CSCA48 Exercise 10

Due: March 31, 2017. 5:00pm

Full Circle

In ex2, we asked you to play the banana game. Now, with our final exercise, we're going to code it.

Your job is to write a function called banana_game, that takes three parameters (s1, s2, c), s1 and s2 are strings, and c is a Container. Your function will return True iff you can turn s1 into s2 using c given the rules provided in the ex2 handout.

A Container is an ADT that has five methods, put, which adds an item to the container[1], get which removes and returns the next item in the container, peek which returns the value of the next item that will be returned by get but doesn't actually modify the container[2], is_empty, with returns True iff the container is empty, and copy() which returns a copy of the container, complete with all the elements it contains. The Container Class (and all related Exception Classes) will be in a file called container.py in the same directory as your code. i.e., you probably want to start your code with the line:

```
from container import *
```

The idea here is that your code should work whether the Container is implemented as a queue, stack, bucket, or any other data structure. This is a good exercise in both recursion (you definitely want to be solving this recursively), and in working with ADTs. Also... it's kind of fun to say "Banana game".

[1] if the container is full, as with a bucket, put will raise a ContainerFullException

[2] if the container is empty, get and peek will raise a ContainerEmptyException