Project I2206 – Dictionary

Project objective :

This project aims to use the skills acquired in the "*Data structures*" and "*Imperative programming 2*" courses in an application framework simulating a dictionary

Organization

Students must work in teams of 2 students.
Students in the same group should submit their names **before Friday 28/05/2021**.
At the end of the project, the following elements must be returned:

- The full source program (well commented).
- A tested and operational executable.
- A small report presenting the work carried out:
    1. Program organization: role of each function, explanation of the program code.
    2. User manual: how to use the program?
    3. Tasks completed as well as others that you could not do.
    4. Difficulties encountered, etc.

Project evaluation

The project evaluation is based on the following elements:
1. The source program:
    - Respect for the project requirements.
    - Technical quality of the program: breakdown into functions and modules, instructions, algorithms, etc.
    - Efficiency, error management ...
    - Presentation of the program: indentation, comments and naming.
2. The provided documentation:
    - Organization of the program and how to use it.
    - Assessment of the defense of the work.
    - Demonstration of the program by the group.
    - Individual questioning on the work carried out, if necessary.

3. Cheating in any form/ways will be punished in a ZERO grade

Project description

In this project, you have to represent a dictionary by a binary search tree. Each node contains, no longer a simple integer, but a pair formed of a string, representing a word from the dictionary, and a list of strings, each element of the list representing a synonym of the word.

Required work

1. Implement the appropriate ADT (Abstract Data Type) to represent a dictionary.
2. Load the dictionary (given as a file) into memory.
3. Test if a word is present in the dictionary.
4. Given a word found in the dictionary, print all the synonyms for that word.

5. Given a word m and one of its synonyms s, add to the dictionary the synonym s of the word m (if it is already present, add a synonym among its synonyms, if it is not present, then add it to dictionary, with the synonym s as the only synonym).
6. You must also suggest other operations such as:
    a. Find the word with the highest number of synonyms
    b. Given a word m, find the dictionary words for which m is a synonym.
    c. Find the most repeated word as a synonym in the dictionary
    d. Balance the dictionary to improve complexity.
7. Given a dictionary represented as a binary search tree, store the dictionary in a file, further ensuring that the words are arranged in alphabetical order.