Université Antonine

Faculty of Engineering
Computer and Communication Engineering Department
Academic Year – 2023/2024

Data Structures Project

The main objective of this project is to create a dictionary library in data structures using linked lists. The application will allow users to search for words that start with the same set of letters as well as to search for synonyms and antonyms. In addition, the users will be able to add more words to the dictionary by first checking for duplication before inserting new words. Moreover, users would have the option to remove or modify some words/synonyms/antonyms in the dictionary.

For this purpose, and in order to achieve the project, you can use the structures defined in Table 1. Please note that it is strictly forbidden to use arrays or any other already available containers in this project except for the arrays of characters.

Table 1 – Data structures to be used in the project

```
struct sWord {
	char data[30];
	sWord *next;
};
```

```
struct dWord {
	char data[30];
	sWord *synonym, *antonym;
	dWord *next, *previous;
};
```

```
struct Dictionary {
	dWord *head, *tail;
};
```

For this purpose, you will make use of the data structures defined in Table 1 where:

- The structure **Dictionary** is a doubly linked list of words present in the dictionary.
- The structure **dWord** is a data structure representing a word in the dictionary. It contains two pointers ‘next’ and ‘previous’ that are required for the proper operation of the doubly linked list **Dictionary**. It also contains two singly linked lists ‘synonym’ and ‘antonym’ that list all the synonyms and antonyms of the word.
- The structure **sWord** is a data structure representing a word that is used as a synonym or antonym. It contains the pointer ‘next’ that is required for the proper operation of the singly-linked lists ‘synonym’ and ‘antonym’.

Your task is to write a program and test it on an input text file containing already some words from the dictionary. The format of the input text file is as follows where the synonyms are separated by ‘:’ while the antonyms are separated by ‘#’. Please mind that if ‘big’ is a synonym of ‘large’, ‘large’ is also a synonym of ‘big’

```
big:large:great:huge#small#tiny#little
difficult:hard#easy
new:recent#used
```

```
clean:tidy#dirty
safe:secure#dangerous
friendly:outgoing#unfriendly
good:great#bad
cheap:inexpensive#expensive
interesting:fascinating#boring
quiet:still#noisy
```

The program should also contain at least the following functions:

1. A function that parses the input text file and returns the doubly linked list of all the words in your input text file.
2. A function that takes the doubly linked list of all the words in your dictionary and writes it back to a text file while making sure that a word is not written more than once. The output file has the same format as the input file.
3. A function that takes as an argument the dictionary and sorts it in alphabetical order by changing the links between the nodes.
4. A function that takes as an argument the dictionary and a new word to be added. You need first to check if the word does not already exist in the dictionary before inserting it in a way to keep the dictionary sorted.
5. A function that takes as an argument the dictionary, an existing word, and a new word to be added as a synonym/antonym. This function needs to add the new word as a synonym or antonym for an existing word in the dictionary. Please keep in mind that if word1 is a synonym for word2, word2 is also a synonym for word1, and they share the same antonyms.
6. A function having the dictionary and some letters as arguments and deletes all the words containing these letters in the dictionary.
7. A function that searches in the dictionary for all the words starting with a given set of letters.
8. A function that searches the dictionary for all the synonyms and antonyms of a given word.
9. A function to delete all the words in the dictionary that start with a given set of letters.
10. A main function that interacts with the user and allows him to make use of all the previous functions.

Instructions:

a) This project can be done by a group of one or two students. When working in a group of two students, the efficiency of the teamwork will also be evaluated.

b) You need to send an email to your instructor, by the 1st of December 2023, specifying whether you will be doing the project alone or in a group. In the latter case, you will need to provide the name of your teammate.

c) The deadline for submitting your project (source code and input file(s)) is by the 15th of December 2023 (midnight).

d) The dates for defending your projects will be specified later on by your instructors. During the project defense, you will briefly present your project, and you will be assigned a task that can be related to any part of the course.