

Lista 5

Zadanie 1 (1 punkt). Rozważ (lub przetestuj) działanie następującej funkcji:

```
def counting_down_list(n):
    lst = []
    for i in range(n):
        lst = [i] * i + lst
    return lst
```

Następnie przepisz tę funkcję do postaci:

```
def counting_down_list(n):
    return [...]
```

gdzie [...] jest schematem konstrukcji listy (tj. listą składaną).

Zadanie 2 (1 punkt). Zaimportuj funkcje: `binom` z modułu `binom` i `binom2` z modułu `pascal` (w materiałach do wykładów 4 i 5), oraz `perf_counter` z wbudowanego modułu `time`. Wyznacz dla jakiego najmniejszego `n` czas wykonania `binom(n, n//2)` przekracza sekundę¹. Następnie zbadaj czasy wykonania `binom2(i, i//2)` dla `i = 1, 2, ..., 10·n`.

Zadanie 3 (1 punkt). Napisz funkcję `pearson_corr(lst1, lst2)`, która dla niepustych list liczb rzeczywistych `lst1=[x0, x1, ..., xn-1]` i `lst2=[y0, y1, ..., yn-1]` oblicza i zwraca ich współczynnik korelacji Pearsona² zdefiniowany wzorem:

$$\text{corr}(x, y) = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=0}^{n-1} (y_i - \bar{y})^2}},$$

gdzie

$$\bar{x} = \frac{1}{n} \sum_{i=0}^{n-1} x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=0}^{n-1} y_i.$$

Zadanie 4 (1 punkt). Rozważmy planszę (macierz) rozmiaru $n \times m$, w której wiersze indeksowane są liczbami $0, \dots, n - 1$, a kolumny liczbami $0, \dots, m - 1$. Sąsiadami pola o współrzędnych (i, j) nazywamy wszystkie pola sąsiadujące z nim brzegiem lub na ukos. Napisz funkcję przyjmującą jako argumenty rozmiar planszy oraz współrzędne pola, zwracającą listę współrzędnych sąsiednich pól (w dowolnej kolejności, ale bez powtórzeń).

¹Wartość `n` zależy od platformy (komputera). Kod napisz tak, aby znajdował tę wartość dla komputera, na którym został uruchomiony. Przykład pomiaru czasu: `timing.py` z wykładu 3.

²https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#For_a_sample