Scientific Software Development in Python, AIMS Rwanda 2016
Assignment 1: arithmetic functions

For this first set of exercises you are asked to create a Jupyter notebook in which you will program several arithmetic functions. There are also questions for which the answer should appear in the notebook.

You are free to do the exercises in any order you like. However, each exercise should start with a Markdown cell with the number of the exercise.

Each function you program should come with

- some examples
- some tests that check that the result is consistent. For example if you have programed the `gcd` function you can have the following in a cell

```
for a in range(50):
    for b in range(50)
        assert gcd(a, b) == gcd(b, a)
        assert gcd(a + b, b) == gcd(a, b)
        for c in range(10):
            assert gcd(c*a, c*b) == c * gcd(a, b)
```

A lot of attention will be paid to the clarity of the code. In particular

- give meaningful name to your variables, for example `s` for a sum and `p` for a product, `counter` for a counter in a `range`, etc
- use comments (using `#`) to explain the delicate steps of your algorithms.

Exercise 1

Write a function `prod(l)` that returns the product of the elements in the list `l`. In case the list is empty, the function should return `1`.

For example

```
>>> prod([1, 3, 2, 3])
18
>>> prod([])
1
```

Exercise 2

Write a function `gcd(x, y)` that computes the greatest common divisor of two integers using Euclide algorithm.

Plot the graphic of the function $n \mapsto \#\{(p,q) : 1 \leq p \leq n,\ 1 \leq q \leq n,\ \gcd(p,q) = 1\}$.

Could you guess the asymptotic?

Recall that the *digits* of an integer n in base b is the sequence of numbers $(n_0, n_1, \ldots, n_d)$ in $\{0, 1, \ldots, b-1\}$ so that

$$n = \sum_{i=0}^{d} n_i b^i.$$

For example, the digits of 12 in base 10 are $(2, 1)$ and in base 2 are $(0, 1, 1)$.

Exercise 3

Write a function `digits(n, b)` that return the list of digits of the number `n` in base `b`.

Let $n = 123576537645123412$ written in base 10. What are its digits in base 2? In base 3?

What is the sum of the digits of 2^{100} written in base 3?

One can also consider the digits of fractions. For example, in base 10 one can write

$$\frac{1}{2} = 0.5 \qquad \frac{1}{3} = 0.(3) \qquad \frac{27}{28} = 0.96(428571)$$

where the part in paranthesis should be repeated periodically.

Exercise 4

Write a function `fraction_digits(p, q, b)` that assumes that $p < q$ and returns a pair of lists `[l1, l2]` where:

- `l1` is the list of digits that constitutes the preperiodic part of the expansion of $\frac{p}{q}$,
- `l2` is the list of digits that constitutes the periodic part of the digits of $\frac{p}{q}$.

For example

```
>>> fraction_digits(1, 2, 10)
[[5], []]
>>> fraction_digits(1, 3, 10)
[[], [3]]
>>> fraction_digits(27, 28, 10)
[[9, 6], [4, 2, 8, 5, 7, 1]]
```

Exercise 5

Write a function `prime_range(n)` that returns the list of prime numbers less than `n`. (*hint: use the sieve of Eratosthenes*)

Use your function `prime_range` to answer the following questions

- How many primes are less than 1243? less than 254321?
- How many primes are there between 43123 and 122505?
- Make a graphic of the function $n \mapsto \#\{p : p \leq n \text{ and } p \text{ is prime}\}$. Could you guess the asymptotic of this function?

We recall that the set of integers is a unique factorization domain (UFD): any non-zero integer can be uniquely written as a product $n = s\ p_0^{k_0}\ p_1^{k_1}\ \ldots\ p_m^{k_m}$, s is the sign (ie either $+1$ or -1), p_i are primes and k_i are positive integers.

Exercise 6

Write a function `factor(n)` that returns the factorisation of a positive number `n` as a list of pairs `[[p0, k0], [p1, k1], ..., [pm, km]]` where the `pi` and `ki` are respectively the prime numbers and exponents.

For example

```
>>> factor(12)
[[2, 2], [3, 1]]
>>> factor(19)
[[19, 1]]
```

What are the factorisation of

- 533850245821893?
- $2^{97} + 1$?
- $2^{97} - 1$?

Exercise 7

A positive number n is k-th power free if there is no integer m greater than 1 so that m^k divides n.

- Write a function `is_kth_power_free(n)` that tests whether the number n is k-th power free.
- Make the list of squarefree (ie 2-nd power free) numbers less than 100.
- Program the function `moebius_mu(n)` that is the Möbius μ function

$$n \mapsto \begin{cases} 0 & \text{if } n \text{ is not squarefree} \\ (-1)^k & \text{if } n \text{ is a product of } k \text{ distinct prime numbers} \end{cases}$$

- Find the smallest integer $n \geq 2$ so that $\mu(n) = 1$, $\mu(n+1) = -1$ and $\mu(n+2) = 1$.

Exercise 8

Write a function `sigma(n, k)` that return the sum of the k-th power of the divisors of n. For example $\sigma(n, 0)$ is the number of divisors of n and $\sigma(n, 1)$ is the sum of divisors. (*hint: you should use the function* `factor`)

Could you write a more efficient function `num_divisors(n)` that is equivalent to `sigma(n, 0)`?

What is the number less than 1000 that has the largest number of divisors?

What is the number less than 1000 that has the largest sum of divisors?