

TRZECIA PRACA DOMOWA

UWAGA! UWAGA!



- Przygotuj funkcję, która odbierze dwie listy, wynikiem powinna być nowa lista, której elementami będą sumy. dla przykładu:

```
5 a = [2, 4, 6]
6 b = [8, 6, 4]
7
8 sum_lists(a, b)
9 # |zwróci [10, 10, 10]
```



TRZECIA PRACA DOMOWA

UWAGA! UWAGA!

- Pytaj użytkownika o liczbę tak długo jak długo nie napisze “koniec”. Wyświetl iloczyn wszystkich liczb parzystych podanych przez użytkownika.
- Przygotuj funkcję, która dla dowolnej liczby policzy jej silnię. Skorzystaj z pętli while. np. `get_factorial(5)` powinno zwrócić wynik $5 * 4 * 3 * 2 * 1$
- Uzupełnij poniższy kod (...) tak aby funkcja zliczała samogłoski

```
1 def count_vowels(text):
2     return sum([... for char in text])
3
4
5 print(count_vowels('ala'))
6 # 2
7 print(count_vowels('programowanie'))
8 # 6
```



TRZECIA PRACA DOMOWA

UWAGA! UWAGA!

- Przygotuj funkcję, która za pomocą wyrażenia list comprehension będzie potrafiła przefiltrować liczby parzyste z listy przekazanej w argumencie.
- Przygotuj funkcję, która za pomocą list comprehension przefiltruje tylko słowa, których długość jest większa od 4 i mniejsza od 8.
- Przygotuj funkcję która zwróci największy wspólny dzielnik dla dwóch liczb przekazanych w jej argumentach.
- Przygotuj funkcję, która odbierze argument, który określa ile kolejnych wyrazów ciągu Fibonacciego ta funkcja zwróci. Ciąg fibonacciego to taki ciąg, gdzie każdy kolejny wyraz jest sumą dwóch poprzednich.
- Dokończ poprzednią pracę domową.. zamianę liczb dziesiętnych na binarne za pomocą pętli while :)



TRZECIA PRACA DOMOWA

UWAGA! UWAGA!



- Przygotuj funkcję, która odbierze od użytkownika procent zdobytych punktów, a w odpowiedzi zwróci ocenę jaką otrzymał użytkownik.

Minimum procentowe	Ocena
<45%	niedostateczny
45%	dopuszczający
55%	dostateczny
80%	dobry
90%	bardzo dobry
95%	celujący



TRZECIA PRACA DOMOWA

UWAGA! UWAGA!

- Przygotuj funkcję, która zliczy ilość znaków w tekście zawierających się wewnątrz nawiasów okrągłych. Nawiasy mogą występować w tekście wielokrotnie, nigdy nie będą się w sobie zawierać.

```
1 def count_letters(text, start='(', end=')'):  
2     pass  
3  
4  
5     count_letters('ała) ma (kota)')  
6     # zwróci 3 + 4  
7  
8     count_letters('<> kod <103>', '<', '>')  
9     # zwróci 3  
10  
11     count_letters('abrakadabra')  
12     #zwróci 0
```



TRZECIA PRACA DOMOWA

UWAGA! UWAGA!

- Przygotuj funkcję, która odbierze dwa punkty w postaci tupli (x, y). Wynikiem powinna być długość odcinka utworzonego w wyniku połączenia tych dwóch punktów.

$$|AB| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

