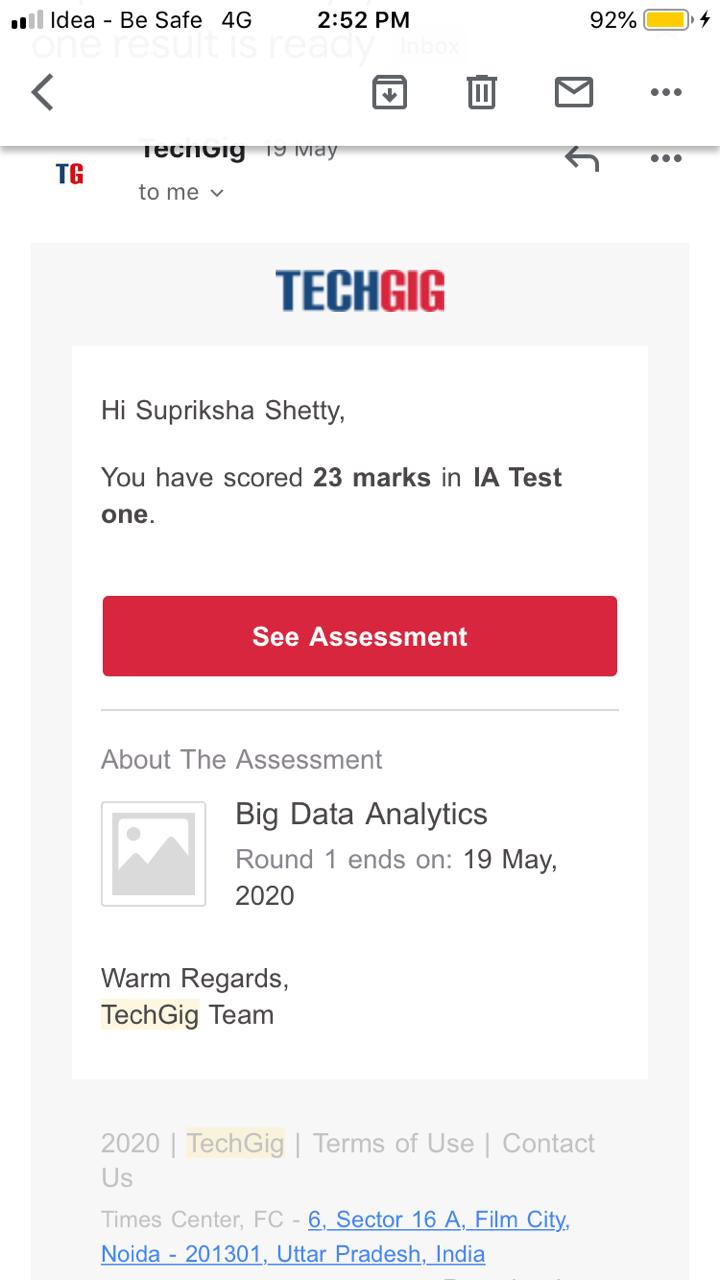
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19/05/2020** | | | | | **Name:** | **Supriksha** | |
| **Sem & Sec** | **8th B** | | | | | **USN:** | **4AL16CS096** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **BDA** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **23** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **AWS** | | | | | | | |
| **Certificate Provider** | | | **AMAZON TRAINING** | | **Duration** | | | **100 Mins** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  **prob1- To add some letters for a given word or letter then to find**  **the shortest palindrome possible**  **Prob2- To check whether the given linked list is palindrome or not** | | | | | | | | |
| **Status:Completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **Yes** | | | |
| **If yes Repository name** | | | | | **Supriksha** | | | |
| **Uploaded the report in slack** | | | | | **Yes** | | | |

Online Test Details:



Certification Course Details:



Coding Challenges Details:

Program 1 :

package shortestpalindromeexample.java;

import java.util.Scanner;

public class ShortestPalindromeDemo {

public static String shortestPalindrome(String str) {

int x=0;

int y=str.length()-1;

while(y>=0){

if(str.charAt(x)==str.charAt(y)){

x++;

}

y--;

}

if(x==str.length())

return str;

String suffix = str.substring(x);

String prefix = new StringBuilder(suffix).reverse().toString();

String mid = shortestPalindrome(str.substring(0, x));

return prefix+mid+suffix;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Enter a String to find out shortest palindrome");

String str=in.nextLine();

System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));

}

Program 2 :

import java.util.Stack;

class Node {

int data;

Node next;

Node(int i)

{

this.data = i;

this.next = null;

}

};

class Main

{

public static boolean isPalindrome(Node head)

{

Stack s = new Stack<>();

Node node = head; // push

while (node != null) {

s.push(node.data);

node = node.next;

}

node = head;

while (node != null)

{

int top = s.pop(); //pop

if (top != node.data) {

return false;

}

node = node.next;

}

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.");

}

}