Programmation avancée - Homework 2

Language : Python

Starts 12/10, Due 18/10, 23h59 (strict deadline, the submission must be validated before 23h59)

If any problem or something is not clear, contact Sylvain Lobry by email (@u-paris.fr).

Once you are done, send your exercises through moodle on the second "devoir". You can only send them once.

In this submission, you will have 4 (and only 4) files: Ex1.py, Ex2.py, Ex3.py Ex4.py. The ExX.py files should only contain the requested function (i.e. no other instruction, no outputs that are not explicitly asked, …) with the name being exactly the one given.

Failing to follow these instructions could lead to not being graded as exercises will be graded automatically (except for exercise 4 which is graded manually)

In each exercise, the modules which can be imported are given. Importing any other module will be considered as cheating.

Good luck!

Exercise 1: Palindrome

import: none

Write a function named "is_pal" that takes one string as parameter and returns True if this string is a palindrome, False otherwise.

Exercise 2: Prime numbers

import: none

Write a function named "is_prime" which takes a number as parameter and returns True if the number is prime, False otherwise.

Exercise 3: Ackermann function

import: none

Write a function named "ackermann" which takes two number as parameters, m and n and return the result of the Ackermann function. This function is defined as follow (from [Wikipedia](#)):

$$A(0, n) = n + 1$$
$$A(m, 0) = A(m - 1, 1)$$
$$A(m, n) = A(m - 1, A(m, n - 1))$$

Exercise 4: Histogram

Imports: none

For this exercise, you will create a function named "make_histo" which will print the histogram of a distribution of numbers. This function takes as input the name of a file containing a list of numbers sampled from the distribution and two optional parameters: nbin and height which both default to 20.

The file passed as argument contains a list of floating-point numbers sampled from a distribution, each separated by ", ". The last number is followed by a new line character. You can find examples [here](#).

The histogram that you will need to create will have nbin bins, and a height (i.e. number of characters on the y-axis) of height.

The bins are uniformly sampled among the minimum and the maximum of the distribution. On the last line of your program, you need to output the limits of the bins as integer (obtained by truncating the floating point value of the limit): therefore, you should print nbin + 1 integer on this line. As a consequence, this line starts with the minimum value found in the distribution and finishes with the maximum. There should be one integer every 3 characters: for instance, if a number is single digit, it should be followed by two spaces.

The histogram is drawn with the '*' character. For the first bin, you will first put a space, then '*' or a space (depending on the height of the first bin, then another space. This is repeated for every bin.

Outputs of this function with the default values for the two small distributions are given in the folder mentioned earlier.