CIS 313 Lab 1

Due: October 16th, 2016 at 11:59pm

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 methods or public fields to Stack or Queue. You also should not change the name of any of the classes or files.

Recommended strategy

In main, read each line, and call isPalindrome.

In isPalindrome, iterate over the characteres 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, or add new public methods. You may, however, add private methods.

Input
The input will be a single line, with a single number n specifying how many lines follow,
followed by n lists of numbers. Note that all numbers are single digit, and are not separated.
For each line, use your stack and queue to determine if the input represents a palindrome.
The input strings can have any non-negative number of characters.

Sample input
5
30325
1133311
613373316
44
56

Sample output
Not a Palindrome.
This is a Palindrome.
This is a Palindrome.
This is a Palindrome.
Not a Palindrome.

In order to test your Stack and Queue, you can use the testContents class,

which calls the printStack and printQueue methods

Sample input
30325
Sample output
Stack Output:
52303
Queue Output:
30325

Grading

This assignment will be graded as follows:

Correctness 50%
Your program compiles without errors (including the submitted files NOT containing package names at the top. Delete the package name from the top of the files before you submit them): 10%
Your program runs without errors: 10%
Your program produces the correct output: 30%

Implementation 50%
Your Stack class implements all of the proper methods in O(1): 20%
Your Queue class implements all of the proper methods in O(1): 20%
Your palindrome test uses a Stack and a Queue, to achieve linear running time O(n) 10%

To earn points for implementation, your code must be clear enough for us to understand

Further, you may not use any data structures from the Java standard library, the C foreign function interface, or arrays

Extra Credit 20%

Use two stacks to implement a queue:
Create a class TwoStackQueue, which has two stacks as member variables.
The signature of this class should be the same as Queue.
Again, look for amortized complexity O(1) for each queue operation.

Testing Extra Credit
You should create a stack and a queue for the implementation of
isPalindrome, in the previous part of this homework assignment.
For testing the extra credit, change your queue to use the queue you implement with two stacks, and test in the same way.