C201 Homework #9 Due 4/16/14

This assignment calls for enhancing the program that you wrote for homework #8. You are to add two new items to the menu used in homework #8, namely ReadList and SortList. The new menu should look like:

A(ddNode B(uildList D(eleteNode R(eadList S(ortList Z(apList Q(uit =>

The menu option "R(eadList", causes TestReadList to be called, which in turn calls ReadList, a function that reads characters from a file and stores them in the linked list. The function ReadList is documented below. **Note that ReadList should read all chars**, including newline characters until eof is reached. **Do not use AddNode in ReadList!**

```
/************************  ReadList  *********************************
 DESCRIPTION  Attempts to open the file named Filename, then read and
              store the characters from the file in a singly linked list with a dummy head node.
              The characters in the list are stored in the same order in which they are encountered
              when reading the file, i.e. the characters read last are at the end of the list.
              Input terminates when end of file is encountered.

 PARAMETERS

  IN-OUT: List   A pointer to a Node. List should point to the dummy head
                 node of a singly linked list ending in NULL.

  IN: Filename   A string that holds the name of the file to be opened
                 and read from.

  OUT: Error     A variable that is set to non-zero value, if an error
                 occurs for any of the following reasons:

                   1)  A file related error occurred.
                   2)  A memory allocation error occurred.

 NOTE          Before building the new list, Zaplist is called to delete
               the old nodes (except dummy node), preventing a memory leak.
-------------------------------------*/
```

The menu option "S(ortList"causes TestSortList to be called, which in turn causes SortList to be called. SortList rearranges the linked list so that it is in natural order, i.e. sorted. Below are the header comments that describe the SortList function.

```
/*****************************  SortList  ****************************
 DESCRIPTION  Arranges the singly linked list pointed to by List in
              natural order. It is assumed that the list has a dummy head node.

              The algorithm used is a linked variation of the selection
              sort and works like this:

              Start with EndSorted aimed at first node of List

              repeat
                Find smallest char between EndSorted and end of List
                Swap smallest element with char in EndSorted
                Change EndSorted to next node
              until we get to end of List

              Note: none of the pointers in the linked list are changed.

 PARAMETERS

   IN, List   A pointer to a singly linked list with a dummy head node.
-------------------------------------*/
```

* **Do *not* code the Bubble sort algorithm** -- read comments above!
* **Do *not*** re-test the routines that you finished when doing homwork #8, but yet you will have to build and zap a list to test the Sortlist.
* In testing ReadList, you should not read in large files. I suggest that you test it on the files, words.1, words.2, words.4, and words.5, that were used in homework #7 and earlier ones.
* Turn in only your output, ReadList function, and the SortList function (points off if you turn in more).
* Program shell can be found on my website as "a9shell.cpp"