CECS 277 – Lab 9 - List Iterator

List of Words

Create a program that reads in a file "words.txt" and stores them into a LinkedList by inserting them in sorted order. Then it displays the list of sorted words.

Create the following functions:

1. readFile – construct a LinkedList and a ListIterator, read in the file and add each word to the LinkedList. Call moveIter to move the iterator to the correct position to add each word so the list is always in kept in sorted order. Return the list.
2. printList – pass in the list, create a ListIterator that starts at the beginning of the list, use the iterator to display the contents of the list.
3. moveIter – pass in a ListIterator and a String for the word. Move the iterator backward or forward from its current position to find the correct location to place the word so the list always stays in sorted order (note: use String's compareToIgnoreCase method so it sorts the words regardless of capitalization).

Notes:

1. Use the iterator to move through the list to add each word, one at a time, inserting them in sorted order. DO NOT add all the items and then just sort the list. The point of the assignment is to practice using iterators.
2. The iterator should be moving back and forth through the list. You should not reset the iterator to the front every time a new word is added from the file.
3. The moveIter method just moves the iterator to the right spot, it doesn't add.

Starting Out:

1. Start with a stripped-down version of the readFile function. Just read in the file and add the words directly to the end of the LinkedList.
2. Create the printForward function.
3. Create the main and display the unsorted words in the LinkedList.
4. Create the moveIter function. Compare the new word to the word at the iterator's position, then move the iterator forwards or backwards to the correct position.
5. Go back and update the readFile function to call moveIter to add the new word at that position when moveIter returns (display the words being compared as you add them, that way you can check that they're being added correctly).

Example Output:

```
a
Alice
Alice
and
and
bank
...
use
very
was
was
what
without
```