

Homework 9 - Recursion

CS 1301 - Intro to Computing - Spring 2020

Important

- Due Date: **Tuesday, April 7th, 11:59 PM.**
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
 - TA Helpdesk
 - Email TA's or use class Piazza
 - [How to Think Like a Computer Scientist](#)
 - [CS 1301 YouTube Channel](#)
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

Responsible Coding: Some functions in this assignment will have a 'Responsible Coding' badge next to them. These functions are written to help you think about how you can use your skills as a programmer to analyze problems related to *sustainability*, such as ethics, health, and the environment.

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```

Written by [Anthony Zheng \(azheng47@gatech.edu\)](mailto:azheng47@gatech.edu) & [Caitlin Yang \(caitlinyang@gatech.edu\)](mailto:caitlinyang@gatech.edu)

Draw Pyramid

Function Name: pyramid()

Parameters: height (`int`), char (`str`)

Returns: None (`NoneType`)

Description: Your love of art and programming has combined to form a passion for creating digital artworks. You want to start off slow, so you decide to write a program that will draw a pyramid for you. Write a function that takes in a height (`int`) and a character (`str`) you want to draw with. Your function should draw a pyramid by **printing** to the Shell. **This must be done recursively. You cannot use a loop.**

Note: The height will always be at least 1

Hint: You should be using periods to move the characters to their respective positions

Hint: It might be helpful to solve the problem first without the periods, meaning print out the pyramid with only the character passed in while maintaining the same shape.

```
>>> pyramid(7, 'a')
a
aa
a.a
a..a
a...a
a....a
a.....a

>>> pyramid(4, 'g')
g
gg
g.g
g..g
```

Count Patterns

Function Name: patterns()

Parameters: string (`str`)

Returns: count (`int`)

Description: You decide to take a break from drawing and write some stories instead. You notice a lot of the words you write have a pattern within them. Write a function that takes in a string and returns the number of times a pattern in the format "aba" appears. The string will contain 0 or more patterns and will always contain at least 3 characters.

To clarify, the pattern contains one unique character surrounded by two identical characters ("010", "xYx", "p3p"). The pattern "aaa" does not count.

```
>>> first = 'axab' # the patterns counted are axa
>>> print (patterns(first))
1

>>> second = 'cdcece' # the patterns counted are cdc, cec, and ece
>>> print (patterns(second))
3
```

Statistics

Function Name: stats()

Parameters: a tuple of numbers (`tup`)

Returns: a list of min and max (`list`)

Description: Now it's time to start your homework! You're working on stats homework when you realize you need to find a min and max in a set of given numbers. You don't want to use your measly graphing calculator for this, so you decide to write your own function instead. Write a function that takes in a tuple of numbers and returns a list in the form **[min, max]**. If an empty tuple is passed in, return an empty list.

You cannot use loops, max(), min(), sort(), sorted(), or anything that may trivialize this function. You must use recursion.

```
>>> tup = (1,5,3,8,4,3,0,-25,64,3,104)
>>> print(stats(tup))
[-25, 104]

>>> tup = (1,)
>>> print(stats(tup))
[1, 1]
```

Fibonacci Dictionary

Function Name: fibtionary()

Parameters: num (`int`)

Returns: dictionary (key: `int`, value: `int`)

Description: You're done with stats, but you still have other math homework to go! You're currently learning about the Fibonacci sequence in math class. The Fibonacci sequence is a series of numbers in the pattern `1 1 2 3 5 8 13 21 ...`, where the next number is found by adding up the two numbers before it. Write a function that takes in a num (`int`), and returns a dictionary whose keys are the numbers from **1 to num** and the values are the corresponding fibonacci numbers at that position. The numbers passed in will always be positive.

```
>>> num1 = 5
>>> print (fibtionary(num1))
{1: 1, 2: 1, 3: 2, 4: 3, 5: 5}

>>> num2 = 2
>>> print (fibtionary(num2))
{1: 1, 2: 1}
```

Grading Rubric

Function	Points
pyramid()	25
patterns()	25
stats()	25
fibtionary()	25
Total	100

Provided

The `HW09.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW09.py` file to the appropriate assignment on Gradescope, the auto-grader will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can re-submit this assignment an unlimited number of times until the deadline; just click the “Resubmit” button at the lower right-hand corner of Gradescope. You do not need to submit your `HW09.py` on Canvas.