CS1301 - HOMEWORK 11: RECURSION

Important

1. Due Date: **Monday, December 4**
2. This homework is graded out of 100 points.
3. This is an Individual Assignment. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. You should not exchange code or write code for others. For individual assignments, each student must turn in a unique program. Your submission must not be substantially similar to another student's submission. Collaboration at a reasonable level will not result in substantially similar code.
4. For Help:
    - TA Helpdesk (Schedule posted on class website.)
    - Email TA's or use Piazza Forums Notes:
    - How to Think Like a Computer Scientists [http://openbookproject.net/thinkcs/python/english3e/]
    - CS 1301 Python Debugging Guide [http://www.cc.gatech.edu/classes/AY2016/cs1301_spring/CS-1301-Debugging-Guide/index.html]

5. Don't forget to include the required collaboration statement (outlined on the syllabus). **Failing to include the Collaboration Statement will result in no credit**.
6. Do not wait until the last minute to do this assignment in case you run into problems.
7. Comment or delete all your function calls. When your code is run, all it should do is run without any errors. Having function calls or extraneous code outside the scope of functions will result in a lower grade. (import statements and global variables are okay to be outside the scope of a function)
8. **Read the entire specifications document before starting this assignment.**
9. **IF YOUR CODE CANNOT RUN BECAUSE OF AN ERROR, IT IS A 0%.**

Introduction

The goal of this homework is to allow you to practice and understand how to use recursion to solve problems. Refer to the rubric to see how points will be rewarded for each function. You have been given HW11.py to fill out with instructions in the docstrings. However, below you will find more detailed information on how to complete the assignment. Read it thoroughly before you begin. You have until Monday, December 4th at 11:55pm to complete this assignment.

CS1301 - HOMEWORK 11: RECURSION

NOTE: YOU MUST USE RECURSION TO WRITE EVERY FUNCTION (NOT ITERATION) TO RECEIVE CREDIT FOR THIS ASSIGNMENT!

Function name: `leaning_pyramid`
Parameters: int
Returns: None
Description: Write a function that takes in a positive integer n (n ≥ 1). Use this integer to draw a leaning pyramid made out of stars (*). The base of the pyramid will consist of n stars, the next level of the pyramid will consist of n - 1 stars, etc. The top will be a single star.

Test Cases:

```
>>> leaning_pyramid(5)
*
**
***
****
*****

>>> leaning_pyramid(3)
*
**
***
```

Function name: `reverse_phrase`
Parameters: string
Returns: string
Description: Write a function that accepts one parameter, a string. Use **recursion** to reverse all the characters in the phrase. Keep in mind that you are reversing the entire string regardless of what it contains.

Test Cases:

```
>>> test_one = reverse_phrase("car")
>>> print(test_one)
rac

>>> test_two = reverse_phrase("Karoline likes chocolate")
>>> print(test_two)
etalocohc sekil eniloraK

>>> test_three = reverse_phrase("123456789")
>>> print(test_three)
987654321
```

CS1301 - HOMEWORK 11: RECURSION

Function name: `find_parenthesized_string`
Parameters: string
Returns: string
Description: Given a string that contains a **single** pair of parenthesis, compute recursively a new string made up of only the parenthesis and the contents between them. Assume the single pair of parentheses will always be included in the string and that no other parenthesis will exist in the string. Return the computed string. **You may not use .find in your solution**.

Test Cases:

```
>>> test_one = find_parenthesized_string('(cs1301)abcdefg')
>>> print(test_one)
(cs1301)

>>> test_two =
find_parenthesized_string('aaldsfj(riverdale)lkjadlksfj')
>>> print(test_two)
(riverdale)

>>> test_three = find_parenthesized_string('()')
>>> print(test_three)
()
```

Function name: `factorial_dictionary`
Parameters: int
Returns: dict
Description: Define a recursive function that accepts a positive integer parameter n ($n \geq 1$) and returns a dictionary. The dictionary should contain, as a key, every number from 1 to the number passed in (inclusive). The corresponding value for each key should be the factorial of that number. **You may not use looping in your solution.**

Hint: When trying to solve factorial_dictionary(n), you first need to recursively get factorial_dictionary(n – 1). How would you get from factorial_dictionary(n – 1) to factorial dictionary(n)?

Test Cases:

```
>>> test_one = factorial_dictionary(5)
>>> print(test_one)
{1: 1, 2: 2, 3: 6, 4: 24, 5: 120}

>>> test_two = factorial_dictionary(2)
>>> print(test_two)
{1: 1, 2: 2}
```

```
>>> test_three = factorial_dictionary(1)
>>> print(test_three)
{1: 1}
```

Function name: `pascal_triangle`
Parameters: int
Returns: list of ints
Description: Write a function that takes in an integer parameter n (n ≥ 0) and returns a list of the elements of that row (the nth row) in the Pascal Triangle. Here is a link to what a Pascal Triangle looks like, and the rules behind making one: http://mathforum.org/workshops/usi/pascal/pascal_intro.html!

Hint: When trying to solve pascal_triangle(n), you first need to recursively get pascal_triangle(n – 1). How would you get from pascal_triangle(n – 1) to pascal_triangle(n)?

Test Cases:

```
>>> test_one = pascal_triangle(6)
>>> print(test_one)
[1, 6, 15, 20, 15, 6, 1]

>>> test_two = pascal_triangle(10)
>>> print(test_two)
[1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1]

>>> test_three = pascal_triangle(0)
>>> print(test_three)
[1]
```

Grading Rubric

NOTE: ALL FUNCTIONS IN THIS HOMEWORK MUST BE SOLVED USING RECURSION OR NO CREDIT WILL BE RECEIVED FOR THAT FUNCTION.

```
leaning_pyramid:                10 pts
reverse_phrase:                 20 pts
find_parenthesized_string:      20 pts
factorial_dictionary:           25 pts
pascal_triangle:                25 pts
```

Provided

The following file(s) have been provided to you.

1. HW11.py
   This is the file you will edit and implement. All instructions for what the methods should do are in the docstrings.

Deliverables

You must submit all of the following file(s). Please make sure the filename matches the filename(s) below. Be sure you receive the confirmation email from T-Square, and then download your uploaded files to a new folder and run them.

1. HW11.py
   If this file does not run (if it encounters an error while trying to run), you will get no credit on the assignment.