Homework 5

Due: December 4, 2015 **Points:**100

You ***must*** put your program in a file called “words.c”. It must also be a C program (not RTF, for example).

You are to write a program that reads one or more files, sorts the words in the files, and then prints each word prefixed by the number of times it appears in the files. For this assignment, a “word” is a maximal sequence of alphanumeric characters. So, for example, “hello”, ”abcdefgh”, “1245”, and “good4you” are all words. Any non-alphanumeric terminates a word. You can assume no word has more than 99 characters.

For example, if given the Declaration of Independence, your program should print the following:

```
1 1776
1 4
1 A
1 Absolved
2 Acts
1 Administration
1 Allegiance
1 Alliances
2 America
1 And
1 Annihilation
1 Appropriations
1 Arbitrary
2 Armies
1 Arms
1 Assembled
4 Assent
```

because all the above words appear once except for “Acts”, “America”, and “Armies”, each of which appears twice, and “Assent”, which appears four times.

Because you will not know how many different words the input files contain, you must use a linked list to hold the words. As you encounter each word in the text, you should add it to the linked list. It’s probably easiest if you do the insertion sort like the example in *linked.c*, but you can build the list and sort it afterwards. The structure for a word should also contain a field to hold a count of the number of times the word appears in the input files.

When you sort the words, sort them in ASCII order. You can use the function

```
int strcmp(char *, char *)
```

to do this. In this ordering, the digits come first, then the capital letters, then the lower-case letters.

Your program is to take 0 or more file names as arguments. If there are no files named, your program should exit with no output. If there are no errors (not naming any files is not an error), exit with exit code 0.

If a file that is named cannot be read, use the function *perror* to print the error message. If `fname` is the name of the file that cannot be opened, call the function like this:

```
perror(fname);
```

This function will print the name of the file followed by a colon and an error message stating why the file could not be opened. This is written on the standard error (not the standard output). Then continue until all named files have been processed, and exit with exit code 1.

If you are unable to allocate memory for a word, please print the error message:

```
%s, file %s, line %d:  ran out of memory\n
```

(the “\n” is a newline) with the first “%s” being replaced with the word, the second with the name of the file being processed when the error occurs, and the “%d” being replaced by the line number being processed when the error occurs. As above, exit with exit code 1.

The format for printing a word is

```
%5d␣%s\n
```

(the “␣” is a blank space and, again, the “\n” is a newline) where the “%d” is replaced by the count (the number of times the word appears in the input file(s) and “%s” is replaced by the word.