

- Rozwiązane zadania należy wysłać mailem na adres `tomasz.golan@uwr.edu.pl` nie później niż w ciągu 90 minut licząc od rozpoczęcia egzaminu.
 - Każde zadanie należy umieścić w osobnym pliku `imie.nazwisko.egzamin.XX.py` gdzie `XX` to numer zadania.
 - Punktacja:
 - bdb: ≥ 5.0 pkt
 - db+: ≥ 4.5 pkt
 - db: ≥ 4.0 pkt
 - dst+: ≥ 3.5 pkt
 - dst: ≥ 3.0 pkt
-

Zadanie 1 (2 pkt)

Napisz program wyznaczania wskaźnika BMI:

$$\text{BMI} = \frac{\text{masa [kg]}}{\text{wzrost}^2 [m^2]}$$

Program powinien przyjmować cztery argumenty wywołania:

- masę
- jednostkę masy: *kg* lub *lb* (przyjmij $1kg = 2.2lb$)
- wzrost
- jednostkę wzrostu: *cm* lub *in* (przyjmij $1in = 2.5cm$)

np. `python moj_program.py 80 kg 180 cm`

Program powinien drukować wartość wskaźnika BMI oraz odpowiadającą jej kategorię:

- $\text{BMI} \leq 18.5$ - niedowaga
- $18.5 < \text{BMI} < 25.0$ - waga prawidłowa
- $\text{BMI} \geq 25.0$ - nadwaga

Program powinien przerywać działanie i drukować odpowiedni komunikat, jeśli podane argumenty są nieprawidłowe.

Zadanie 2 (2 pkt)

W pliku `https://tomaszgolan.github.io/wfia_python/data/liczby.txt` znajdują się liczby naturalne oddzielone przecinkiem. Napisz program, który wczytuje te liczby oraz:

- znajduje największą liczbę podzielną przez 3;

- znajduje wszystkie liczby palindromiczne (czytane od lewej i od prawej strony mają taką samą wartość);
- dla każdej liczby liczy sumę jej cyfr i zapisuje wyniki w nowym pliku (z zachowaniem oryginalnej kolejności).

Zadanie 3 (2 pkt)

Niech a_k będzie ciągiem liczb naturalnych, takich że dwa pierwsze wyrazy to $a_1 = 1$ i $a_2 = 2$, a każdy następny ($a_k, k > 2$) jest najmniejszą liczbą naturalną, która do tej pory nie wystąpiła w ciągu i która nie jest względnie pierwsza z wyrazem bezpośrednio poprzedzającym (tj. $\text{NWD}(a_k, a_{k+1}) > 1$), czyli

1, 2, 4, 6, 3, 9, 12, 8, 10, 5, 15, 18, 14, 7, 21, 24, 16...

Napisz program, który drukuje n pierwszych wyrazów ciągu a_k (jak powyżej - oddzielonych przecinkami). n pobierane od użytkownika (ze standardowego wejścia). Program powinien pytać o podanie n tak długo, aż zostanie podana prawidłowa wartość (liczba naturalna). *Wskazówka: `math.gcd` wyznacza największy wspólny dzielnik.*

Dodatkowo program powinien rysować wykres (k, a_k) - przykład dla $n = 100$:

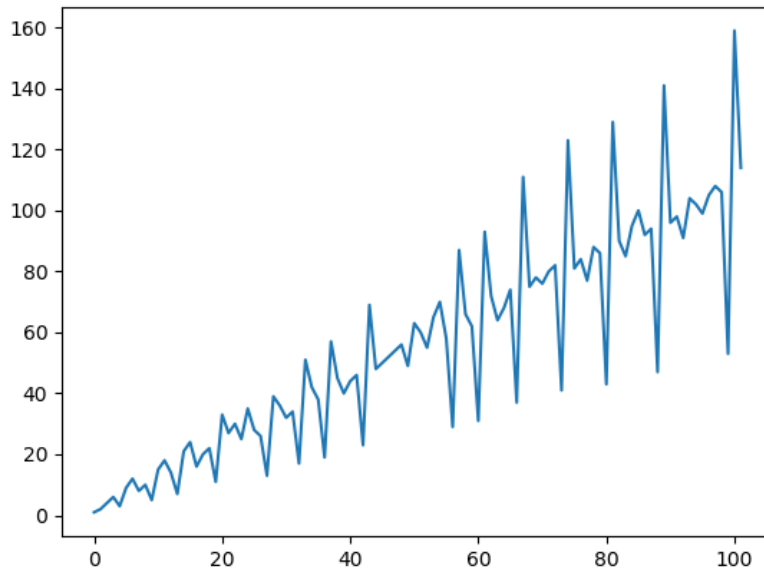


Figure 1: *Wskazówka: `matplotlib.pyplot.plot`*