**Printing out the User's guess**

Next, implement the function getGuessedWord that takes in two parameters - a string, secretWord, and a list of letters, lettersGuessed. This function returns a string that is comprised of letters and underscores, based on what letters in lettersGuessed are in secretWord. This shouldn't be too different from isWordGuessed!

Example Usage:

>>> secretWord = 'apple'

>>> lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']

>>> print getGuessedWord(secretWord, lettersGuessed)

'\_ pp\_ e'

When inserting underscores into your string, it's a good idea to add at least a space after each one, so it's clear to the user how many unguessed letters are left in the string (compare the readability of \_\_\_\_ with \_ \_ \_ \_ ). This is called *usability* - it's very important, when programming, to consider the usability of your program. If users find your program difficult to understand or operate, they won't use it!

For this problem, you are free to use spacing in any way you wish - our grader will only check that the letters and underscores are in the proper order; it will not look at spacing. We do encourage you to think about usability when designing

For this function, you may assume that all the letters in secretWord and lettersGuessed are lowercase.

def getGuessedWord(secretWord, lettersGuessed):  
    '''  
   secretWord: string, the word the user is guessing  
   lettersGuessed: list, what letters have been guessed so far

  returns: string, comprised of letters and underscores that represents

  what letters in secretWord have been guessed so far.

  '''