PRACTICE ASSESSMENT 4

Task:
Using the starter repository, implement some of the functions below. Write the required tests. You primarily get points for the tests you write. On the actual assessment, you will be asked to write and test:

- one very simple function (2 points)
- one medium difficulty function (5 points)
- one more difficult function, like some of the examples below (3 points)

Setup and Submission:
See readme.md for setup instructions. Commit and push to submit.

Note on Additional Test Cases:
You may write additional test cases as well. Additional test cases will not affect your grade.

Build Specifications:
There are several functions described below. Each has associated test cases. Like the assessment, you get a point for each correct and passing test you write. Like on the real assessment (and similar to real TDD), start with the simplest test cases to get quick points then try to get the rest of the test cases.

As a practice for the assessment, pick at least two of the functions to do. The easier ones are at the top. Don't worry about the number of points, those are just for fun (see extended challenge).

Extended Challenges:
If you have more time, keep working on exercises and see how many points you can score! Track your points. The maximum possible is 51.

continued on next page...

THE CHALLENGES

In misc.js, create a **`countLetters`** function and export it.

- This function has two parameters:
  - **`word`** - a string
  - **`letter`** - a string, a single English letter to find in the word
- The function returns the number of times the letter appears in the word.
- Letters count, even if they do not match case.

Test cases: (2 points each)

- ("hello", "h") → 1
- ("hello", "l") → 2
- ("suspicious", "s") → 3
- ("Apple", "P") → 2
- ("San Francisco", "s") → 2

In misc.js, create a **`convertTemperature`** function and export it.

- This function has two parameters:
  - **`fromTemp`** - a string including a number and a unit, e.g. "32F", "23C"
  - **`toUnit`** - a string indicating the desired temperature unit, e.g. "F", "C"
- The function converts the fromTemp to the desired unit and returns the result as a number.

Test cases: (2 points each)

- ("50F", "F") → 50
- ("28C", "C") → 28
- ("32F", "C") → 0
- ("20C", "F") → 68
- ("-15C", "F") → 5

continued on next page...

In misc.js, create a **`reverseCase`** function and export it.

- This function has one parameter, a string.
- It returns a new string that has the same letters and characters, but the capitalization of every letter is reversed.

Test Cases (1 point each)

- "a" → "A"
- "A" → "a"
- "HI" → "hi"
- "Car" → "cAR"
- "PoPuLAR" → "pOpUlar"
- "" → ""
- "123" → "123"
- "2:00pm" → "2:00PM"

In misc.js, create a **`scorePalindrome`** function and export it.

- This function has one parameter, a string.
- It calculates the palindrome score by counting the number of characters that are mirrored and returns the score.
- A character is mirrored if it's in the same position forwards and backwards. For example, if the first and last characters are the same, that's one point. If the second and the next-to-last characters are the same, that's a point.
- When a word has an odd number of letters, the middle letter does NOT count as a point.

Test Cases: (1 point each)

- "" → 0
- "noon" → 2 (explanation: n and o are mirrored.)
- "socks" → 1 (explanation: s is mirrored.)
- "peoples" → 1 (explanation: e is mirrored)
- "java" → 0
- "pop" → 1 (explanation: the middle letter (o) by itself does not count)
- "a" → 0
- "abcdcxa" → 2
- "abcdeedcbx" → 4

continued on next page...

In misc.js, create a **`countDoubleLetters`** function and export it.

- In the file word-utils.js, create a **`countDoubleLetters`** function and export it.
- This function has one parameter, which is a single word as a string.
- The function calculates and returns the number of double letters in the word. (e.g. success has two double letters, "cc" and "ss", so return 2.
- Hint: If you have trouble figuring this algorithm out completely, you can still get some points for correct and passing tests.

Test Cases (2 points each)

- "sandy" → 0
- "apple" → 1
- "poll" → 1
- "success" → 2
- "fizzbuzzers" → 2
- "feelinggood" → 3
- "mississippi" → 3