

## Dodatkowe strony z zadaniami:

1. Hacker Rank - <https://www.hackerrank.com/domains/python>
2. Code Wars - <https://www.codewars.com/>
3. Exercism.io <https://exercism.io/my/tracks/python>

## Zadania dodatkowe:

Wykonując poniższe zadania pamiętaj, aby wydzielać kod za pomocą tworzenia małych funkcjonalnych funkcji. Powodzenia!

1. Mając dane dotyczące osób (imię, nazwisko, ulubiony sport, płeć, wiek) wykonaj poniższe punkty.  
Adam, Nowak, siatkówka, mężczyzna, 25  
Agata, Kowalska, pływanie, kobieta, 42  
Mateusz, Markowski, koszykówka, mężczyzna, 29  
Magda, Popławska, siatkówka, kobieta, 32
  - a) Stwórz zmienne typu słownik opisujące każdą osobę. Pamiętaj, aby zachować takie same nazwy kluczy.
  - b) Stwórz listę wszystkich osób.
  - c) Korzystając z listy z punktu b) przygotuj listę, która będzie zawierać osoby, których ulubionym sportem jest siatkówka.
  - d) Stwórz funkcję, która policzy ile osób lubi dany sport.
2. Utwórz program, który po podaniu przez użytkownika wagi w g, wyświetli wynik w dag i kg zaokrąglając do 4 miejsc po przecinku.
3. Napisz program, który wczytuje od użytkownika liczby całkowite, do momentu gdy zostanie podane 0. Program powinien wypisać ile zostało podanych liczb aparyzycznych i nieaparyzycznych.
4. Napisz program, który wczyta od użytkownika dowolną cyfrę, następnie wypisze jej tabliczkę mnożenia do 10.  
Np. dla 2:  
 $1 \times 2 = 2$   
 $2 \times 2 = 4$   
 $3 \times 2 = 6$   
...  
 $10 \times 2 = 20$
5. Napisz program, który zapyta użytkownika o imię, nazwisko, email, telefon.
  - a. Popraw zmienną imię i zmienną nazwisko, tak aby każde z nich rozpoczynało się od dużej litery( jakiej funkcji możemy użyć?)
  - b. Sprawdź czy użytkownik jest kobietą (jakiej funkcji możemy użyć?)
  - c. Połącz zmienną imię i zmienną nazwisko i zapisz pod nową nazwa: "data"
  - d. Sprawdź czy email kończy się ciągiem "com"
  - e. Usuń '-' i spacje ' ' ze zmiennej telefon
6. Napisz program, który poprosi użytkownika o ścieżkę do pliku wejściowego i wyjściowego. Program powinien wczytać plik wejściowy, zliczyć ilość linii, wyświetlić liczbę słów. Następnie zapisze nazwę pliku wejściowego i zebrane informacje do pliku wyjściowego (jeśli plik nie istnieje, niech program go utworzy).

7. Sprawdź czy podane przez użytkownika słowo jest **pangramem** i wyświetl o tym informację.

Pangram - ciąg znaków (zdanie, zdania), które zawierają wszystkie dostępne litery alfabetu.

Przykład:

Zdanie "The quick brown fox jumps over the lazy dog." jest pangramem, ponieważ conajmniej raz występuje w nim każda litera alfabetu łacińskiego.

Pamiętaj o wielkości liter ;)

8. Poproś użytkownika o podanie pojedynczego słowa, następnie pozwól użytkownikowi wprowadzić listę słów, po wpisaniu przez użytkownika znaku "0" przerwij podawanie listy słów. Sprawdź czy podane słowo na starcie posiada swój **anagram** we wprowadzonej liście słów.

Z Wikipedii: "Anagram - oznacza wyraz, wyrażenie lub całe zdanie powstałe przez przestawienie liter bądź sylab innego wyrazu lub zdania, wykorzystujące wszystkie litery (głoski bądź sylaby) materiału wyjściowego."

9. Liczenie występowania słów. Poproś użytkownika o podanie zdania lub ciągu słów. Wypisz ile razy w zdaniu wystąpiło każde słowo.

Przykład:

Wejście: "banan, truskawka, Truskawka, TRUSKAWKA, banan, jabłko, Jabłko, mandarynka".

Wyjście:

banan 2

truskawka 3

jabłko 2

(kolejność dowolna)

10. Wykonaj zadanie z punktu 4, dodatkowo wyświetlając wynik w kolejności od najczęściej występujących do najmniej.
11. Przygotuj prosty kalkulator (dodawanie, odejmowanie, mnożenie, dzielenie). Użytkownik powinien mieć możliwość wyboru działania i podania dwóch liczb. Pomyśl jakie działania warto zabezpieczyć, czy są jakieś dodatkowe warunki wykonywania pewnych obliczeń. Jeśli tak, pamiętaj o bloku try..catch.