

CIS 313, Intermediate Data Structures Fall 2020

CIS 313 Lab 1

Due: Monday at 11.55 pm, Oct. 19, 2020

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, files, or existing methods.

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 only use a list to display a stack or queue in `printQueue()` and `printStack()` but not for other public methods.

*“A Palindrome is a sequence of characters that reads the same backward or forward. Example, **TENET**”. In `isPalindrome()`, iterate over the characters in the line, 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

NEW NOTE: You need to add docstrings to the classes and the functions. An example has been provided in the updated lab1.py file. Following link is a tutorial on docstrings.
Link: <https://www.programiz.com/python-programming/docstrings> or <https://bit.ly/33Tf8aM>

Add docstrings to following functions:

When to add docstrings?

1. Add to classes
2. Add to methods that are responsible for major computation such as `enqueue()`, `dequeue()`, `push()`, `pop()`, `isPalindrome()`.
3. You don't have to add for private methods with predefined meanings such as `__init__()` or `__repr__()` though one is present in the lab1.py file but only as an example.
4. You don't have to add docstrings for individual methods with a single line body such as `getNext()`, `setData()`. A brief description in the docstrings of the class is sufficient.

NEW NOTE!! (10/14 - 10PM)

Lab1 - v1.1

The functions `pop()` of class `Stack`, `dequeue()` of class `Queue` and class `TwoStackQueue` must return the element(data) that has been popped() or dequeued().

The AutoGrader has been updated to check for these requirements. You **MUST** recheck your scripts before submission. If you are already returning the required element and your implementation of the data structures is correct then your script should pass the modified AutoGrader.

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 values:

```
Hello  
ni t l n  
&$(^)$&  
My gym  
12TENET12  
63488436
```

Note: All above sequences are of the type string

Output Description:

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

```
False  
True  
False  
True
```

Lab1 - v1.1

False

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
- Test for edge cases

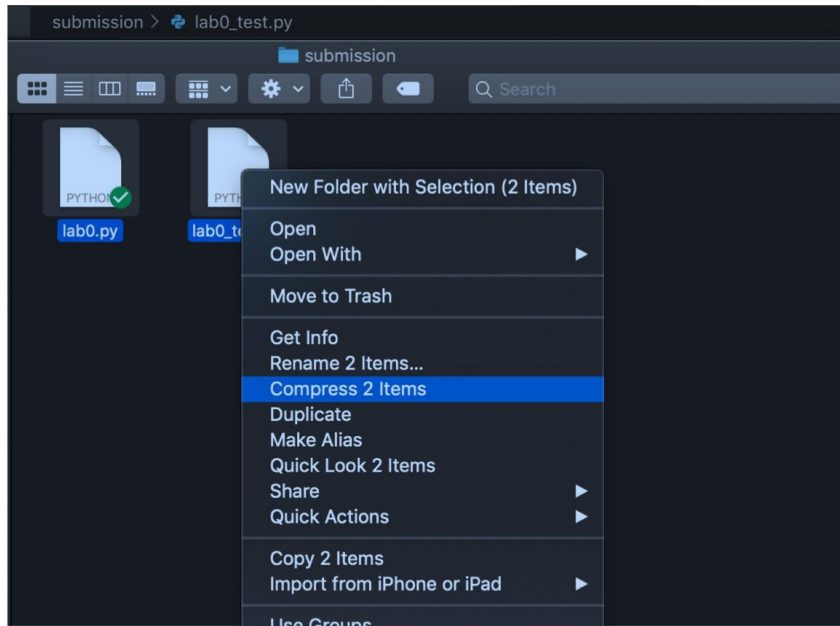
NEW NOTE!!:

- Add at least 5 new test cases to the test_lab1.py file.
- New 5 test cases are required but more are encouraged, as many as would be helpful in checking range of inputs and behaviours.
- Example test case, your code should be able to handle the case of removal of an element from an empty stack or queue.

Files to submit:

You should submit the **lab1.py** with the filled in methods for 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 link on how to 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.

Lab1 - v1.1



Grading

This assignment will be graded as follows:

- AutoGrader - 70 points
- Style - 25 points + 5 points for docstrings (**UPDATED!!**)
- **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**
- **If you do not attempt Extra Credit, you may only see a maximum of 70 points on GradeScope. In that case, you might also see some grading test cases (appended with '_EC') fail which are meant for the extra credit questions.**

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

Lab1 - v1.1

Testing Extra Credit

In order to test your TwoStackQueue class, simply test it in all the same ways you tested your original program.