

```
1 #Zadanie "Dzielenie"
2 # Napisz funkcję, która otrzyma dwa argumenty pierwszym będzie liczba, którą chcemy
3 # podzielić bez reszty a drugim argumentem będzie dzielnik. Należy sprawdzić czy
4 # można dokonać dzielenia a jeśli tak zwrócić informację czy liczba jest podzielna bez reszty czy nie.
5
6 def dzielenie(dzielna, dzielnik):
7     if dzielnik == 0:
8         return "Nie można dzielić przez zero!"
9     elif dzielna % dzielnik == 0:
10         return f"Liczba {dzielna} jest podzielna całkowicie przez {dzielnik}."
11     else:
12         return f"Liczba {dzielna} nie jest podzielna przez {dzielnik}."
13
14 print(dzielenie(34, 4))
15 print(dzielenie(8, 2))
16 #~~~~~
17
18 # Zadanie "Szablon osoby"
19 # Napisz funkcję, która przyjmuje następujące argumenty:
20 # imie (str), wiek (int), wzrost_m (float),
21 # a zwraca napis: "Jan, lat 20, 1.75 m wzrostu" - oczywiście argumenty należy podstawić do szablonu.
22 # Wzrost ma zawsze pokazywać dwa miejsca po przecinku.
23
24 def osoba(imie: str, wiek: int, wzrost_m: float):
25     return f"{imie}, lat {wiek}, {wzrost_m:2.2f} wzrostu."
26
27 #:N.Mf N miejsc na liczbe całkowitą . M miejsca na ułamek, f -> float
28 print(osoba("Alicja", 20, 1.74))
29 #~~~~~
30
31 # Zadanie "Unikalna lista"
32 # Napisz funkcję, która jako argument otrzymuje listę elementów, w której mogą
33 # występować powtórzenia, a zwraca listę unikalnych elementów.
34 # Dla [1,2,3,3,3,3,4,5] oczekujemy [1, 2, 3, 4, 5]
35
36 def unikalna_lista(lista):
37     wynik = [] #pusta lista wynikowa
38     for elem in lista:
39         if not elem in wynik:
40             wynik.append(elem)
41
42     return wynik
43
44 moja_lista = [4, 6, 2, 5, 5, 5, 4, 7, 4]
45 wynik = unikalna_lista(moja_lista)
46 print(wynik)
47 #~~~~~
48
49 # Zadanie "Suma cyfr"
50 # Napisz funkcję, która otrzymuje liczbę całkowitą, a zwraca sumę jej cyfr. (249 -> 2+4+9=15)
51
52 #pamiętamy, że string to lista znaków
53
54 def suma_cyfr(liczba: int) -> int:
55     liczba_str = str(abs(liczba))
56     wynik = 0
57     for cyfra in liczba_str:
58         wynik += int(cyfra)
59     return wynik
60
61 a = 26834
62 print(suma_cyfr(a))
63 print(suma_cyfr(-7463))
```

```
64 #~~~~~
65
66 # Zadanie "Liczby losowe"
67 # Napisz funkcję, która zwraca listę losowych liczb. Rozmiar listy zależy od argumentu.
68 # Dodatkowo: Funkcja powinna otrzymać dwa dodatkowe argumenty: minimalna i
69 # maksymalna wartość, która może zostać wylosowana.
70
71 import random
72
73 def losowa_lista(rozmiar: int, minimum: float, maximum: float):
74     lista = []
75     for i in range(rozmiar):
76         lista.append(random.randint(minimum, maximum))
77     return lista
78
79 print(losowa_lista(10, 0, 20))
80 #~~~~~
81
82 # Zadanie "Logowanie"
83 # Napisz funkcję, która zapyta użytkownika o hasło i login. Funkcja ma zwrócić 'True', jeśli
84 # podano poprawne hasło i login lub 'False' w innym przypadku.
85
86 def logowanie(pop_login: str, pop_haslo: str) -> bool:
87     login = input("Podaj login: ")
88     haslo = input("Podaj hasło: ")
89
90     #wersja 1
91     if login == pop_login and haslo == pop_haslo:
92         return True
93     else:
94         return False
95
96     #wersja 2
97     # return login == pop_login and haslo == pop_haslo
98
99 print(logowanie("login", "haslo"))
100 #~~~~~
101
102 # Zadanie "Logowanie z określoną liczbą prób"
103 # Wykorzystaj powyższą funkcję w funkcji, która pozwala na 'n' prób logowań.
104 # Zwraca 'True' jeśli udało się zalogować lub 'False' jeśli przekroczono liczbę prób.
105 # Funkcja również musi przyjmować poprawne hasło i login.
106 # Wprowadzenie niepoprawnej wartości 'n' powinno zostać obsłużone (zapytanie
107 # jednorazowe dla takich przypadków) - czyli jeśli n < 1 to przyjmujemy, że n=1.
108
109 def logowanie_n(n: int, p_login: str, p_haslo: str) -> bool:
110     n = max(1, n)
111
112     for i in range(n):
113         if logowanie(p_login, p_haslo):
114             return True
115         print(f"Niepoprawna próba numer {i+1}.")
116
117     print("Logowanie nieudane - przekroczono liczbę prób.")
118     return False
119
120 print(logowanie_n(3, "login", "haslo"))
```