Problem

Submissions

Leaderboard

Discussions

Editorial

Tutorial

Welcome to Day 18! Today we're learning about Stacks and Queues. Check out the Tutorial tab for learning materials and an instructional video!

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backwards and forwards. Can you determine if a given string, s, is a palindrome?

To solve this challenge, we must first take each character in s, enqueue it in a queue, and also push that same character onto a stack. Once that's done, we must dequeue the first character from the queue and pop the top character off the stack, then compare the two characters to see if they are the same; as long as the characters match, we continue dequeueing, popping, and comparing each character until our containers are empty (a non-match means s isn't a palindrome).

Write the following declarations and implementations:

1. Two instance variables: one for your $stack$, and one for your $queue$.
2. A void pushCharacter(char ch) method that pushes a character onto a stack.
3. A void enqueueCharacter(char ch) method that enqueues a character in the $queue$ instance variable.
4. A char popCharacter() method that pops and returns the character at the top of the $stack$ instance variable.
5. A char dequeueCharacter() method that dequeues and returns the first character in the $queue$ instance variable.

Input Format

You do not need to read anything from stdin. The locked stub code in your editor reads a single line containing string s. It then calls the methods specified above to pass each character to your instance variables.

Constraints

- s is composed of lowercase English letters.

Output Format

You are not responsible for printing any output to stdout.

If your code is correctly written and s is a palindrome, the locked stub code will print `The word, s, is a palindrome.`; otherwise, it will print `The word, s, is not a palindrome.`

Sample Input

```
racecar
```

Change Theme   Java 8 ▼

```java
1  import java.io.*; …
3
4  public class Solution {
5
6      private Stack<Character> stack;
7      LinkedList queuey;
8
9      public Solution(){
10         stack = new Stack<>();
11         queuey = new LinkedList();
12     }
13
14     public void pushCharacter(char c){
15         stack.push(c);
16     }
17
18     public void enqueueCharacter(char c){
19         queuey.addLast(c);
20     }
21
22     public char popCharacter(){
23         return stack.pop();
24     }
25
26     public char dequeueCharacter(){
27         return (char) queuey.remove(0);
28     }
```

Line: 22 Col: 27

⬆ Upload Code as File
☐ Test against custom input

Run Code   **Submit Code**

You have earned 30.00 points!
You are now 3 challenges away from the 4th star for your 30 days of code badge.

57%   19/22

30
Days of Code
★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 12:21:27

Go to Dashboard   **Try a Random Challenge**

✅ **Test case 0**

✅ **Test case 1**

✅ **Test case 2** 🔒

✅ **Test case 3** 🔒

✅ **Test case 4** 🔒

Compiler Message

Success

Input (stdin)   Download

```
1  racecar
```

Expected Output   Downl

```
1  The word, racecar, is a
```

Exit Full Screen View ↙

Change Theme | Java 8 ▼ | ↺ ⚙

```java
import java.io.*;…

public class Solution {

    private Stack<Character> stack;
    LinkedList queuey;

    public Solution(){
        stack = new Stack<>();
        queuey = new LinkedList();
    }

    public void pushCharacter(char c){
        stack.push(c);
    }

    public void enqueueCharacter(char c){
        queuey.addLast(c);
    }

    public char popCharacter(){
        return stack.pop();
    }

    public char dequeueCharacter(){
        return (char) queuey.remove(0);
    }
```

Line: 22 Col: 27

⬆ Upload Code as File
☐ Test against custom input

Run Code   **Submit Code**

You have earned 30.00 points!
You are now 3 challenges away from the 4th star for your 30 days of code badge.

57%   19/22

30 Days of Code

Problem

Welcome to Day 18! Today we're learning about Stacks and Queues. Check out the Tutorial tab for learning materials and an instructional video!

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backwards and forwards. Can you determine if a given string, s, is a palindrome?

To solve this challenge, we must first take each character in s, enqueue it in a queue, and also push that same character onto a stack. Once that's done, we must dequeue the first character from the queue and pop the top character off the stack, then compare the two characters to see if they are the same; as long as the characters match, we continue dequeueing, popping, and comparing each character until our containers are empty (a non-match means s isn't a palindrome).

Write the following declarations and implementations:

1. Two instance variables: one for your *stack*, and one for your *queue*.
2. A void pushCharacter(char ch) method that pushes a character onto a stack.
3. A void enqueueCharacter(char ch) method that enqueues a character in the *queue* instance variable.
4. A char popCharacter() method that pops and returns the character at the top of the *stack* instance variable.
5. A char dequeueCharacter() method that dequeues and returns the first character in the *queue* instance variable.

Input Format

You do not need to read anything from stdin. The locked stub code in your editor reads a single line containing string s. It then calls the methods specified above to pass each character to your instance variables.

Constraints

- s is composed of lowercase English letters.

Output Format

You are not responsible for printing any output to stdout.

If your code is correctly written and s is a palindrome, the locked stub code will print `The word, s, is a palindrome.`; otherwise, it will print `The word, s, is not a palindrome.`
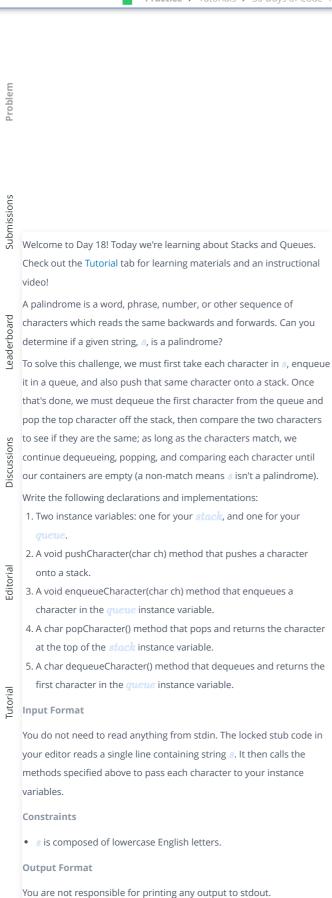
Sample Input

`racecar`

Congratulations
You solved this challenge. Would you like to challenge your friends?

The next challenge in this tutorial will unlock in 12:21:27

Go to Dashboard   **Try a Random Challenge**

✓ **Test case 0**

✓ **Test case 1**

✓ **Test case 2** 🔒

✓ **Test case 3** 🔒

✓ **Test case 4** 🔒

Compiler Message

Success

Input (stdin)   Download

1   `racecar`

Expected Output   Download

1   `The word, racecar, is a`