

Lista 8

Uwaga: Materiały do tej listy znajdują się w pliku <http://math.uni.wroc.pl/~jagiella/files/p1python/lista8.zip>. Jeśli otwieranie niektórych plików tekstowych sprawia problemy, funkcję `open` można wywołać z parametrem opcjonalnym `encoding='utf8'`.

Zadanie 1 (0,75 punktu). Magazyn w sklepie reprezentujemy słownikiem, który towarowi (napi-sowi) przyporządkowuje dodatnią liczbę naturalną, reprezentującą ile towaru znajduje się w ma-gazynie. Napisz funkcję `merge_stores(store1, store2)`, która dla magazynów `store1, store2` zwraca słownik reprezentujący sumę obu magazynów. Przykładowo, wywołanie:

```
merge_stores({'banan': 20, 'kokos': 10, 'maseczka': 2},
             {'chleb': 20, 'banan': 10, 'gruszka': 3})
```

powinno zwrócić słownik tożsamy z:

```
{'kokos': 10, 'maseczka': 2, 'chleb': 20, 'banan': 30, 'gruszka': 3}
```

Zadanie 2 (1,5 punktu). Dla dostatecznie dużego tekstu, język w którym został on napisany można odgadnąć na podstawie częstotliwości występujących w nim liter. Przykładowo, trzy najczęstsze litery w języku angielskim (zaniedbując ich wielkość) to odpowiednio e, t, a; natomiast w języku polskim są to odpowiednio a, i, e. Napisz funkcję `guess_language(input_file, languages)`, gdzie `input_file` to nazwa pliku zawierającego tekst w nieznanym języku, a `languages` to słownik, którego kluczami są napisy - nazwy języków, a wartościami trójki najpopularniejszych liter tego języka. Zadaniem funkcji jest odgadnięcie (na podstawie słownika `languages`) języka, w którym został napisany tekst i zwrócenie jego nazwy. Jeśli języka nie da się ustalić, funkcja powinna zwracać napis `"unknown"`.

Przykładowy test do tego zadania znajduje się w pliku `test2.py`.

Zadanie 3 (1,75 punktu). Rozważmy grupę ludzi, z których każda para może znać się ze wzajemnością lub wcale. W materiałach do listy znajdują się pliki `osoby.csv` i `relacje.txt`. Pierwszy zawiera dane o fikcyjnych osobach. Każda jego linijka zawiera, oddzielone średnikami, dane pojedynczej osoby. Te dane to kolejno:

- Imię i nazwisko (napis),
- Miasto zamieszkania (napis),
- Unikalny numer osoby (liczba naturalna).

Przykładowa linijka w pliku: `Artur K.;Poznań;56278`

Drugi z plików zawiera (kompletne) informacje o tym, które osoby się znają. Każda jego linijka zawiera parę różnych liczb naturalnych n, m (najpierw mniejszą z nich) oddzielonych spacją, oznaczającą, że osoby o numerach n i m wzajemnie się znają. Przykładowa linijka w pliku: `52340 56278`
Napisz program, który wczytuje dane z obu plików i zawiera następujące funkcje:

- `knows_people(n)` - zwraca uporządkowaną listę numerów osób, które znają co najmniej n osób.
- `related_to_city(city_name)` - zwraca uporządkowaną listę numerów osób, które mieszkają lub znają kogoś mieszkającego w mieście o nazwie `city_name`.
- `special_cities()` - zwraca uporządkowaną listę nazw miast, w których mieszka osoba, która zna dowolnego Patryka.
- `special_cities2()` - zwraca uporządkowaną listę nazw miast, w których mieszka osoba, która zna kogoś, kto zna dowolnego Patryka.

Uwaga: W powyższych podpunktach przyjmujemy, że żadna osoba nie zna samej siebie.