

## Lab 04 Generatory

### Zad 1.

Stwórz funkcję **kwadrat\_liczb (limit)**, która przyjmuje jako argument limit (liczbę całkowitą) i zwraca generator kolejnych kwadratów liczb naturalnych od 1 do podanego limitu (włącznie).

**Przykład użycia:**

```
for liczba in kwadrat_liczb(5):
    print(liczba)
```

### Zad 2

Stwórz funkcję **liczby\_pierwsze (start, end)**, która przyjmuje jako argumenty początek i koniec zakresu liczb, a następnie zwraca generator kolejnych liczb pierwszych w tym zakresie.

**Przykład użycia:**

```
for liczba in liczby_pierwsze (2, 20):
    print(liczba)
```

### Zad 3

Stwórz funkcję **polacz\_listy(list1, list2)**, która przyjmuje jako argumenty dwie listy i zwraca generator, który zwraca na przemian elementy z pierwszej i drugiej listy. Jeśli jedna z list jest dłuższa generator powinien kontynuować zwracanie elementów z dłuższej listy, gdy elementy z krótszej się skończą. *Sugestia: Wewnątrz generatora dostęp do elementów list powinien być z wykorzystaniem funkcji next*

**Przykład użycia:**

L

```
list1 = [1, 2, 3, 4]
list2 = ['a', 'b', 'c', 'd', 'e', 'f']

for item in polacz_listy(list1, list2):
    print(item, end="-")
```

1-a-2-b-3-c-4-d-e-f-

### Zad 4

Stwórz funkcję **ciag\_lucasa (limit)**, która przyjmuje jako argument limit (liczbę całkowitą) i zwraca generator liczb z ciągu lucasa do podanego limitu (włącznie).

[Ciąg \(liczby\) Lucasa \(math.edu.pl\)](http://math.edu.pl)

```
for liczba in ciag_lucasa (100):
    print(liczba)
```