

# Exercises week 1: arithmetic functions

For this first sheet of exercises you are asked to create a Jupyter notebook in which you will program several arithmetic functions.

## 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 `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?

## Exercise 3

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

Use the 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?

## Exercise 4

Write a function `factor(n)` that returns the factorisation of `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 5

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?