Coding in Action Lab I
Python-04

1 Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for all your exercises.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded.
- These exercises are carefully laid out by order of difficulty - from the easiest to the hardest. We suggest you to solve them in the given order.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If your program contains a syntax error, you'll get 0.
- You cannot leave any additional file in your directory other than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called Google / man / the Web / ....
- Examine the examples thoroughly. Your code must *exactly* reproduce the examples, for the specified test cases. Beware that the examples could very well call for details that are not explicitly mentioned in the problem statement.
- By Odin, by Thor ! Use your brain !!!

Exercise 00: `print_words.py`

Create a program that reads the file "words.txt", asks the user to insert an integer n and prints only the words in the file with more than n characters, in alphabetical order. You can assume that the file has a single word per line.

Turn-in directory:	`ex00/`
Files to turn in:	`print_words.py`
Allowed functions:	`len, open, print, input`

Examples:

```
42~ > python print_words.py
Insert an integer: 19
The words longer than 19 are the following:
counterdemonstration
counterdemonstrations
counterdemonstrators
hyperaggressivenesses
hypersensitivenesses
microminiaturization
microminiaturizations
representativenesses
42~ >

42~ > python print_words.py
Insert an integer: 20
The words longer than 20 are the following:
counterdemonstrations
hyperaggressivenesses
microminiaturizations
42~ >

42~ > python print_words.py
Insert an integer: 21
The words longer than 21 are the following:
42~ >
```

Exercise 01: `write_words.py`

Create a program that does the same as Exercise 00, but instead of printing it writes on the file "long_words.txt".

Turn-in directory:	`ex01/`
Files to turn in:	`write_words.py`
Allowed functions:	`len, open, print, input`

Example:

```
42~ > python write_words.py
Insert an integer: 20
The words longer than 20 have been written on "long_words.txt"
42~ > cat long_words.txt
counterdemonstrations
hyperaggressivenesses
microminiaturizations
42~ >
```

Exercise 02: `print_words.py` (v2)

Create a program that does the same as Exercise 00, but asking to the user the name of the file to read from. The program must use a "try-except" statement to handle the case in which the specified file does not exist.

Turn-in directory:	`ex02/`
Files to turn in:	`print_words.py`
Allowed functions:	`len, open, print, input`

Examples:

```
42~ > python print_words.py
Insert file name: words.txt
Insert an integer: 20
The words longer than 20 are the following:
counterdemonstrations
hyperaggressivenesses
microminiaturizations
42~ >

42~ > python print_words.py
Insert file name: wrds.txt
Error! The specified file does not exist!
42~ >
```

Exercise 03: `dump_word_dict.py`

Create a program that constructs a dictionary whose keys are integers and such that the value associated to n is the number of words in "words.txt" whose length is n. Finally, the program "dumps" the dictionary onto a binary file "word_count.pickle" using the `pickle` module.

Turn-in directory:	`ex03/`
Files to turn in:	`dump_word_dict.py`
Allowed functions:	`len, open`

Example:

```
42~ > python dump_word_dict.py
42~ > python -m pickle word_count.pickle
{2: 85,
 3: 908,
 4: 3682,
 5: 8254,
 6: 14371,
 7: 21721,
 8: 26442,
 9: 16656,
 10: 9199,
 11: 5295,
 12: 3165,
 13: 1960,
 14: 1023,
 15: 557,
 16: 261,
 17: 132,
 18: 48,
 19: 16,
 20: 5,
 21: 3}
42~ >
```

Submission and peer-evaluation

Turn in your assignment in your Git repository as usual. Only the work inside your repository will be evaluated during the defense. Do not hesitate to double check the names of your files to ensure they are correct.