MATLAB Programming NYCU Fall 2022 LAB#7 2022/11/1

For this lab, you will modify the work done in the previous lab to get a little experience on MATLAB structures. Some parts in your function are similar to the previous lab, including:

- Reading the text file (use the same text file as the last lab).
- Extracting the individual words.
- Identifying the unique words and their counts.
- Sorting the words according to some criterion.

What are different from the previous lab:

The function header becomes

```
A = my_word_count(fn, sort_mode)
```

Here `fn` is the path to the file. The output `A` is an array of structures with three fields: `word` is the word itself (make it a string scalar or character vector), `len` is the length of the word, and `count` is its times of occurrence. Example:

`A(1).word` is `'she'`, `A(1).len` is 3, and `A(1).count` is `4`.

`A(2).word` is `'sells'`, `A(2).len` is 5, and `A(2).count` is `4`.

The second input `sort_mode` is a string scalar or character vector that indicates how the words should be sorted. It can be `'word+'` or `'word-'`, meaning sorting in dictionary order; `'len+'` or `'len-'`, meaning sorting by length; `'count+'` or `'count-'`, meaning sorting by count. Here the part of `'+'` or `'-'` in `sort_mode` indicates that the sorting is in the ascending or descending direction, respectively.

You can use `[A.field]` to put all the values of the same field in the structure array into a vector, and sort this vector. This applies to `[A.len]` and `[A.count]`. For sorting the words, the method depends on whether your words are strings or character vectors. If they are strings, you can directly use `[A.word]` for sorting. If they are character vectors, you need to use `{A.word}` to put all the words into a cell array of character vectors, and they apply `sort` to this cell array.

Finally, if you call the function with no output argument, let the function print out the words and their lengths and counts in the sorted order.