The LoL (List of Lists) Lab

This week we will look at solving more complex problems by using list comprehensions to prepare information in such a way that we can make use of existing functions as a powerful part of our solution.

Finding an Extremal Value

Consider the following function (provided in `lab5pr1.py`). It is comparable to the lecture example that finds a maximum Scribble™ score from a list of words.

```
def longest(lst):
    """ input: L, a list of strings
        returns one of the strings
        with the greatest length
    """
    LoL = [ [len(s), s] for s in lst ]
    bestpair = max(LoL)
    return bestpair[1]
```

It uses a list comprehension to create a list of 2-item lists, `LoL`, with each sub-list composed of the length of a string together with the string itself, `[len(s), s]`. It then uses the Python `max()` function to find a 2-item pair with the largest length, `max(LoL)`. The string itself, the second item in the maximal 2-item list, is returned.

For example:

```
In [1]: W = ['once', 'upon', 'a', 'time']

In [2]: longest(W)
Out[2]: 'upon'
```

Make sure that you understand how this function works. You should be able to describe what the intermediate value `bestpair` would look like for a given input list, i.e. what kind of value is it and what would it contain. For example, the function returns `bestpair[1]`, but what would the value in `bestpair[0]` represent?

Add several printed test cases to your program to show how `longest()` behaves with various lists of strings.

With the function `longest()`, if more than one string is the same length as the longest, it only returns one of these strings, determined by the *lexicographical*, (alphabetical), order of the strings.

Write a new function, all_`longest()`, that has the same input as `longest()`, (a list of strings), but returns instead a list of all of the strings that are the same length as the longest one.

```
In [1]: words = ['once', 'upon', 'a', 'time']

In [2]: all_longest(words)
Out[1]: ['once', 'upon', 'time']
```

There are many possible ways to design this function but you may find it simplest to use more than one list comprehension, first to create the list of 2-item pairs and then another to create a list to return, containing all the suitable strings.

Submitting your work

Once you have completed the new function, and provided suitable printed tests, you are ready to submit your work.

Check that your file is named correctly, `lab5pr1.py`, and that both you and your partner's names are included in the header comment.

Upload your file to the Lab 5 page on Moodle. Make sure that both partners will have access to the work you did today for when you each want to review the lab material.

Have one of the TAs in the lab come over and briefly evaluate your work.