Samuel Bohannon\

Programming Assignment Four

Due November 1st 2015

Description of the problem: In this program, a spell checker was implemented. A dictionary file in random order is to be read into an array of linked lists, then another text file containing a book is to be read one word at a time. These words are to be searched for in the dictionary, if they are not there, than the word is likely misspelled. The total number of words found and words not found is to be counted, along with the total number of comparisons for words found and words not found. These counters should be printed on the console.

Algorithm: First an array of 26 linked lists were instantiated using the My Linked List class in a previous assignment. The dictionary file is read, and each word is inserted into the proper linked list depending on the first letter of the words (a = 0, b = 1, etc). The book file is then read one word at a time. The dictionary, now split up into linked lists, is searched using each word in the book file as the key. Counters are maintained during this process. One for the number of comparisons, which uses an overloaded contains method in the MyLinkedList class to keep track, and the total number of words found and words not found. Finally, the averages are found and ouput.

Observations and results: This program is noticeably slower than the binary search version. The binary search spell checking program executes in around 14 second, this linked list version executes in around 44 seconds. The reason is because of the large difference in the number of comparisons. In this program there are around 7300 comparisons per word not found, and around 3500 comparisons per word found. This is quite a difference from the earlier spellchecker program, in which the average comparison per word found was around 15, and the average comparisons per word not found was 17.

Conclusion: This program has the advantage of being able to use a dictionary in random order and searching for words without sorting the data first, but it takes much longer than the binary search. This program was not too difficult to implement, it was comparable to the last spell checker program, and it used a lab which had been previously completed, so most of the work was already done.

Output:

```
Number of words found: 937492

Number of words not found: 54648

Average number of comparisons for words found: 3558.07

 Average number of comparisons for words not found: 7380.38
```