12 października 2021

Kurs rozszerzony języka Python

Lista 1.

Każde zadanie jest warte 2 punkty. Na pracowni do oceny należy przedstawić trzy zadania.

Zadanie 1.

W Polsce podatek od towarów i usług (VAT) liczy się na dwa sposoby: w przypadku faktur sumuje się wartości netto i mnoży się przez 23%, a w przypadku kas fiskalnych i paragonów liczy się VAT 23% od każdej pozycji osobno i na końcu się sumuje. Zaprogramuj w Pythonie dwie funkcje zwracające podatek VAT dla zadanej listy zakupów

- `vat_faktura(lista)`
- `vat_paragon(lista)`

gdzie `lista` jest listą liczb reprezentujących cenę netto. Zbadaj eksperymentalnie, czy te dwie funkcje dają te same wyniki:

```
zakupy = [0.2, 0.5, 4.59, 6]
print(vat_faktura(zakupy) == vat_paragon(zakupy))
```

Zbadaj, czy reprezentacja liczb za pomocą klasy `Decimal` daje inną odpowiedź.

Zadanie 2.

Napisz funkcję `is_palindrom(text)`, która zwraca **True** jeśli argument jest palindromem. Zakładamy, że `text` może być zarówno pojedynczym słowem (np. *rotor* czy *oko*), ale też dłuższym wyrażeniem: *"Kobyła ma mały bok."*; w takim przypadku ignorujemy znaki przestankowe, spacje i wielkość liter.

Sprawdź, czy funkcja poprawnie działa dla tekstów obcojęzycznych:

```
is_palindrom("Eine güldne, gute Tugend: Lüge nie!")
```

Zadanie 3.

4 października jest Światowy Dzień Tabliczki Mnożenia. Zaprogramuj funkcję `tabliczka(x1, x2, y1, y2)`, która wypisze na ekran tabliczkę mnożenia dla liczb $[x_1, \ldots, x_2] \times [y_1, \ldots, y_2]$; np. `tabliczka(3, 5, 2, 4)` powinno wypisać

```
   3  4  5
2  6  8 10
3  9 12 15
4 12 16 20
```

Zwróć uwagę, by szerokości kolumn były jednakowe oraz odpowiednie do liczby cyfr w liczbach.

Zadanie 4.

Zaprogramuj eksperyment polegający na rzucie monetą. Będziemy rzucać tak długo, aż trzy razy pod rząd wypadnie ta sama strona monety. Zaprogramuj ten eksperyment tak, aby wykonał zadaną jako parametr liczbę eksperymentów i na koniec podał, ile średnio wykonano rzutów w serii. Eksperyment powinien być tak zaprogramowany, aby można było czekać na dowolną liczbę orłów lub reszek pod rząd.

Zadanie 5.

Zaprogramuj funkcję, która dla zadanej listy stringów `lista_slow` zwróci najdłuższy wspólny prefiks dla przynajmniej trzech elementów `lista_slow`. Na przykład[1]

```
common_prefix(["Cyprian", "cyberotoman", "cynik", "ceniąc",  "czule"])
```

powinno zwrócić

```
"cy"
```

Wielkość liter nie ma dla nas znaczenia.

Marcin Młotkowski

[1] Inspiracja: *Cyberiada*, Stanisław Lem