí `True`, právě když `n` je narcistické číslo.

---

**2. (16 bodů)** Napište definici funkce `celkove_skore`, která vezme jako parametr řetězec s informacemi o famfrpálovém turnaji. Tento řetězec bude obsahovat záznamy oddělené bílými znaky. Každý záznam odpovídá jednomu zápasu (názvy soupeřících týmů a jejich skóre), viz PŘÍKLAD v šabloně. Funkce vrátí slovník, ve kterém klíče jsou názvy týmů a hodnoty jsou jejich celkové skóre.

(Na pořadí klíčů ve slovníku nezáleží, v testech je použita funkce `pprint`, která klíče seřadí podle abecedy.)

**3. (16 bodů)** Napište definici funkce `prumerna_delka_slova`, která vezme jako parametr řetězec obsahující text a vrátí průměrnou délku slova v tomto textu.

Text bude obsahovat pouze písmena, mezery, čárky a tečky. Pokud bude mezi dvěma slovy čárka nebo tečka, vždy za ní bude mezera. Tyto předpoklady nemusíte v kódu kontrolovat.

Výsledek není nutné zaokrouhlovat, postará se o to funkce `round` v doctestu. Pokud průměrnou délku slova z nějakého důvodu nelze spočítat, funkce má vyhodit výjimku typu `ValueError`.

---

**4. (16 bodů)** Napište program `z4_autocomplete.py`, který načte ze souboru `phrases.txt` seznam často hledaných frází a jejich četnost vyhledávání (každý řádek je ve formátu *četnost:fráze*). Poté bude program načítat ze standardního vstupu začátky frází (jeden začátek na každém řádku) a ke každému začátku vypíše všechny vyhovující fráze seřazené podle klesající četnosti. Program skončí, když bude ze vstupu zadán prázdný řádek.

Ukázka spuštění programu (řádky zobrazené tučně jsou vstup, normálním písmem výstup):

**how to**

- how to learn python
- how to make plutonium

**blablabla**

**what**

- what is the meaning of life
- what is justin bieber's favourite colour

**wh**

- what is the meaning of life
  - who is justin bieber
  - what is justin bieber's favourite colour
- 

**5. (16 bodů)** Napište program `z5_mesta.py`, který ze souboru `cities.csv` načte souřadnice měst. Program najde nejsevernější a nejjižnější město a vypíše jejich názvy a vzdálenost (vzdušnou čarou).

Pro zjednodušení předpokládejme, že města jsou umístěna v rovině (Země je plochá). Souřadnice  $x$  odpovídá směru západ-východ a souřadnice  $y$  směru jih-sever. Vzdálenost mezi body  $A$  a  $B$  v rovině lze spočítat jako  $d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$

---

**Bonus (10 bodů)** V souboru `cities.csv` se skrývá chyba – zlý skřítek prohodil souřadnice dvou měst. Pokuste se odhalit, která dvě města to jsou. (Odevzdejte kód, který vám pomohl odhalit chybu, a do komentáře uveďte, která města jsou prohozena.)

Nápověda:

```
from matplotlib import pyplot as plt
plt.scatter([1, 2, 3], [5, 8, 4])
plt.annotate('Alice', (1, 5))
plt.annotate('Bob', (2, 8))
plt.annotate('Cyril', (3, 4))
plt.show()
```