DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19-05-2020**	**Name:**	**Sathvik R Shetty**
Sem & Sec	**VI B**	**USN:**	**4AL17CS089**

Online Test Summary

Subject	**CGV IA Test**		
Max. Marks	**60**	**Score**	**56**

Certification Course Summary

Course	**Web Development with Python and Javascript**		
Certificate Provider	**Harvard University**	**Duration**	**12 weeks**

Coding Challenges

Problem Statement:

1. We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome

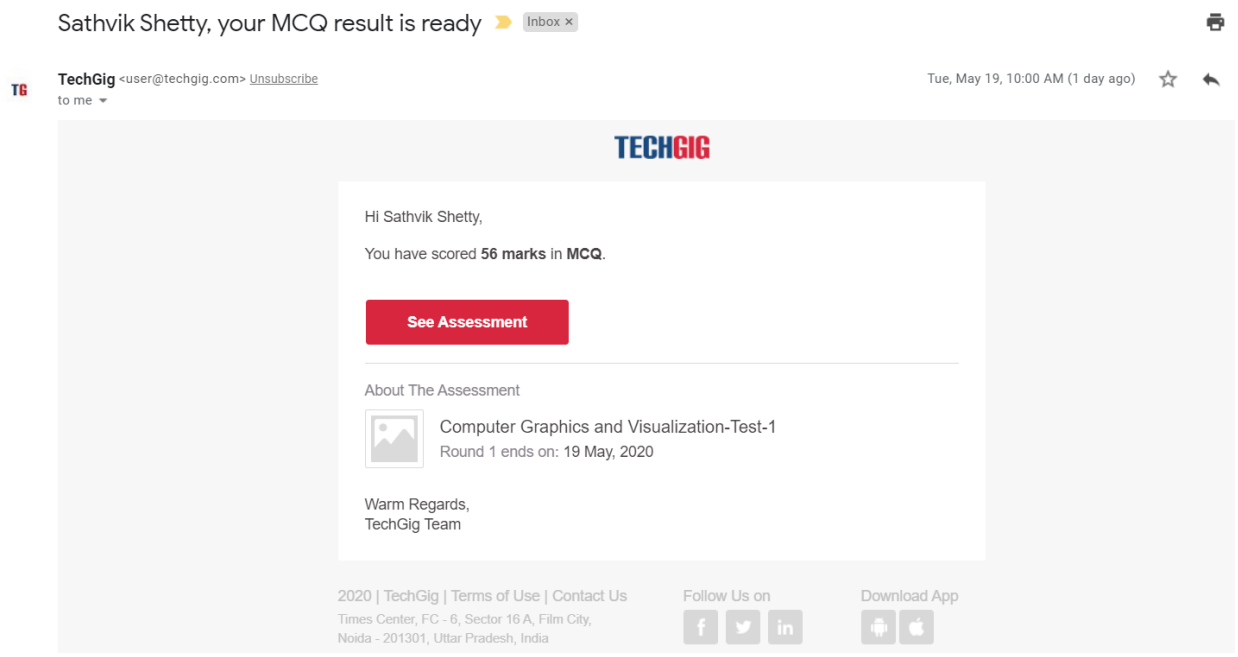
2. Write a simple code to identify given linked list is palindrome or not by using stack.
First take a Stack. Traverse through each node of the linked list and push each node value to Stack.

Status:Completed

Uploaded the report in Github	**Yes**
If yes Repository name	https://github.com/sathvikshetty22/Online-Coding
Uploaded the report in slack	**Yes**

Online Test Details

CGV TEST Details:



Sathvik Shetty, your MCQ result is ready Inbox ×

TechGig <user@techgig.com> Unsubscribe
to me

Tue, May 19, 10:00 AM (1 day ago)

TECHGIG

Hi Sathvik Shetty,

You have scored **56 marks** in **MCQ**.

See Assessment

About The Assessment

Computer Graphics and Visualization-Test-1
Round 1 ends on: 19 May, 2020

Warm Regards,
TechGig Team

2020 | TechGig | Terms of Use | Contact Us
Times Center, FC - 6, Sector 16 A, Film City,
Noida - 201301, Uttar Pradesh, India

