## MATLAB Programming NCTU Fall 2017 LAB#6 2017/11/6

For this lab, you will try some string processing in MATLAB, as well as a bit of file processing. You should write a function that takes a path to a text file as its only input. The function should find the individual English words in the string, and determine the unique English words and count the time of occurrence for each unique word. There are two output arguments: One is a cell array of strings containing the unique words, and the other is a vector containing their times of occurrence.

For this lab, you can use any string processing function provided by MATLAB.

First, use a text editor to save the following text (which is an English tongue-twister) to a text file:

```
She sells seashells by the seashore.
The shells she sells are surely seashells.
So if she sells shells on the seashore,
I am sure she sells seashore shells.
```

The unique words you find should be case-insensitive, that is, 'She' and 'she' are the same word. Punctuations (標點符號) should be excluded.

You function header should look like

```
[word, count] = my_word_count(fn)
```

Here **fn** is the path to the file.

The output **word** should be like

```
'she', 'sells', 'seashells', 'by', …
```

The output **count** should be like

```
4 4 2 1 …
```

The ordering of the words can be different from the example here.

Finally, if you call the function without an output argument, let the function print out the words and their counts. Let the output be like the following:

```
she           4
sells         4
seashells     2
by            1
...
```

There are two approaches that you can use to get the list of unique words:

(1) Maintain the list of unique words as you read the file. Keep the unique words in a cell array of strings, and compare each new word to the strings in this cell array. The counts are updated as well.

(2) Read all the words into a cell array of strings without checking for duplicates. Once you have collected all the words, call **unique** to get a list of unique words. The word counts are calculated afterward.

You can try just one or both.

For your thoughts: Which will work better, and why? (Note: **unique** has the same time complexity as **sort**.)