CSCD 240

Problem Description
You are required to use a singular linked list to count word occurrence in a large text file, in which we have all types of punctuations. The node structure must use a void pointer for the data. You are required to use a linked list with the head reference declared within main. I don't care if you use a size or not. If you do, the size must also be local to main. You may only use a static array to read in a line from the file.

What is Required?
A node structure that only contains
void * data and struct node * next

The void * data will point to a structure of your choosing.

Two files are provided for processing, testfile1.txt and testfile2.txt. Testfile1 is the smaller file. Testfile2 is a big text file, which is a subset of Wikipedia and contains around half of million English words.

Basic Idea
We assume all words in the provided text are correct English words. Suppose we have a line of text, with a newline at the end.

You are a student. Who's your advisor? i.e. teacher. I'm your friends.

We can see that English words are delimited by white space or punctuations. Using strtok you will break the sentence into tokens. For the given text your tokens would be
You
are
a
student
Who
your
advisor
i
teacher
I
Your
friends

After we extracted a word, if its length is one, (contains only one letter), we throw it away, meaning, we will not insert a one-letter word into the linked list EXCEPT THE WORDS "I" or "A". NOTE: we will insert only one I or i into the list since we are case insensitive.

What you MUST do?

1) Create a generic linked list.h and linked list.c that contains the node structure described above. It must have:
    a. addFirst
    b. addLast
    c. addIndexed – index out of bounds do nothing – can do this without size
    d. addOrdered – function pointers help here
    e. removeFirst
    f. removeLast
    g. removeIndex – index out of bounds do nothing– can do this without size
    h. clear – cleans up all memory in the list
    i. print – prints the list
2) Read the file and insert unique words into the list testing all your functions
3) If the word is in the list already increment a counter for it. At the end you will printout the list and the number of times the word appeared in the file.
4) Create a make file with multiple targets. The main target will be named hw5
5) Provide an output file with the word count similar to below.
6) Provide a valgrind run proving you are leak free.
7) Formatting must match in the tabular format – don't care about the border

<u>To Turn In</u>

A single zip file containing:

- All your C/H files – you will want to keep your linked list files separate from other files
- Main will be named cscd240hw5.c
- All input files
- All output files
- You should really know by now what to turn in and the naming scheme

Sample Run

English Word	Count
a	2
Basically	1
calling	1
each	1
file	3
for	1
from	2
fscanf	1
function	1
I	1
in	1
my	1
scanning	1
string	1
strings	1
the	1
then	1
using	1