

## **Exercises 6**

**31.03.2014**

**Rules:** The document contains a set of 4 exercises: exercises 1 to 3 worth 2 points, exercise 4 worth 4 points. You need to provide for each one a Python script. All files must be put in a ZIP archive named `FirstName_LastName_exn.zip`, where `n` is the number of the exercise session (see `ex_set_1.pdf`). The files must be named `exercise_m.py`, where `m` is the number of the exercise. The ZIP archive must be uploaded on ILIAS until the specified deadline.

Good luck!

**Exercise 1.** Write a Python script that creates a dictionary of words from the input text found in the `input_ex1.txt` file. The dictionary should contain only the 4-letter words in the text (attention: **not** 4 characters, but only letters `[a-zA-Z]`). Output the dictionary to an output file.

**Exercise 2.** Write a Python script that extracts every word in the dictionary created in Exercise 1 and checks if by changing a letter a new valid word is obtained. The check will be done against the list in the `english_dictionary.txt` file. If the new word is valid, it will be added to the created dictionary. This way, your dictionary will expand with new words very close to the ones already in it.

**Exercise 3.** Using the dictionary created in Exercise 1, write a Python script that performs the spellcheck of a given word (the word is received as a *command line parameter* when the script is executed). The spellchecker will try to find if the wrongly spelled word belongs to the dictionary in one of the following 2 cases:

- a) one letter in the word is missing
- b) 2 letters in the word are interchanged

**Exercise 4.** Implement in Python the SOUNDEX algorithm and Damerau-Levenshtein distance algorithm. The Damerau-Levenshtein distance has to also take into account adjacent transpositions. You may use the dictionary created in Exercise 1 for implementing/testing. Choose 5 words spelled incorrectly and give 5 more variants for each having the same SOUNDEX code and order them by their Damerau-Levenshtein distance.

**Note.** The SOUNDEX algorithm is nicely explained [here](#).