CSCA48 Exercise 2

Due: January 27, 2017. 5:00pm

Last week, we had you play the **banana game**. Several of you came to us and asked us how we were going to mark it. Well, rather than just tell you, we thought we'd show you. Or rather, we thought we'd let you show us. Your job this week is going to be to write a function that verifies solutions for the banana game. The interesting thing is that we'll be able to do it for any type of container without a lot of extra work (thanks to the power of ADTs).

banana_verify

Write a function called `banana_verify`, that takes 4 parameters:

- A source word (str)
- A goal word (str)
- A container (Container)
- A list of moves (list of str)

The source and goal words are the source and goal words for the banana game. The list of moves is a list of moves, with `M` meaning `move`, `P` meaning `put`, and `G` meaning `get`. So the `CAT` example from the exercise 2 handout would have the moves list:

```
['P', 'M', 'G', 'M']
```

A `Container` is an ADT that has four methods: `put(item)` adds an item to the container[1], `get()` removes and returns the next item in the container, `peek()` returns the value of the next item that will be returned by `get` but doesn't actually modify the container[2], and `is_empty()` returns `True` iff the container is empty.

You won't know how the `container` object is implemented, and you shouldn't care. It could be a stack, or a queue, or a bucket, or something else entirely. Your job is to test whether the moves in the moves list, preformed using the specified container will turn the source word into the goal word, in which case, your function will return True. Otherwise it should return False.

Your code must start with the line:

```
from container import *
```

we will provide the container class (you'll probably want to write your own for testing purposes), as well as the necessary exception classes.

[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`