This lab gives you practice (1) reading and augmenting someone else's code and (2) putting together many of the little pieces we've seen so far this semester. It also serves in part as a lead-in to Homework Assignment #3, which will be appearing on the course site soon. *Please do not hesitate to ask questions if you don't understand the existing code or what is being asked.*

Before beginning, grab a copy of the file `lab12.py`. This file contains the basic outline of a simple word-guessing game, similar to Hangman (without the pictures) or television's *Wheel of Fortune* (without the money and prizes). The idea is that the program will "think" of a secret phrase, and the user will try to discover the phrase by guessing one letter at a time.

For this lab, you will be filling in the details of some of the functions. (For the homework, you'll be augmenting the implementation in various ways.)

Important: Do not make any changes to the `main` function in this lab. Also, do not remove any of my comments.

Your Tasks

1. Complete the definition of the function `stringify`, which should convert a list of characters into a string. For example, `stringify (['a','b','-','a'])` should return `'ab-a'`.

2. The function `getGuess ()` currently prompts the user for input and returns the first character of whatever the user enters.

   However, if the user simply hits Enter/Return, then an indexing error occurs (try it out to see what happens).

   Add code to make this function more robust, as follows: if the user enters only Enter/Return, the function repeatedly (as many times as necessary) prompts the user for a proper character and reads in their response. Hint: use a `while` loop!

3. The function `update` is intended to update a list (representing the current status of how much the user has correctly guessed) with the result of the user's most recent guess.

   For example, if `mylist` is initially the list `['a','-','-','-','-']`, the effect of the function call `update (mylist,"abcdc",'c')` is that `mylist` is now `['a','-','c','-','c']`.

   Add the necessary code (and remove `pass`) in `update` to produce this desired effect.

4. The function `compare` takes a list of characters and a string, and determines whether they contain the same sequence of characters; it returns `True` or `False` accordingly. (You may assume that each element of the list is a single character: your code does not need to handle other situations.)

   For example, `compare (['a','b','c','c'],"abcc")` returns `True`, but `compare (['a','b','c','c'],"abc")` and `compare (['a','b','c','c'],"acbc")` both return `False`.

What and How to Submit

Submit your code through Blackboard. In addition, you should hand in the a printout of your code (`lab12.py`).