

Technologie informacyjne i komunikacyjne

Temat III. Python.

Przykłady – seria 2. Kolekcje.

Zadanie 1. `times` – Tabliczka mnożenia.

Napisz funkcję `times` przyjmującą jako argument dowolną liczbę naturalną n i zwracającą listę zawierającą kolejno liczby

$$n, 2n, 3n, \dots, 10n.$$

Korzystając z tej funkcji, napisz program `times`, który przyjmuje jako argument wywołania liczbę naturalną, a następnie wypisuje na standardowe wyjście tabliczkę mnożenia przez tę liczbę.

Przykładowe wykonanie

Wywołanie:

```
Windows:      py times.py 3
macOS, Linux: python3 times.py 3
```

Wyjście:

```
1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
4 x 3 = 12
5 x 3 = 15
6 x 3 = 18
7 x 3 = 21
8 x 3 = 24
9 x 3 = 27
10 x 3 = 30
```

Zadanie 2. `mostfrequent` – Najczęściej występujący element listy.

Napisz funkcję `most_frequent` przyjmującą jako argument dowolną listę i zwracającą ten element tej listy, który występuje w niej najczęściej. W przypadku, gdy elementów takich jest kilka, funkcja może zwrócić dowolny z nich.

Korzystając z tej funkcji, napisz program `mostfrequent`, który wczytuje ze standardowego wejścia listę oddzielonych przecinkami elementów dowolnego typu i wypisuje na standardowe wyjście jej najczęściej powtarzający się element lub jeden z takich elementów, gdy jest ich kilka.

Przykładowe wykonanie

Wywołanie:

```
Windows:      py mostfrequent.py
macOS, Linux: python3 mostfrequent.py
```

Wejście:

```
1,3,żółw,3,Ala ma kota,żółw,3,3.14 [Enter]
```

Wyjście:

```
3
```

Zadanie 3. lotto – Losowanie liczb.

Napisz program `lotto`, który losuje sześć różnych liczb całkowitych z przedziału od 1 do 49 włącznie, a następnie wypisuje te liczby na standardowe wyjście w kolejności rosnącej.

Wskazówka. Wykorzystaj zbiór (`set`). Generowanie całkowitych liczb pseudolosowych umożliwia funkcja `randint` z modułu `random`.

Przykładowe wykonanie

Wywołanie:

```
Windows:      py lotto.py
macOS, Linux: python3 lotto.py
```

Wyjście:

```
13 17 24 34 37 46
```

Zadanie 4. intersection – Część wspólna zbiorów liczb całkowitych.

Napisz funkcję `intersection` przyjmującą jako argumenty dwie listy liczb całkowitych oraz zwracającą zbiór złożony z liczb, które są elementami jednocześnie obu tych list.

Korzystając z tej funkcji, napisz program `intersection`, który wczytuje ze standardowego wejścia dwa ciągi liczb całkowitych oddzielonych spacjami, zapisane w osobnych liniach, i wypisuje na standardowe wyjście te liczby, które pojawiają się w obu ciągach, w kolejności rosnącej, każdą tylko raz.

Przykładowe wykonanie

Wywołanie:

```
Windows:      py intersection.py
macOS, Linux: python3 intersection.py
```

Wejście

```
1 25 7 3 [Enter]
3 4 -9 1 144 [Enter]
```

Wyjście

```
1 3
```

Zadanie 5. words – Liczenie słów.

Napisz program `words`, który wczytuje ze standardowego wejścia ciąg słów oddzielonych spacjami, a następnie wypisuje te słowa na standardowe wyjście w kolejności alfabetycznej, każde tylko raz, podając przy każdym z nich liczbę jego wystąpień oraz listę pozycji w ciągu, na których słowo to występuje.

Wskazówka. Wykorzystaj dwa słowniki (`dict`). W obu kluczem niech będzie słowo, zaś wartością – w pierwszym słowniku: liczba wystąpień tego słowa, w drugim słowniku: lista pozycji w ciągu, na których to słowo występuje.

Przykładowe wykonanie

Wywołanie:

```
Windows:      py words.py
macOS, Linux: python3 words.py
```

Wejście

```
pies kot źółw kot kot pies źółw [Enter]
```

Wyjście

```
kot 3 [2, 4, 5]
pies 2 [1, 6]
źółw 2 [3, 7]
```

Zadanie 6. histogram – Histogram tekstowy.

Napisz program `histogram`, który przyjmuje jako argumenty wywołania ciąg nieujemnych liczb całkowitych, a następnie wypisuje na standardowym wyjściu ciąg „słupków” złożonych z gwiazdek (*), wyrównanych do dołu. Słupków powinno być tyle, ile wczytanych liczb, a wysokości (tzn. ilości gwiazdek) w kolejnych z nich powinny odpowiadać kolejnym z wczytanych liczb.

Przykładowe wykonanie

Wywołanie:

Windows: `py histogram.py 1 3 2 8 5`

macOS, Linux: `python3 histogram.py 1 3 2 8 5`

Wyjście

```
*
*
*
**
**
* **
****
*****
```

Zadanie 7. Ciąg Collatza.

Niech k będzie liczbą naturalną. Ciągami Collatza nazywamy ciąg $(c_n^k)_{n=0}^\infty$ określony wzorem

$$c_n^k \stackrel{\text{def}}{=} \begin{cases} k, & \text{gdy } n = 0, \\ \frac{1}{2}c_{n-1}, & \text{gdy } c_{n-1} \text{ jest liczbą parzystą,} \\ 3c_{n-1} + 1, & \text{gdy } c_{n-1} \text{ jest liczbą nieparzystą.} \end{cases}$$

Na przykład

$$(c_n^{12}) = (12, 6, 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots),$$

$$(c_n^{13}) = (13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots),$$

$$(c_n^{15}) = (15, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots).$$

Gdy pewien wyraz ciągu Collatza ma wartość 1, wyrazy następujące po nim mają wartości 4, 2, 1, 4, 2, 1, ... – sekwencja 4, 2, 1 powtarza się w nieskończoność. Przypuszcza się, że ciąg Collatza (c_n^k) osiąga wartość 1 dla dowolnego k , hipoteza ta pozostaje jednak problemem otwartym.

- Napisz program `collatz`, który wczytuje ze standardowego wejścia liczbę naturalną k , a następnie wypisuje na standardowe wyjście początkowe wyrazy ciągu Collatza (c_n^k) aż do pierwszego z wyrazów o wartości 1 włącznie.
- Napisz program `lcollatz` znajdujący tę liczbę naturalną $k < 10^6$, dla której ciąg Collatza (c_n^k) ma najwięcej wyrazów występujących przed pierwszym z wyrazów o wartości 1.

Opracowanie: Bartłomiej Zglinicki.