

# C2184 Úvod do programování v Pythonu

## Povinné domácí úkoly

Úkoly v této sadě řešte do připravených souborů `find.py`, `header.py`, `sum_paper.py`, `collect_sequences.py`, `count_atoms.py`, `birthdays.py`, které pak odevzdáte do odevzdáárny. Doctesty a typové anotace v těchto souborech považujte za součást zadání (proto v tomto zadání nejsou vzorové vstupy a výstupy ani testovací buňky).

Doctesty spustíte z příkazového řádku:

```
python -m doctest ***.py          # default mode
python -m doctest ***.py -v       # verbose mode
```

Při spouštění testů je důležité, abyste byli přímo ve složce, kde máte uložené programy, jinak nebudou sedět relativní cesty a testy neprojdou. U doctestů je uvedena formulka `# doctest: +NORMALIZE_WHITESPACE`, která zaručí tolerantnější porovnávání bílých znaků (např. že ve výstupu nemusí přesně sedět rozdělení na řádky).

Všechny soubory ve složce `data` jsou v kódování UTF-8, stejné kódování použijte i při zápisu výstupních souborů.

### DÚ 9.1: Hledá se Nemo

V souboru `find.py` doplňte funkci `find`, která vezme dva argumenty – hledané slovo a cesta k prohledávanému souboru. Funkce vrátí seznam všech řádků v souboru, které obsahují hledané slovo. Do textu řádku nezahrňujte znak nového řádku na konci.

### DÚ 9.2: Hlavička souboru

V souboru `header.py` doplňte funkci `print_header`, která vezme jeden argument – cestu k souboru. Funkce **vypíše** na výstup prvních 10 řádků tohoto souboru (nebo méně, je-li kratší). Pokud soubor neexistuje nebo ho nelze načíst, vypíše se `{nazev_souboru} not found`.

### DÚ 9.3: Sběr papíru

Soubor `data/paper.txt` obsahuje informace o sběru papíru – kdy, kdo, a kolik kg donesl (první řádek je hlavička).

V souboru `sum_paper.py` doplňte funkci `sum_paper`, která vezme jeden argument – cestu k souboru s informacemi o sběru. Funkce načte tento soubor a spočítá celkovou hmotnost sesbíraného papíru pro každou osobu. Funkce vrátí slovník se jmény osob (klíče) a hmotnostmi papíru (hodnoty). Záznamy ve slovníku budou seřazeny podle hmotnosti sestupně.

Tip: Na seřazení slovníku podle hodnot lze použít tento postup: předělat na seznam dvojic (hodnota, klíč), seznam seřadit (řazení dvojic probíhá podle prvního prvku, tj. podle hodnoty), a pak předělat zpátky na slovník. Předělávání ze slovníku na seznam a zpátky se nejlíp dělá pomocí generátorových výrazů.

## DÚ 9.4: Sběr sekvencí

Formát FASTA slouží k ukládání sekvencí nukleových kyselin (DNA, RNA) a proteinů. Tento formát je velmi jednoduchý – před každou sekvencí je řádek začínající znakem `>` s názvem sekvence, pak následuje samotná sekvence – viz soubor `data/collected_seqs-expected.fasta`.

Naším úkolem je doplnit v souboru `collect_sequences.py` funkci `collect_sequences`, která vezme dva argumenty – cestu ke vstupní složce X a k výstupnímu souboru Y.

Funkce projde všechny soubory s příponou `.txt` ve složce X, načte z nich sekvence a uloží je do souboru Y ve formátu FASTA. Název každé sekvence bude název souboru, ze kterého byla načtena (bez přípony). Sekvence budou seřazeny abecedně podle svého názvu (nejvhodnější je seřadit soubory před tím, než je budeme procházet). Funkce nemá nic vracet ani vypisovat, sesbírané sekvence má uložit souboru Y.

Pokud bude program fungovat správně, měl by se obsah výstupního souboru `data/collected_seqs.fasta` shodovat s `data/collected_seqs-expected.fasta`.

**Poznámka:** První doctest volá vaši funkci, druhý doctest volá pomocnou funkci `diff`, která porovná váš výstupní soubor se vzorovým výstupním souborem. Funkce `diff` vypíše `Files are identical`, pokud oba soubory existují a jsou stejné. Pokud nejsou stejné, vypíše rozdíly (– znamená chybějící řádek, + řádek navíc).

## DÚ 9.5: Počítáme atomy

Formát PDB slouží k ukládání struktur biomolekul. Obsahuje pro každý atom v molekule jeho typ (uhlík, vodík...), 3D souřadnice a další informace. PDB soubory obsahují ještě řádky s různými doplňujícími informacemi, ale řádky týkající se atomů poznáme podle toho, že vždy začínají řetězcem `ATOM` nebo `HETATM`. Označení prvku je vždy na 76.–77. znaku řádku (číslováno od nuly).

V souboru `count_atoms.py` doplňte funkci `count_atoms`, která vezme jeden argument – cestu k PDB souboru. Funkce vrátí slovník s typy obsažených atomů (klíče) a počty atomů každého typu (hodnoty). Klíče ve slovníku budou seřazeny podle abecedy.

## DÚ 9.6: Narodeninový problém

Alice a Bob pořádali večírek, na který přišlo 30 lidí. Dva lidi na večírku zjistili, že mají narozeniny ve stejný den. „To je ale náhoda,“ říká Bob.

Ve skutečnosti je však pravděpodobnost, že mezi 30 lidmi se najdou dva se stejným dnem narozenin, nečekaně velká (kolem 70 %). Tato skutečnost se označuje jako **narodeninový paradox**.

Naším úkolem je pro zadané číslo  $n$  odhadnout pravděpodobnost, že mezi  $n$  lidmi dojde ke kolizi narozenin (tj. že se najdou aspoň dva, kteří mají narozeniny ve stejný den).

Tento úkol budeme řešit pomocí simulace: vygenerujeme  $n$  náhodných dní v roku a podíváme se, jestli došlo ke kolizi. Toto zopakujeme dostatečný počet krát (10 000) a spočítáme, v jakém procentu pokusů došlo ke kolizi.

V souboru `birthdays.py` doplňte funkci `collision_probability`, která vezme jako argument počet lidí  $n$  a vrátí nasimulovanou pravděpodobnost, že mezi  $n$  lidmi dojde ke kolizi narozenin.

### Poznámky:

- Pro zjednodušení uvažujme, že nikdo nemá narozeniny 29. února. Dále předpokládejme, že ze zbylých 365 dní je každý den stejná pravděpodobnost narození.
- Na generování náhodných dní použijte vhodnou funkci z modulu `random`. Tip: není třeba generovat konkrétní den a měsíc (např. 11.2.), stačí nám generovat pořadové číslo dne v roku, tj. číslo od 1 do 365.
- Použijte počet pokusů = 10 000.
- Když chceme zjistit, jestli se v seznamu nacházejí dva stejné prvky, jde to zapsat velmi jednoduše pomocí funkcí `len` a `set` (nemusíme procházet všechny kombinace prvků a zkoušet jestli jsou stejné).
- Samozřejmě pokaždé, když funkci spustíme, nám může vrátit trochu jiný výsledek. Proto je v doctestech vždy uvedený přípustný interval výsledků (např. `0.65 <= collision_probability(30) <= 0.75`).