

# Zadanie 1

Napisać funkcję, która dla podanego ciągu liczb całkowitych znajdzie dowolny spójny podciąg o największej wartości sumy jego elementów. Napisać testy dla podanej funkcji uwzględniające różne przypadki.

Wskazówka: można użyć algorytmu Kadane

# Zadanie 2

Zaimplementować funkcję szyfrującą i deszyfrującą (tylko duże litery alfabetu łacińskiego i spacja) za pomocą szyfru Vigenère'a (patrz Wikipedia). W tym przypadku w odróżnieniu od zadania przed laboratorium, kluczem będzie nie litera a napis. W przypadku jeśli nie można przeprowadzić szyfrowania lub deszyfrowania (klucz ma długość 0, klucz lub tekst szyfrowany lub deszyfrowany zawierają znaki spoza zakresu 'A' - 'Z') funkcje mają zgłaszać błąd (na co powinny być testy).

Proszę pamiętać o podziale problemu na podproblemy (napisaniu i użyciu funkcji realizujące te prostsze operacje).

Napisać testy dla podanych poniżej funkcji szyfrującej / deszyfrującej (oraz wszystkich napisanych funkcji przez nie wykorzystywanych - podproblemy).

```
def encrypt_vigenere2( key, plaintext ):
    pass

def decrypt_vigenere2( key, ciphertext):
    pass
```

# Zadanie 3

Napisać funkcję, która dla podanej listy studentów z ich wynikami z laboratoriów z przedmiotu zwróci listę studentów z informacją o łącznej liczbie punktów z laboratoriów oraz średnim procentowym wyniku z laboratorium. Wszystkie wartości średnie mają być zaokrąglone do jednego miejsca po przecinku.

Dane wejściowe: \* kolekcja zawierająca maksymalne liczby punktów dla poszczególnych ćwiczeń - rozmiar tej kolekcji określa liczbę laboratoriów, \* kolekcja z informacjami o wynikach studentów: \* imię i nazwisko studenta, \* kolekcję punktów zdobytych z poszczególnych laboratoriów - Kolekcja może zawierać za mało lub za dużo pozycji, których z wyników może nie być liczbą (lub wartością, której nie można przekształcić na liczbę).

Dane wyjściowe: \* kolekcja zawierająca listę studentów: \* imię nazwisko, \* suma punktów z laboratorium, \* średni wynik procentowy z laboratorium, \* średni wynik z laboratorium dla wszystkich studentów.

Jeżeli student nie ma wyniku z danego laboratorium (kolekcja z wynikami jest za krótka) należy przyjąć wartość 0.

Jeżeli jako wynik z laboratorium jest podana wartość inna niż liczba (nie można jej przekształcić na liczbę) należy jako sumę punktów z laboratorium i średnią procentową zwrócić wartość None oraz nie uwzględniać tego studenta w wyliczeniu wartości średniej z laboratorium dla wszystkich studentów.

Jeżeli zamiast kolekcji wyników dla studenta będzie podana pojedyncza liczba, należy jako sumę punktów z laboratorium i średnią procentową zwrócić wartość None i nie uwzględniać tego studenta w wyliczeniu wartości średniej z laboratorium dla wszystkich studentów.

Wewnętrznie w funkcji należy przechwycić i odpowiednio obsłużyć wyjątki sygnalizujące błędy (opisane wyżej oraz wynikające z obliczeń).

### Przykład 1:

```
# dane wejściowe
[10, 20, 30]
[("Adam Abacki", [5, 10, 15]), ("Basia Babacka", [10, 20, 30]), ("Cecylia Cabacka", [1, 2, 3])]

# wynik
[("Adam Abacki", 30, 50), ("Basia Babacka", 60, 100), ("Cecylia Cabacka", 6, 10)]
32
```

### Przykład 2:

```
# dane wejściowe
[10, 20, 30]
[("Adam Abacki", [5, 10, 15]), ("Basia Babacka", [10, 20, 30]), ("Cecylia Cabacka", 55)]

# wynik
[("Adam Abacki", None, None), ("Basia Babacka", 60, 100), ("Cecylia Cabacka", None, None)]
100
```