HW 4: While Loops; Strings as Sequences

Due October 13th, 6pm, codePost.io
Please submit all files created for this assignment to codePost. You can submit as many times as you like up until the deadline. ***If you are planning to use one or more late days, please notify us by sending a private Piazza message to "Instructors".***

Familiarize yourself with how to approach a programming assignment and review the grading rubric on this assignment's page in Canvas before getting started.

Code style, documentation, and other requirements
All code style, documentation, and other requirements from the previous assignments apply to this one, too.

We will be checking that you are using GitHub regularly (at least 4 commits per homework). GitHub usage counts toward your professional skills score.

There is no written component for this assignment.

Programming Component (95 pts)

Problem 1: palindrome.py and test_palindrome.py

Write a function called `is_palindrome` that takes a string as input and returns True if the supplied string is a palindrome or False if not. For our purposes, a palindrome is any string that:

- reads the same backwards and forwards, ignoring letter case and spaces*, and
- is at least 2 characters** in length.

*For example, "Radar" is a palindrome even though the first letter is uppercase but the last letter is lowercase. The phrase "taco cat" should also be considered a palindrome because it reads the same backwards and forwards when the space is ignored.

**This includes non-letters, such as punctuation and numbers. For example, "!radar!" would be considered a palindrome.

codePost will use our own test suite to auto-grade your function for correctness. This means there is no specific requirement for a main() function. You may use main() for your own informal testing or leave it out altogether.

Example function output:

```
is_palindrome("madam  Im adam") # True
```

```
is_palindrome("a") # False
is_palindrome("RADar") # True
```

Don't forget to test your function thoroughly in test_palindrome.py! The `is_palindrome` function should only require a few lines of code. We'll be looking for simple solutions that are well tested.

Problem 2: hangman.py and test_hangman.py

Your task is to implement a version of the game Hangman in which a player attempts to guess a secret word one letter at a time. Normally Hangman is a two-player game but in your version the user will play three rounds against the computer. Use the following as the secret word in each round:

1. "APPLE"
2. "OBVIOUS"
3. "XYLOPHONE"

Here are the steps to follow in each round:

1. Print the secret word with each letter replaced with the underscore character, `"_"`
2. Prompt the user to enter a letter or word using the following message, `"Enter a letter or word: "`.
3. If the user enters a single letter, check if that letter is in the secret word (case-insensitive). The user can make up to 6 letter guesses in a round. If the user guesses a letter they have already guessed, print the message `"You've already guessed that letter!"` and don't count it as one of their 6 guesses. Go straight to step 6.
4. If the user enters a word, check if the word matches the secret word (case-insensitive). Word guesses should not count toward the user's total number of guesses.
5. After the user has made their guess, check if the game is over. The game is over if the user has guessed 6 letters but not revealed the word (they lose) or, if the user correctly guesses the word.
6. If the game is NOT over:
    a. Print the secret word with all letters *except any the user has correctly guessed* replaced by underscores.
    b. Print `"Your guesses so far: "` followed by each letter the user has guessed so far.
    c. Return to step 2.
7. If the game *is* over, print the result.
    a. If the player won, print `"You win!"`
    b. If the player lost, print `"You lose! The word was "` followed by the secret word
    c. Move on to the next round, if applicable.

When the user has played all three rounds, print "`You won` **<x>** `out of 3`", where **<x>** is the number of rounds the player won.

Don't forget to test your functions in test_hangman.py

Example input / output (user input is shown in blue)

```
_____
Enter a letter or word: A
A____
Your guesses so far: A
Enter a letter or word: e
A___E
Your guesses so far: AE
Enter a letter or word: apple
You win!

_______
Enter a letter or word: a

_______
Your guesses so far: A
Enter a letter or word: e

_______
Your guesses so far: AE
Enter a letter or word: i
___I___
Your guesses so far: AEI
Enter a letter or word: o
O__IO__
Your guesses so far: AEIO
Enter a letter or word: u
O__IOU_
Your guesses so far: AEIOU
Enter a letter or word: s
You lose! The word was OBVIOUS

_________
Enter a letter or word: a

_________
Your guesses so far: A
Enter a letter or word: e
________E
Your guesses so far: AE
Enter a letter or word: e
You've already guessed that letter!
________E
```

```
Your guesses so far: AE
Enter a letter or word: i
________E
Your guesses so far: AEI
Enter a letter or word: o
___O__O_E
Your guesses so far: AEIO
Enter a letter or word: u
___O__O_E
Your guesses so far: AEIOU
Enter a letter or word: s
You lose! The word was XYLOPHONE
You won 1 out of 3
```