

Zadanie 1 - rozkład na czynniki pierwsze.

Zaimplementuj funkcję $czp(p)$ która zwróci listę czynników pierwszych zadanej liczby naturalnej n . Zrób to rekurencyjnie sprawdzając podzielność liczby przez kolejne liczby naturalne (aż do $\sqrt[n]{n}$) – rekurencja pojawia się, gdy liczba jest podzielna – wtedy uruchamiamy algorytm na jej dzielnikach, o ile nie są pierwsze.

```
In [2]: from math import sqrt
import numpy as np
```

```
In [3]: def czp(p):
    devidors = []
    if p == 1:
        return [1]
    else:
        devidors.append(1)

    for n in range(2, int(sqrt(p))+1):

        if p % n == 0:
            devidors.append(n)
            devs = czp(n)

            for i in devs:
                if i not in devidors:
                    devidors.append(i)

    return devidors

print(czp(6))
```

[1, 2]

```
In [7]: def czp(number):
    res = []
    start = 2 #szukam dzielnika, zaczynam od 2
    while start < sqrt(number): #sprawdzam podzielność liczby przez kolejne liczby naturalne (aż do √n)
        if number % start == 0: #warunek sprawdzenia podzielności
            return czp(number // start) + czp(start) # jak znajdujemyu jakis dzielnik to zwracamy dzielnik jego
        else:
            start += 1
    if not res: #sprawdzenie czy liczba jes pierwsza
        return [number]

czp(334)
```

Out[7]: [167, 2]

Zadanie 1.2 - *sito Eratostenesa*

```
In [ ]: def sera(p):
    if p <= 1:
        return
    x = np.ones(p-1)
    for n in range(int(sqrt(p))):
        if x[n] == 1:
            for j in range(int(sqrt(p)/2)):
```

In []: