Exercises 5
24.03.2014

Rules: The document contains a set of 3 exercises, first worth 2 points, the other two worth 4 points each. You need to provide for each one a Python script. All files must be put in a ZIP archive named FirstName_LastName_ex*n*.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 implements the **Fowler–Noll–Vo** (FNV-1) hash function.
Note. You can find here the pseudocode and some indications on the computation elements.
Attention! It is **not** allowed to use Python modules/functions that already implement this hashing algorithm. It is **not** allowed to use code posted on the internet that already implements this hash function in Python.

Exercise 2. Implement in Python a dictionary for strings using the hash function defined in Exercise 1. The collisions will be handled with linked lists. The dictionary must support at least the following operations:

- insert(string): computes the hash of the string and adds it to the dictionary. If the string already exists, throws an Exception
- remove(string): looks for the string by its hash value and removes it from the dictionary. If the string is not in the dictionary yet, throws an Exception
- lookUp(string): returns the string if in the dictionary, NULL otherwise
- size(): returns the number of slots used (hash values in the dictionary)
- capacity(): returns the total capacity of the dictionary

Note. It would be best if the constructor of the dictionary got as a parameter the initial capacity. Once half of the dictionary is filled, you double the capacity.

Exercise 3. Write a Python script that parses input_ex3.txt file and adds the words to the dictionary defined in Exercise 2. The script should return in an output file the total number of collisions, the maximum number of collisions per word, the final size and the final capacity of the dictionary. We consider as “words” the strings having at least 3 characters and all of them in the interval [a-zA-Z].