

Spell Checker

In this lab we will implement a program to do spell checking of text documents against a dictionary of words. One straight-forward way to implement a fast spell checker is to use a hash table. If a word is found, it is assumed to be spelled correctly. Otherwise, it is assumed to be spelled incorrectly.

You are provided with a dictionary file that contains the root words.

Basic functional requirements:

- Spell checker loads the dictionary words from the given text file into hash table (memory). Each word in the file serves as a key as we insert entries into the hash table.
- Spell checker will be reading consecutive words from the document to be checked. Each word will be looked up in the dictionary. If a word is not found in the dictionary, the word is considered a spelling error.
- Spell checker will list the misspelled words in the document
- Spell checker gives the number of words that begins with a given character

Software design requirements:

- Use separate chaining
- Use Linked List as a bucket of the hash table. LinkedList header file is provided.

The basic interface of SpellChecker class:

- `SpellChecker(string fileName, int size=100){`
 - takes the dictionary file name and the capacity of hash table as input parameters. Constructor will load the file into the hash table.
- `checkSpelling("testfile.txt");`
 - will be reading consecutive words from the test file, and checks if each word is in the dictionary. If a words is not found in the dictionary, it will be printed as a misspelled word.
- `countWordsBeginsWith('a');`
 - returns the number of words that begins with the given character in the dictionary