ISIT307 Assignment 2

To be started in: week 3

Due: by the date specified for the assignment 2 drobox in Moodle.

Description

The purpose of this assignment is to practise defining functions and string processing. You will write several PHP functions that support keyword auto-complete.

You may use a form that supports entry of a partial keyword (but you are not required to). The key requirement is to write a PHP function that supports the auto-complete operation.

Your program will contain a list of keywords that users have searched for before (in an array). And a second list of search counts (also in an array). The idea is: if the user types "ph" and many users have previously typed "php" your program would offer "php". Like Google auto-complete your program should offer multiple choices, with the most likely one at the top. So your auto-complete function will need to return a list (array) of possible matches in an appropriate order.

Note each completed search adds an additional entry into the list of prior searches if it is the first time for that search term, or updates the list of search counts, if it is not the first time. Your test code will include a sequence of keyword searches that demonstrate this behaviour.

Example lists:

```
searchTerms = {'Wollongong hospital', 'Wollongong weather', 'Sydney traffic' }
searchCounts = { 1, 10, 2 }
```

so there have been 13 prior searches, of which 10 were "Wollongong weather".

Try to make this auto-complete function relate to your web application scenario.

Marking Scheme

Your assignment must be uploaded into the corresponding Moodle dropbox. Your submission will be two files: (i) a PDF file that contains your code and a testing report that demonstrates a number of use cases (ii) a zip file of all your code. Ensure your code is readable i.e. indented properly in both the PDF and text code files. Use a fixed width font in the PDF. Change tabs to 3-4 spaces if needed. Code that is not easy to read will receive a lower mark.

A weak submission (some functions have been implemented, but it does not execute) will receive 1 mark. A reasonable submission (most functions have been implemented and integrated with your web application, but it runs with some problems) will receive 2 marks. A correct submission will receive 3 marks providing it is marked in the correct time.