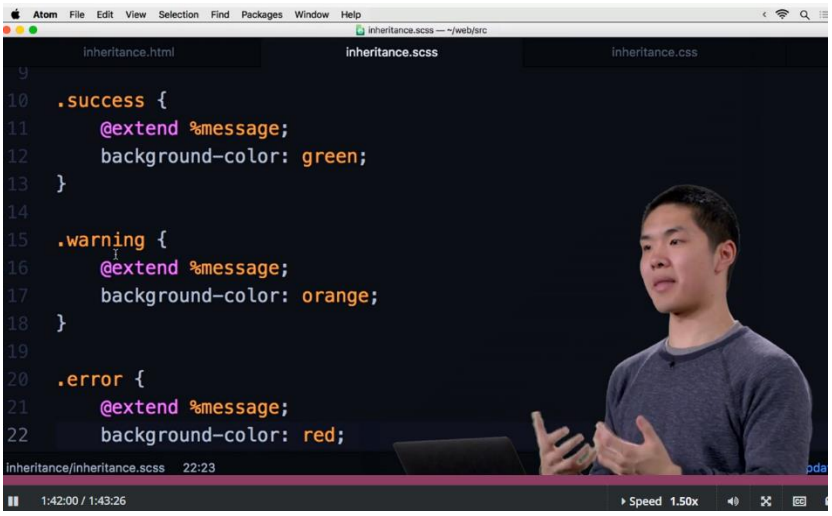
Follow Us on

Download App

Online Certification Details

Lesson-2

- HTML
- CSS



Coding Challenge Details

1.We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome
For example we take "S": S will be the shortest palindrome string.
If we take "xyz": zyxyz will be the shortest palindrome string
So we need to add some characters to the given string or character and find out what will be the shortest palindrome string by using simple java program.

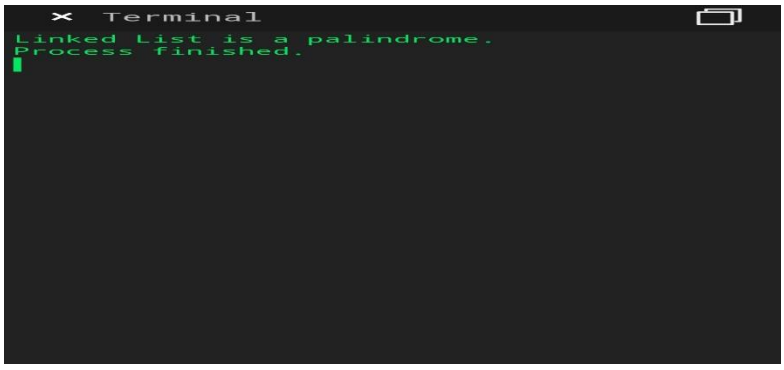
```
1
2  import java.util.Scanner;
3
4  public class ShortestPalindromeDemo {
5
6  public static String shortestPalindrome(String str)
7
8  int x=0;
9  int y=str.length()-1;
10
11 while(y>=0){
12 if(str.charAt(x)==str.charAt(y)){
13 x++;
14 }
15 y--;
16 }
17
18 if(x==str.length())
19 return str;
20
21 String suffix = str.substring(x);
22 String prefix = new StringBuilder(suffix).reverse()
23 String mid = shortestPalindrome(str.substring(0, x)
24
25 return prefix+mid+suffix;
26 }
27
28 public static void main(String[] args)
29 {
30
31 Scanner in = new Scanner(System.in);
32
33 System.out.println("Enter a String to find out shor
34
35 String str=in.nextLine();
36
37 System.out.println("Shortest palindrome of "+str+"
38
39 }
40 }
```

```
Terminal
Enter a String to find out shortest palindrom
pubg is a good game
Shortest palindrome of pubg is a good game is
emag doog a si gbupubg is a good game

Process finished.
```

2. Write a simple code to identify given linked list is palindrome or not by using stack.
First take a Stack. Traverse through each node of the linked list and push each node value to Stack.

Once the traversal & copying is done, iterate through linked list from head node again.
In each iteration, pop one stack element and compare with node value in respective iteration. It is expected to match stack popped value with node value.
In case of all matches, its a palindrome. Any one element mismatch makes it not a palindrome.

```
import java.util.Stack;

// Data Structure to store a linked list node
class Node {
int data;
Node next;

Node(int i)
{
this.data = i;
this.next = null;
}
};

class Main
{
// Function to determine if a given linked list is
public static boolean isPalindrome(Node head)
{
// construct an empty stack
Stack<Integer> s = new Stack<>();

// push all elements of the linked list into the st
Node node = head;
while (node != null) {
s.push(node.data);
node = node.next;
}

// traverse the linked list again
node = head;
while (node != null)
{
// pop the top element from the stack
int top = s.pop();

// compare the popped element with current node's d
// return false if mismatch happens
if (top != node.data) {
return false;
}

// advance to the next node
node = node.next;
}

// we reach here only when the linked list is palin
return true;
}

public static void main(String[] args)
{
Node head = new Node(1);
head.next = new Node(2);
head.next.next = new Node(3);
head.next.next.next = new Node(2);
head.next.next.next.next = new Node(1);

if (isPalindrome(head)) {
System.out.print("Linked List is a palindrome.");
} else {
System.out.print("Linked List is not a palindrome."
}
}
}
```



```
Terminal
Linked List is a palindrome.
Process finished.
```