

Kurs rozszerzony języka Python

Lista 4.

Z poniższych zadań wybierz jedno i rozwiąż. Zadbaj o to, aby nie generować niepotrzebnie list (np. ze wszystkimi potencjalnymi rozwiązaniami). Rozwiązania mogą być oparte o metodę *brute force*. Funkcja rozwiązująca zadanie powinna zwracać iterator, tak aby można było uzyskać wszystkie rozwiązania np. wykorzystując instrukcję `for-in`:

```
for rozwiazanie in rozwiazywanie_zadania(dane_wejscowe):
    print(rozwiazanie)
```

Można założyć, że dane wejściowe są zawsze poprawne i nie trzeba ich dodatkowo sprawdzać.

Każde zadanie jest warte 4 punkty.

Zadanie 1.

Kryptarytm to zadanie, w którym litery należy zastąpić cyframi tak, aby powstało poprawne działanie. Przykładem takiego kryptarytmu jest

```
  KIOTO
+ OSAKA
-----
  TOKIO
```

Napisz program rozwiązujący takie kryptarytmy. Przyjmij, że dane wejściowe zawierają trzy słowa i operator.

Zadanie 2.

Poniższe zadanie polega na rekonstrukcji dwuwymiarowego obrazu na podstawie rzucanego cienia. Zakładamy, że obraz jest prostokątem czarno-białych pikseli. Cień to dwa wektory, opisujące ile jest zaczernionych pikseli w wierszu bądź kolumnie. Poniżej przykład obrazu rozmiaru 4×4 :

```
  2  1  3  1
  █  █  █  █  1
  █  █  █  █  3
  █  █  █  █  1
  █  █  █  █  2
```

którego cień opisują dwa wektory: $H = (2, 1, 3, 1)$, $V = (1, 3, 1, 2)$. Dla danego cienia może istnieć wiele różnych obrazów. Wystarczy, jeśli program znajdzie jeden.

Zadanie 3.

W popularnej łamigłówce *sudoku* zadanie polega na wypełnieniu diagramu 9×9 cyframi od 1 do 9 tak, aby w każdym wierszu i każdej kolumnie żadna cyfra się nie powtarzała. Dodatkowo, w każdym podkwadracie 3×3 nie może powtarzać się żadna cyfra. Poniżej jest przykład prawidłowo wypełnionego diagramu:

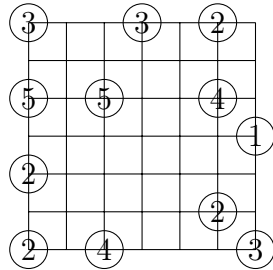
5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Zaprogramuj funkcję `rozwiązanie_sudoku(s)` która dla częściowo wypełnionego diagramu `s` zwraca poprawne jego wypełnienie (bądź **None** gdy nie ma rozwiązania). Reprezentacja diagramu jest dowolna.

Zaprogramuj również funkcję, która wyświetli czytelnie diagram.

Zadanie 4.

*Buduj mosty!*¹ polega na uzupełnianiu diagramu



Zadanie polega na zbudowaniu mostów między wyspami, oznaczonymi kółkami na diagramie wg. następujących zasad:

- mosty są tylko w pionie lub poziomie;
- każdy most łączy dwie wyspy;
- mosty się nie przecinają, nie mogą też przechodzić przez wyspy;
- dwie wyspy mogą łączyć co najwyżej dwa mosty.

Liczby w kółkach oznaczają, ile mostów dochodzi do danej wyspy. Przykładowe rozwiązanie można znaleźć na stronie Wikipedii.

Marcin Młotkowski

¹Znana też jako *Hashiwokakero*