Lista 4A

Uwaga: W celu zrobienia zadań 2 i 3, zapoznaj się z materiałami na stronie `http://math.uni.wroc.pl/~jagiella/files/p1python/mat05.html`.

Zadanie 1 (1 punkt). Ciąg Fibonacciego jest zdefiniowany poprzez

$$f_0 = 0,\ f_1 = 1,$$
$$f_n = f_{n-1} + f_{n-2},\ (n \geqslant 2).$$

Napisz funkcje obliczające wartość f_n trzema sposobami: iteracyjnie, rekurencyjnie, oraz z wykorzystaniem wzoru[1]:

$$f_n = \left\lfloor \frac{\phi^n}{\sqrt{5}} + \frac{1}{2} \right\rfloor,\ \phi = \frac{1+\sqrt{5}}{2}.$$

Ze względu na sposób, w jaki Python reprezentuje liczby rzeczywiste, implementacja za pomocą wzoru może nie zwracać poprawnych wyników. Znajdź najmniejsze n, dla którego wersja używająca wzoru podaje wynik różniący się od rzeczywistego.

Zadanie 2 (1 punkt) Niech `N` będzie liczbą naturalną. Zapisz jako listy składane (czyli pojedyncze wyrażenia postaci `[...]`) następujące listy:

- Listę napisów `''`, `'ab'`, `'aabb'`, `'aaabbb'`, ..., w której ostatni napis ma długość 2`N`.
- Listę wszystkich par liczb (n, m) dla naturalnych $n \leqslant m <$ `N` (każda para powinna się pojawić dokładnie raz, ich kolejność jest dowolna).
- Listę list postaci `[]`, `[1]`, `[-2, -2]`, `[4, 4, 4]`, `[-8, -8, -8, -8]`, `[16, 16, 16, 16]`, ..., której ostatnim elementem jest (odpowiedniej postaci) lista długości `N`.

Zadanie 3 (2 punkty) Sudoku[2] to łamigłówka logiczna, polegająca na rozmieszczeniu na planszy 9×9 (podzielonej na dziewięć „podregionów" 3×3) liczb naturalnych od 1 do 9 (z powtórzeniami) w taki sposób, aby jednocześnie:

- Każdy wiersz zawierał wszystkie liczby od 1 do 9.
- Każda kolumna zawierała wszystkie liczby od 1 do 9.
- Każdy z dziewięciu „podregionów" zawierał wszystkie liczby od 1 do 9.

Przykładowe rozwiązanie[3] Sudoku (pogrubione linie oddzielają „podregiony" 3×3):

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

Napisz funkcję `check_solution(board)`, która dla listy `board` reprezentującej planszę 9×9 wypełnioną liczbami rozstrzyga (zwracając `True` lub `False`), czy jest ona poprawnym rozwiązaniem Sudoku. `board` jest listą dziewięciu list, z których każda reprezentuje pojedynczy wiersz planszy (listę dziewięciu liczb)[4].

---

[1] $\lfloor x \rfloor$, czyli *podłoga z* x: zaokrąglenie x w dół.

[2] `https://en.wikipedia.org/wiki/Sudoku`

[3] Zazwyczaj rozwiązywanie takich łamigłówek rozpoczyna się od częściowo wypełnionej planszy, narzucającej (często jedyne) rozwiązanie.

[4] Możesz przetestować swoje rozwiązanie przy użyciu pliku `http://math.uni.wroc.pl/~jagiella/files/p1python/sudoku_test.py`