**CIS 313, Intermediate Data Structures Fall 2019**
**CIS 313 Lab 1**


**Due: Friday at 11.55 pm, Oct. 25, 2019**
**This lab involves implementing linear data structures - stacks and queues.**

# Overview

Fill out all of the methods in the provided skeleton code. You may add additional methods, but should NOT add additional public data to Stack or Queue. You also should not change the name of any of the classes or files. **The isPalindrome() method accepts strings as inputs.**

You may not use Python built-in data abstractions like lists or dictionaries. You may use string, and string functions if you think they will help. **You may use lists to store the values of the stack or queue and return the list within the printStack() and printQueue() methods. This will make writing test cases a lot easier and this is the way I wrote test cases.**

Some general things to consider while doing the assignment:

- **The input to the isPalindrome() method is a single string.**
- In isPalindrome, iterate over the characters of the string, and process with Stack and Queue.
- You should think about how to use a Stack and Queue to determine if a string is a palindrome.
- In the Stack and Queue classes, fill out each of the public methods. Do NOT change the arguments or return types of the public methods. You may, however, add private methods.
- **You need to write your own test cases for this assignment**

## Input Description:

You need to implement all the methods in Stack and Queue Class and the method isPalindrome. You need to write as many test cases as required to test your code. You need to write your test cases to check if the stack, queue class and the isPalindrome function is working properly. A few test cases are given to you in the file **test_lab1.py**. Please make sure your program runs properly for all the possible edge cases.

**Note:**
- Dequeueing from an empty Queue and Popping from an empty Stack should return None.
- **Your program should be able to handle input strings with capital letters and spaces in between the strings.**

**Possible set of input string values:**

Hello
ni t I n
aabacabaa
{(< >)}
{(<<({
344484443 **(note the number input given is a string)**

**This is not an exhaustive list but your code should be able to handle strings with spaces and capitalizations. You can implement this functionality in isPalindrome functions.**

**Hint: convert all the capital letters to lowercase and remove the spaces in the string before processing it.**

# Output Description:
For each line of the input, if the string of numbers reads the same way forward as backwards, you function isPalindrome should return True. Otherwise, it should return false. For example, using the sample input above, your program should output:

False
True
True
False
True
True

# Testing Protocol

We strongly suggest you test your program in the following ways:
- While creating each individual class, write test cases to test queues, stacks and isPalindrome separately.
- Test with multiple examples to make sure every method in the class is working properly.

**Note: I have included the test cases for all the extra-credit questions as well in gradescope. So do not be surprised if some test cases fail. As long as you score 70 points you have implemented the basic requirements for the project and will probably get 100 points for the project. If you implement the extra credit part as well, you should be able to pass all the extra credit test cases as well. So this means you have successfully implemented the extra credit test cases and will get 120 points.**

# Files to submit:

You should submit the **lab1.py** with the filled in methods fo all the classes. You should also submit the **test_lab1.py** file with all the test cases that you had written to test your code. These two files should be compressed and uploaded in **.zip** format under the filename **"submission.zip"**. Please make sure you select the **lab1.py** and **test_lab1.py** files and compress them. Do not compress the folder named submission directly which will result in another subfolder named submission being created after converting to zip format. The screenshot is an example on how to do achieve this. Once you submit your files to Gradescope, you will get the results and the score for your assignment. If any of the test cases fail, you need to modify your code and resubmit until all the test cases pass in order to get full credit.
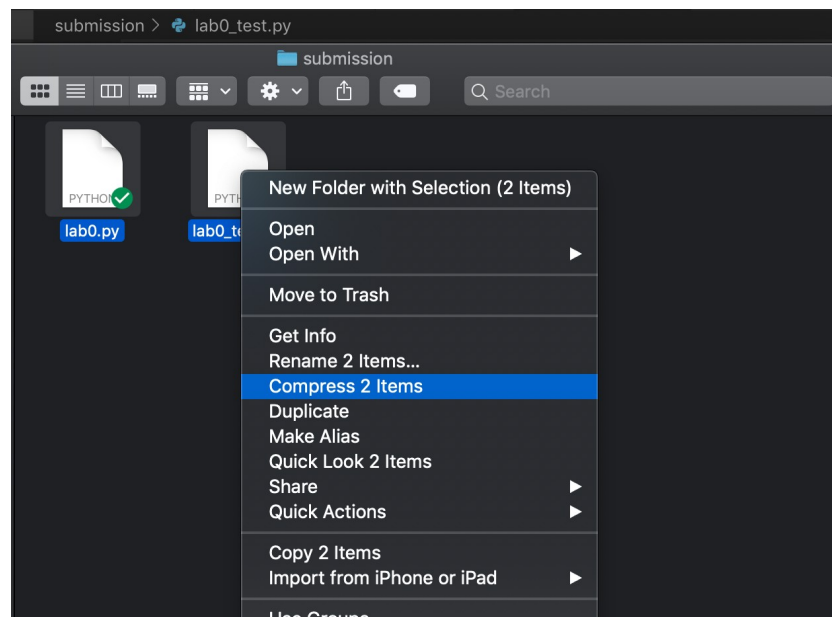


# Grading:

This assignment will be graded as follows:

- AutoGrader 70 points
- Style 30 points
- Extra Credit 20 points

**To earn points for style, your code must be clear enough for us to understand. Further, you may not use any data structures from the python library including lists, and dictionaries**

## Extra Credit 20 points

To receive points for the extra credit you must implement a queue using two stacks:
  • Create a class called TwoStackQueue, which has two stacks as member variables
  • The signature of this class should be the same as the Queue class.
  • It should have the same public methods
  • Each public method should match in argument types (argument number, etc), and return type.
  • Note that this is not the same as having the same methods.
  • In particular, you may modify the bodies of public methods, and change add or remove private
  • methods.
  • Use the TwoStackQueue instead of a Queue in the isPalindromeEC() method.


## Testing Extra Credit

In order to test your TwoStackQueue class, simply test it in all the same ways you tested your original program. Pass all the test cases with "zero" points on gradescope to get the extra 20 points.