

## Lista zadań nr 3

**Zadanie 1** Napisz nieskończony generator produkujący kolejne dodatnie liczby całkowite, następnie napisz nieskończony generator kolejnych kwadratów dodatnich liczb całkowitych - wykorzystaj poprzedni generator. Zaprojektuj i napisz funkcję `select()`, która tworzy  $n$ -elementową listę wartości dowolnego obiektu iterowalnego - przetestuj funkcję na zdefiniowanych wcześniej nieskończonych generatorach (w definicji użyj funkcje `iter()` oraz `next()`). Zdefiniuj generator produkujący trójelementowe krotki zawierające tzw. trójki pitagorejskie tzn. krotki postaci  $(a, b, c)$  gdzie  $a^2 + b^2 == c^2$  - wykorzystaj wyrażenie generatora (załóż, że  $a < b < c$ ). Wyświetl 15 pierwszych trójek korzystając z funkcji `select()`.

**Zadanie 2** Napisz klasę, która implementuje iterator ciągu Fibonacciego zwracającego kolejne wyrazy ciągu mniejsze od  $n > 0$ . Wykonaj to samo zadanie pisząc odpowiedni generator (funkcję generatora). Następnie, utwórz iterator (na podstawie nieskończonego generatora ciągu Fibonacciego) zwracający liczby Fibonacciego od  $F_{100000}$  do  $F_{100020}$  i zapisz te liczby do pliku tekstowego. Ile cyfr ma liczba  $F_{100000}$ ?

**Zadanie 3** Napisz generator `gen_time()`, który produkuje kolejne sekwencję czasu w postaci krotki (godziny, minuty, sekundy). Generator powinien przyjmować w postaci krotek czasu czas startowy, czas końcowy i krok czasu. Zamiast zwykłych krotek możesz skorzystać z krotek nazwanych. Przykładowe działanie:

```
>>> for time in gen_time((8, 10, 00), (10, 50, 15), (0, 15, 12) ):
    print(time)
```

```
(8, 10, 0)
(8, 25, 12)
(8, 40, 24)
(8, 55, 36)
(9, 10, 48)
(9, 26, 0)
(9, 41, 12)
(9, 56, 24)
(10, 11, 36)
(10, 26, 48)
(10, 42, 0)
>>>
```

**Zadanie 4** Napisz "generator" kolejnych liczb pierwszych. Jak dużą liczbę pierwszą potrafisz wygenerować? Zapisz 10000 początkowych liczb pierwszych do pliku.

**Zadanie 5** Napisz rekurencyjną funkcję generatora, która pozwoli przekształcać zagnieźdzoną sekwencję na postać jednowymiarową w postaci listy wartości (pomijaj ciągi tekstowe) tj. na przykład

$$([1, 'kot'], 3, (4, 5, [7, 8, 9])) \rightarrow [1, 'kot', 3, 4, 5, 7, 8, 9].$$