

# Kurs języka Python

## Lista 1.

### 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))
```

### Zadanie 2.

W portfelu znajdują się monety 1, 2 i 5 złotych, a także banknoty 10 i 20 złotych. Za zakupy jest do zapłacenia 123 zł. Można rachunek zapłacić na różne sposoby, np. dając 123 złotówki albo sześć dwudziestozłotówek + jedna dwuzłotówka + złotówka. Napisz program, który dla zadanej kwoty podaje jakie i ile monet i banknotów mam wyjąć z portfela, aby łącznie wyjąć jak najmniej monet i banknotów. Zakładam, że monet i banknotów mam zawsze wystarczająco dużo.

### Zadanie 3.

Zaprogramuj funkcję `romb(n)`, która wypisze romb o wysokości  $2 * n - 1$ . Poniżej przykład takiego rombu dla wywołania `romb(4)`:

```
#
###
#####
#####
#####
#####
#####
#####
###
#
```

### Zadanie 4.

Napisz program który szyfruje tekst za pomocą następującego algorytmu opartego na algorytmie XOR: do zaszyfrowania jest potrzebny klucz  $k$ , tj. liczba z przedziału  $[0 \dots 255]$ . Kolejne litery tekstu zamieniamy na odpowiedni kod ASCII, obliczamy wynik operacji XOR z  $k$  i do szyfrogramu wstawiamy wynik operacji zamieniony na odpowiedni znak ASCII. Na przykład tekst *Python* za pomocą klucza 7 (binarnie: 0000 0111) szyfrujemy tak:

litery	P	y	t	h	o	n
ASCII	0101 0000	0111 1001	0111 0100	0110 1000	0110 1111	0110 1110
XOR	0101 0111	0111 1110	0111 0011	0110 1111	0110 1000	0110 1001
szyfr	W	~	s	o	h	i

Program ma mieć postać funkcji `zaszyfruj(tekst, klucz)`, która dla podanego tekstu i klucza zwraca zaszyfrowany tekst. Zaprogramuj również funkcję `odszyfruj(szyfr, klucz)`.

#### Zadanie 5.

Napisz jednoargumentową funkcję `rozklad(n)` która oblicza rozkład liczby  $n$  na czynniki pierwsze i zwraca jako wynik listę par  $[(p_1, w_1), (p_2, w_2), \dots, (p_k, w_k)]$  taką, że

$n = p_1^{w_1} * p_2^{w_2} * \dots * p_k^{w_k}$  oraz  $p_1, \dots, p_k$  są różnymi liczbami pierwszymi. Na przykład

```
>>> rozklad(756)
[(2, 2), (3, 3), (7, 1)]
```

#### Zadanie 6.

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

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

Zadbaj o ładne formatowanie, tj. wyrównanie kolumn.

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

*Marcin Młotkowski*