Zestaw 1

1. Narysować odwrócony pusty trójkąt z gwiazdek o zadanej długości nieparzystej, np. dla 11 (każdy kolejny wiersz maleje o dwie gwiazdki) wyglądać ma:

```
**********
 *       *
  *     *
   *   *
    * *
     *
```

Parametr długości podstawy (tej u góry) wprowadzić poprzez input('jakis tekst')

2. Napisać program, który czyta podane jako zewnętrzne argumenty liczby naturalne, a następnie każdą rozkłada na czynniki pierwsze (co polega na zapisaniu dowolnej liczby naturalnej za pomocą iloczynu liczb pierwszych). Wymagany jest format wyjściowy w postaci a_1^k_1*a_2^k_2*…*a_3, jeśli k_i==1 to opuszczamy wykładnik potęgi. Przykładowo, jeśli wywołamy:

```
zadanie2.py  4407  13041599400
```

to powinno się wypisać (proszę tak to sformatować):

```
4407 = 3*13*113
13041599400 = 2^3*3^4*5^2*805037
```

Do wczytania zewnętrznych argumentów proponuję na początek coś bardzo podobnego do tego, co jest w języku C++, czyli użycie listy argumentów (bez używania getopt czy argparse):

```
import sys # importujemy modul

argv = sys.argv[1:] # argv to lista, a 1: robi selekcje bez pierwszego argumentu – nazwy programu

for i in range(1,len(argv)): # za pomoca generatora
          print(int(sys.argv[i])) # wpisujemy, rzutowanie z typu str na int przyda sie potem
```

Opis i program w C++ https://www.algorytm.edu.pl/algorytmy-maturalne/rozklad-na-czynniki.html
Pomocny może też być kalkulator rozkładu na czynniki pierwsze https://www.liczebnik.pl/czynniki-pierwsze.php

3. Napisać program rysujący "miarkę" o zadanej długości. Należy prawidłowo obsłużyć liczby składające się z kilku cyfr (ostatnia cyfra liczby ma znajdować się pod znakiem kreski pionowej). Należy zbudować pełny string, a potem go wypisać. [Zad. 3.5 https://ufkapano.github.io/algorytmy/lekcja03/zadania.html]

```
|....|....|....|....|....|....|....|....|....|....|....|....|
0    1    2    3    4    5    6    7    8    9   10   11   12
```

4. Napisać program rysujący prostokąt zbudowany z małych kratek. Należy zbudować pełny string, a potem go wypisać. Przykładowy prostokąt składający się 2x4 pól ma postać:

```
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
```

[Zad. 3.6 https://ufkapano.github.io/algorytmy/lekcja03/zadania.html]

5. Dla dwóch sekwencji znaleźć: (a) listę elementów występujących jednocześnie w obu sekwencjach (część wspólną, bez powtórzeń), (b) listę wszystkich elementów z obu sekwencji (sumę, bez powtórzeń). Zademonstrować na przykładach. P.s. można doczytać sobie o typie set [Zad. 3.8 https://ufkapano.github.io/algorytmy/lekcja03/zadania.html]