Problem 2

1 point possible (ungraded)

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.
    '''
    # FILL IN YOUR CODE HERE...

```

Press ESC then TAB or click outside of the code editor to exit

Unanswered

Submit You have used 0 of 30 attempts

PSET3: Problem 2

Topic: Sandbox / PSET3: Problem 2

Hide Discussion

Add a Post

Show all posts by recent activity

my ans

l=len(s) c=0 start=0 end=3 if l<3: print("Number of times bob occurs is: 0") else: while(l >= 3): slice=s[start:end] if slice == "bob": c... 2

Previous Next

Hide Notes