CSCA08 Exercise 4

Due: October 20, 2013. 5:00pm

The functions in this exercise are designed to be solvable in a number of ways. Just getting them working shouldn't be much of a challenge, but exploring different ways of solving the same problem will be fun. As always, all of your functions should be in a single file (`ex4.py`), and that file should not use `print`, `input` or `import`.

insert

Write a function called `insert` that takes 3 parameters, `listA`, `listB`[1] and an index, then returns a copy of `listA` with the elements of `listB` inserted at the index. Your code should work with both strings and lists (remember that strings are just a special form of list) e.g.,

```
>>> insert([1, 2, 3], ['a', 'b', 'c'], 2)
[1, 2, 'a', 'b', 'c', 3]
```

Challenge: Complete this function both with and without loops

up_to_first

Write a function called `up_to_first` that takes two parameters, a list (or string) and an object, and returns a copy of the list up to (but not including) the first occurrence of that object, or all of the elements if that object is not in the list. e.g.,

```
>>> up_to_first([1, 2, 3, 4], 3)
[1, 2]
```

Challenge: Complete this function using list slicing (without a loop)

cut_list

Write a function called `cut_list` that "cuts" a list. That is, given a list and an index, returns a copy of the list, but with the items before and after the index swapped. As always, your code should also work with strings. e.g.,

```
>>> cut_list([0,1,2,3,4,5,6,7,8,9], 3)
[4, 5, 6, 7, 8, 9, 3, 0, 1, 2]
>>> cut_list("ABCDEFGX1234",7)
'1234XABCDEFG'
```

Challenge: Complete the function in a single line of code (you may have to either use short variable names, or go slightly beyond 80 characters for this)

Submission

For each function, you only need to submit a single version. Make sure the versions that you do submit work and are PEP-8 compliant. If you wish, you may submit another file called `challenges.py` with alternate versions of your code, but only your `ex4.py` file will be marked.

[1] You should probably come up with better names than these