CS 115 - Introduction to Programming in Python

Lab 03

Lab Objectives: Functions

Notes:

- You should not use string functions, lists, tuples, dictionaries in your solution.
- For each of the functions below, you should include a docstring comment. The docstring should have the following format:

```
"""Summary or Description of the Function

    Parameters:
    argument1 (type): Description of arg1

    Returns:
    type: return value
"""
```

1. Create a program, `Lab03_yourname_Q1.py` that is used to capitalize neat reversible words, which will be defined below. The program should define the following functions:

   a) `is_neat_reversible(s)`: takes a string s as a parameter and returns True if the input string s is neat reversible, False otherwise.
   A string is *"neat reversible"* if, after moving its first character to the end, reversing it results in the original string. For example, the word "uneven" is *"neat reversible"* since moving its first letter, "u", to the end gives "nevenu", which when reversed gives "uneven" again.

   Examples:
   `is_neat_reversible`('uneven') -> True
   `is_neat_reversible`('potato') -> True
   `is_neat_reversible`('a') -> True
   `is_neat_reversible`('apple') -> False
   `is_neat_reversible`('the') -> False
   `is_neat_reversible`('') -> False

   b) `uppercase_word_at(s, index)`: takes a string s and an integer *index* as parameters and returns a string which is identical to s except the letters in s starting from the specified *index* up to the first space character are capitalized.

   Examples:
   `capitalize_word_at`('there are many trees', 10) -> 'there are MANY trees'
   `capitalize_word_at`('there are many trees', 15) -> 'there are many TREES'
   `capitalize_word_at`('there are many trees', 2) -> 'thERE are many trees'

   c) `capitalize_neat_reversibles(s)`: takes a string s as a parameter, and using the functions defined in parts a) and b), creates and returns a new string in which all neat reversible words are capitalized.

2. In the script, using the functions defined above, input strings from the user until an empty string is entered. See a sample run below.

Sample Run: (User inputs are red)

```
Enter a sentence: it has a nice grammar
neat reversibles capitalized:
 IT has A nice GRAMMAR

Enter a sentence: the banana is grown in the tropics
neat reversibles capitalized:
 the BANANA IS grown IN the tropics

Enter a sentence: put it in your dresser
neat reversibles capitalized:
 put IT IN your DRESSER

Enter a sentence: an uneven floor puts pressure on the parts of the dresser
neat reversibles capitalized:
 AN UNEVEN floor puts pressure ON the parts OF the DRESSER

Enter a sentence: there are many trees in the garden
neat reversibles capitalized:
 there are many trees IN the garden

Enter a sentence: there are many trees around the city
neat reversibles capitalized:
 there are many trees around the city

Enter a sentence:
bye!
```