Scientific Programming
(Wissenschaftliches Programmieren)

Exercise 1

1. Palindrome

- Create a Python-script which checks whether a given word is a palindrome.
- The script should read a phrase as input and print a message whether the phrase is a palindrome or not.

2. Fibonacci numbers (#1)

- Create a Python-script which generates Fibonacci numbers.
- The script should read the number of desired Fibonacci numbers as input and calculate (and print) those Fibonacci numbers. (The printed numbers should be aligned to the right with a field width of 10 characters.)
- Make sure the script also works correctly, when the requested number of Fibonacci terms is only one or two.

Hint: The first two elements of a Fibonacci series are 1, all other elements are the sum of the previous two elements: 1, 1, 2, 3, 5, 8, 13, ...

3. Fibonacci numbers (#2)

- Create a Python-script which generates Fibonacci numbers.
- The script should read the maximal value of the numbers to produce, and print all Fibonacci numbers which are less than or equal to this maximal value.

4. Prime checking

- Create a Python-script which checks whether a given number is a prime number.
- It should print a message telling whether the number is a prime or not.
- Make sure, the script also works correctly when checking the numbers 0 or 1.

Hint: The script should check for the necessary range of numbers whether they are divisor of the input number or not. (Try to find the minimal range of numbers to check!)

5. Prime factorization*

- Create a Python-script / IPython-notebook which factors an integer into primes.
- Given a positive number as input, the script should print the prime factors of the number (one prime factor per line) and how often that prime factor is contained. (e.g. for 8 it should print “2**3”; for 126 “2**1”, “3**2”, “7**1” in separate lines).