

Zadanie 1. Napisz funkcję sprawdzającą czy liczba przesłana jako argument jest liczbą pierwszą (wykorzystaj *else* do pętli *for*). Napisz funkcję zwracającą podaną liczbę Mersenne'a ($M(p) = 2^p - 1$). Sprawdź które liczby Mersenne'a dla $p=1, 2, \dots, 31$ są pierwsze. Oznacz typ zwracany przez funkcję (przy pomocy `-> typ`).

<https://docs.python.org/3/tutorial/controlflow.html#defining-functions>

Zadanie 2. Napisz funkcję przyjmującą dwa argumenty: tekst do wypisania oraz szerokość linii (argument o domyślnej wartości 80), wypisującą przesłany tekst wycentrowany w linii.

Przykładowe wywołania:

```
printCenter("Zażółć gęślą jaźń")
printCenter("Zażółć gęślą jaźń", 70)
```

<https://docs.python.org/3/tutorial/controlflow.html#more-on-defining-functions>

<https://docs.python.org/3/tutorial/controlflow.html#function-annotations>

Zadanie 3. Napisz funkcję przyjmującą cztery argumenty: x początkowe (domyślnie -1), x końcowe (domyślnie 1), liczbę przedziałów (domyślnie 10) oraz funkcję fx . Funkcja powinna wypisywać wartości x oraz funkcji przesłanej jako argument fx dla x z podanego zakresu. Zaprezentuj działanie funkcji wykorzystując *lambda* wyrażenia (prześlij fx jako *lambda*).

Przykładowe wywołania:

```
>>> printFx(lambda x: int(x**2), -1, 1, 10)
-1.0 -0.8414709848078965
-0.7777777777777778 -0.7016978761467353
-0.5555555555555556 -0.5274153857718655
-0.33333333333333337 -0.32719469679615226
-0.11111111111111116 -0.11088262850995304
0.11111111111111116 0.11088262850995304
0.33333333333333326 0.32719469679615215
0.5555555555555554 0.5274153857718653
0.7777777777777777 0.7016978761467352
1.0 0.8414709848078965
>>> printFx(lambda x: sin(x), -1, 1, 10)
-1.0 -0.8414709848078965
0.0 0.0
1.0 0.8414709848078965
>>> printFx(lambda x: x**2, -1, 1, 10)
0.0 1.0
1.0 3.0
2.0 5.0
```

<https://docs.python.org/3/tutorial/controlflow.html#keyword-arguments>

<https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>

Zadanie 4. Napisz cztery funkcje:

1. $k4$ zwracająca pseudolosową liczbę całkowitą od 1 do 4 włącznie,
2. $k8$ zwracająca pseudolosową liczbę całkowitą od 1 do 8 włącznie,
3. $k10$ zwracająca pseudolosową liczbę całkowitą od 1 do 10 włącznie,
4. $k20$ zwracająca pseudolosową liczbę całkowitą od 1 do 20 włącznie.

[spoiler: strona 3]

Napisz funkcję *getDice* przyjmującą jako argument zmienną tekstową, która:

1. Dla tekstu "k4" zwraca funkcję *k4*,
2. Dla tekstu "k8" zwraca funkcję *k8*,
3. Dla tekstu "k10" zwraca funkcję *k10*,
4. Dla tekstu "k20" zwraca funkcję *k20*.

Napisz funkcję *rollDice* przyjmującą jako argument tekst w formacie "XkY", gdzie X oraz Y są liczbami naturalnymi. Funkcja ta powinna wyciągnąć z tekstu wartości X oraz Y [spoiler: strona 4], i wywołać X razy funkcję zwróconą przez *getDice* z argumentem "kY". Funkcja *rollDice* powinna zwracać sumę wylosowanych wartości.

Przykładowe wywołania:

```
>>> rollDice("1k4")
2
>>> rollDice("2k10")
15
>>> rollDice("4k8")
11
```

Co się stanie jeśli wywołamy funkcję *rollDice* z argumentami "10k4", "100k4", "1k7", "0k4"? Jak zmodyfikować funkcję, aby działało wywołanie "1e5k4" (1e5 losowań k4)?

<https://docs.python.org/3/library/random.html>

Zadanie 5. Napisz funkcję wypisującą napisy końcowe (filmu, serialu, tego programu). Funkcja ta powinna przyjmować dowolną liczbę argumentów i wypisywać każdy kolejny co sekundę, wycentrowany – najpierw argumenty zwykłe, potem kluczowe. Wykorzystaj funkcję z zadania 3.

Przykładowe wywołanie i wypisany tekst:

```
>>> credits("Cudowniejszy program pokazowy",
            "Powrót Złego Markiza Cul-de-Sac",
            Scenariusz="Autor",
            Scenografia="Autor",
            Podziękowania="Python")
          Cudowniejszy program pokazowy
          Powrót Złego Markiza Cul-de-Sac
          Scenografia: Autor
          Scenariusz: Autor
          Podziękowania: Python
```

Do pobrania długości tekstu można użyć funkcji *len()*. Do czekania wykorzystaj funkcję *time.sleep()* – aby jej użyć należy zaimportować moduł *time*:
`import time`

Dodaj dokumentację do funkcji – opisz jakie argumenty przyjmuje i co robi.

<https://docs.python.org/3/tutorial/controlflow.html#arbitrary-argument-lists>

<https://docs.python.org/3/tutorial/controlflow.html#documentation-strings>

ad. Zadanie 4. Losowanie:

```
import random
```

```
def k20():  
    return random.randint(1, 20)
```

ad. Zadanie 4. Dzielenie tekstu:

```
tekst="2k8"
```

```
liczby=tekst.split("k")
```

```
print(liczby[0], liczby[1])
```

```
# a, b=tekst.split("k")
```

```
# print(a, b)
```