CS 5001 Intensive Foundation of CS
Homework 2: More Simple Functions

Due: 5:59pm on Thursday, January 24th

The goal of this assignment is continue practicing writing and testing functions. **You should work individually on this assignment.**

1 Getting Started

You should start by creating a folder specifically for this assignment. Create a `README.txt` file including three things: your name, the course name, and the name of this assignment. You should also use it answer any questions that may be asked in this assignment along with including any comments that you want the grader to consider when grading your assignment. **You should not include any code in your `README.txt` file.**

2 More Functions

Following the same process as we did in Lab 2, you will design, implement, and test the following functions. You should start by downloading the provided `2019spring-cs5001-homework02-StarterFiles.sip` archive provided on Bottlenose and extract the `moreFunctions.py` and `moreFunctionsTest.py` files into the folder that you created specifically for this assignment. You should modifying these files.

2.1 Function: `printFlag`

Design and implement a function called `printFlag` that can print a flag. Your implementation should be implemented in such a way that it only prints single characters: '`*`' , '`R`', and '` `' (space). The pattern should match the following exactly:

```
* * * * * RRRRRRRRRRRRRRRRRRRR
* * * * *
* * * * * RRRRRRRRRRRRRRRRRRRR
* * * * *
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

Hint: each line is 30 characters long in total.

Since this function is not doing much other than printing the flag, testing is a lot less interesting, but it provides us a way to run the code in an automatic way. Your *Given-When-Then* could read:

```
Given: --
 When: printFlag is called
 Then: a flag is printed to the screen and is visually inspected
```

2.2 Function: `star`

Design and implement a function called `star` that could be used instead of the `*` operator. Your function should have two parameters that represent the two operands of the `*` operator. *Hint:* remember that the `*` operator works for both numeric and string values.

2.3 Function: `reformatName`

Design and implement a function called `reformatName` that will receives two strings values that representing someone's first and last name. Your function should return a single string containing the first name that is all lower case except for the first letter, a space, and the last name in all upper case.

Testing this function will take a little bit more effort. You will want to think about the different combinations of capitalization that are possible and write a different test for each combination.

Hint: Strings are highly versatile because they come with a bunch of different functions already defined. For example, if you have a string called `s`, you can use it to change the case of the letters in a string by using `s.capitalize()`, `s.lower`, `s.swapcase()` or `s.upper()`. You will need to use two of these functions for this problem, experiment to see what each one does.

3 Submit Files to Bottlenose

Before submitting, be sure that you have added a comment at the top of your source files that includes your name and the current date. Then create an archive of the folder that you created for this assignment containing all of the files that you have created and upload this archive to Bottlenose for grading. When you upload, be sure to check whether there are any auto-graders that evaluated your code. When you have fixed any issues pointed out by the auto-graders and are completely satisfied with your submission, you should complete the Homework 2 Review questions. **You will not be able to resubmit after completing the review**