Python for physicists - exercise 2

Submission instructions - please read carefully:

• To be submitted by *** in the moodle (Lemida) system.

• *** files with py suffixes must be submitted - named exactly as detailed below for each exercise.

That is to say that:

• Do not submit complete projects, libraries, zip files, etc., and do not submit all exercises in one file, but in separate files with the names listed below.

• Make sure that the files run and do what is needed (on a recent version of Python, 3.5 or higher).

• Use only the commands we learned in the practice (for example: we have not yet learned while loops).

Exercise 1. Submit it as file name: py02-01.ex

The user must receive a positive integer representing a number of products, in the following way:

Please enter number of product:

Assume that the input is correct, i.e. that the user entered a positive whole number.

After that, product prices must be entered - whole and positive numbers, in the following way:

Please enter price of product:

Every time after a user enters a price, the message will be displayed again and another price will be recorded.

Number of repetitions is as the number of products the user entered in the first input.

At the end of the program, the highest price among all the prices, the lowest price among all prices, and the average price of all products must be printed.

Exercise 2. Submit it as file name: ex02-02.py

Take a natural number *n*. If it is even, divide it by 2: $n \to \frac{n}{2}$

If it is odd, multiply it by 3 and add 1: $n \to 3n + 1$

Repeat the process over and over again.

Koltz's conjecture says that no matter what number *n* we start with, we will inevitably end up with 1.

Write a program that receives from the user a natural number *n* and a natural number *m*) (assume that the input is correct), and checks whether within *m* repetitions of the above process you reach 1, when starting from the number n.

If so, print: *n* reached 1

Else, print: *n* did not reach 1 yet

Example: *m* = 10, *n* = 3.

$3 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$ so, we print: n reached 1

Another example: *m* = 10, *n* = 7.

$7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40 \to 20$

So, we print: *n* did not reach 1 yet.