CSE 331 **Spring 2018**

Project 3

100 points
Due February 15th 8:00 pm
This is not a team work, do not copy somebody else's work.

Reference: Author of this project is Sarah Byrum

Assignment Overview

Your job is to implement a stack class. In addition to that, you must implement a function called Palindrome, which returns True if a given phrase is a palindrome. Use the stack class you created to implement this.

Assignment Deliverables

Stack.py

Be sure to use the specified file name(s) and to submit your files for grading **via D2L Dropbox** before the project deadline.

Assignment Specifications

This stack project would be trivial using methods on Python lists. Therefore, **the use of any python list method (outside of the grow and shrink methods) or use of a collection type outside of the stack class will result in a zero.**
Your task will be to complete the methods listed below.

- __str__
  - Given
  - Provides a string representation of the stack
- get_size
  - Returns number of items currently in the stack
  - Time complexity: O(1)
  - Space complexity: O(1)
- is_empty
  - Returns True if the stack is empty, False if not
  - Time complexity: O(1)
  - Space complexity: O(1)

- Top
  - Returns, but does not remove, the top item from the stack
  - Returns None if stack is empty
  - Time complexity: O(1)
  - Space complexity: O(1)
- Push
  - Adds an item to the top of the stack
  - Returns nothing
  - Time complexity: O(1)*
  - Space complexity: O(1)*
- Pop
  - Removes and returns the top value from the stack, or None if empty
  - Time complexity: O(1)*
  - Space complexity: O(1)*
- Grow
  - Resize the stack to be 2 times its previous size when stack size equals capacity
  - Time complexity: O(N)
  - Space complexity: O(N)
- Shrink
  - Resize the stack to be half its current size when stack size is less than or equal to half its capacity
  - Stack capacity should **never** shrink below 2
  - Time complexity: O(N)
  - Space complexity: O(N)

* refers to amortized time, or average case performance when the operation is done many times. Normally, pushing or popping an item to/from the stack takes constant time, until the case where you have to grow or shrink the stack. Each grow/shrink operation takes O(N) time, but you've waited longer to do it (until the stack is full/half empty), so the cost is "spread out" among the prior insertions/removes.

In addition to the above accessors/methods, you must complete the Palindrome function. This function takes in a phrase and returns True if it is a palindrome, False if not. This function should utilize the stack class created above. As shown in Project3_Main.py Test 3, you should ignore all punctuation and capitalization in the phrase. The time and space complexity for this function should both be O(N).

Assignment Notes

Points will be deducted if your solution has any warnings of type:

- You are provided with the skeleton code for the stack class definition. Your job is to complete the methods given. You may add more methods than what is provided, but may not alter the original method signatures or class definitions in any way.
- The newest distribution python 3.6 interpreter will be used to execute your solution.
- You are required to complete the docstrings for any unmade and created function signatures.
- You should not be concerned about error checking as far as invalid input types (all values are integers), incorrect file names, etc…
- To test your classes, Project3_Main.py is provided. Compare your results to the output below.
- Errors when using your solution that cause the grading script to fail will result in a 25% deduction.
- Your solution will be run against 7 test cases checking for various edge cases against your solution.

Testing your work

Run your project on Pycharm see sample run below

```
-----TEST 1-----
['2'] Capacity: 2
Empty Stack
['4'] Capacity: 2
Top: 8
Top: 5
['4', '8', '6'] Capacity: 4
['4', '7'] Capacity: 2

-----TEST 2-----
['2', '5', '14', '7', '3', '7'] Capacity: 8
['2', '5', '14', '7', '3', '7', '89', '23', '7'] Capacity: 16
['2', '5', '14', '7', '3', '7', '89'] Capacity: 8
['2', '5', '14', '7', '3', '7', '89', '8', '9'] Capacity: 16
Top: 9
Top: 8
['2', '5', '14', '7', '3', '7', '89', '8', '9', '15', '3', '7', '8', '45'] Capacity: 16
Top: 5
Top: 9
['2', '5', '14', '7', '3', '7', '89', '8', '9', '15', '3', '7', '8', '45', '34', '2', '5', '9'] Capacity: 32

-----TEST 3-----
True
True
False
True
True

Process finished with exit code 0
```