DAILY ONLINE ACTIVITIES SUMMARY

Date:	**20/05/2020**	**Name:**	**sushmitha**
Sem & Sec	**6th sem**	**USN:**	**4al17cs102**

Online Test Summary

Subject	**SSCD**		
Max. Marks	**30**	**Score**	**20**

Certification Course Summary

Course	**python for machine learning**		
Certificate Provider	**IBM**	**Duration**	**12 HRS**

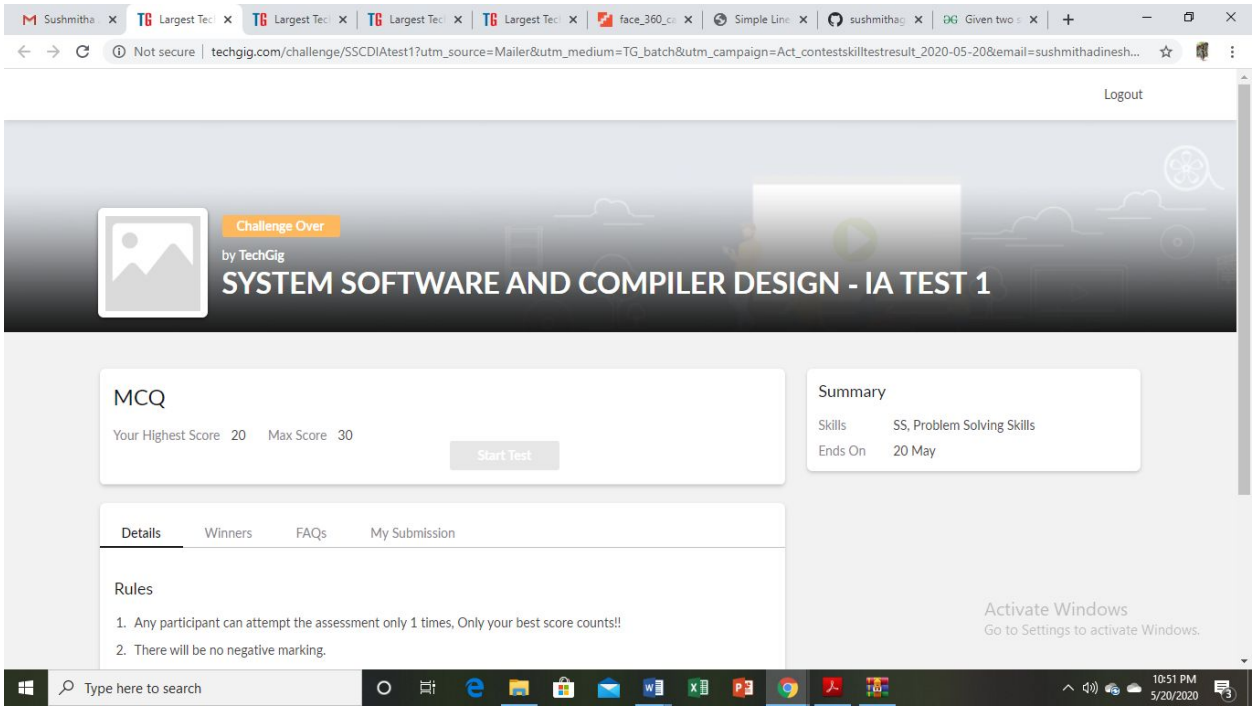
Coding Challenges

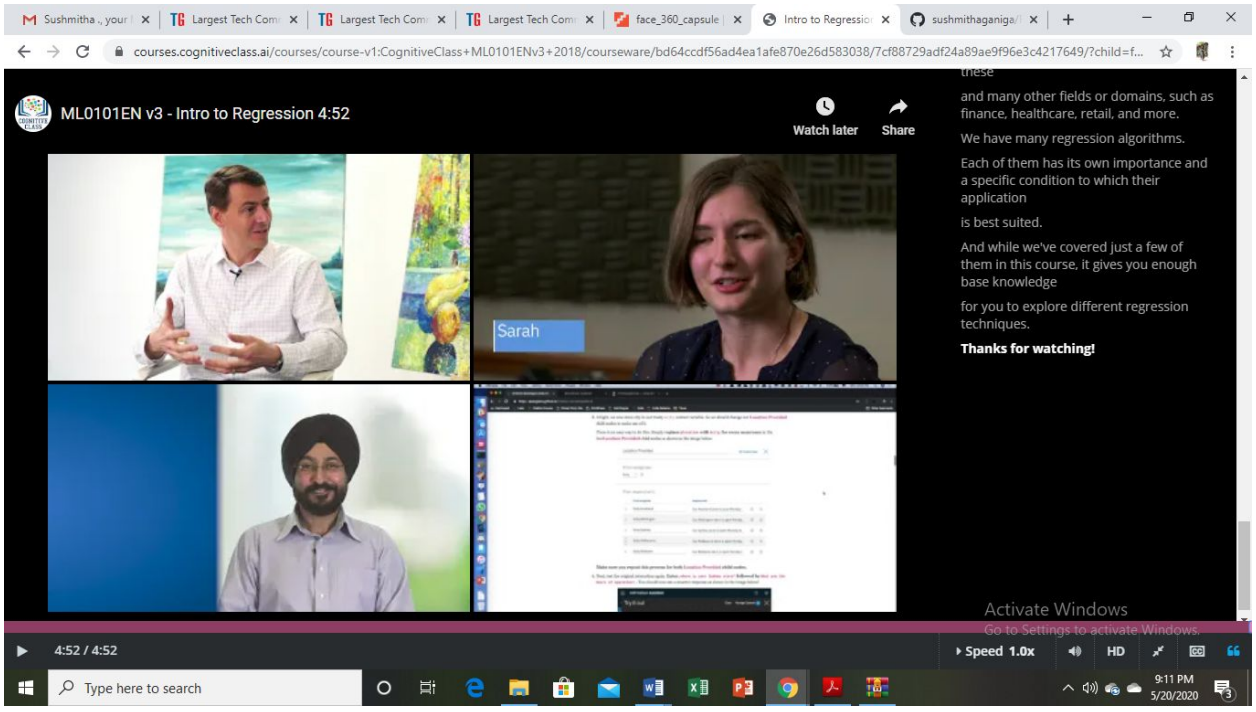
Problem Statement:.	
Status: done	
Uploaded the report in Github	**yes**
If yes Repository name	**https://github.com/sushmithaganiga/coding/tree/master**
Uploaded the report in slack	**yes**

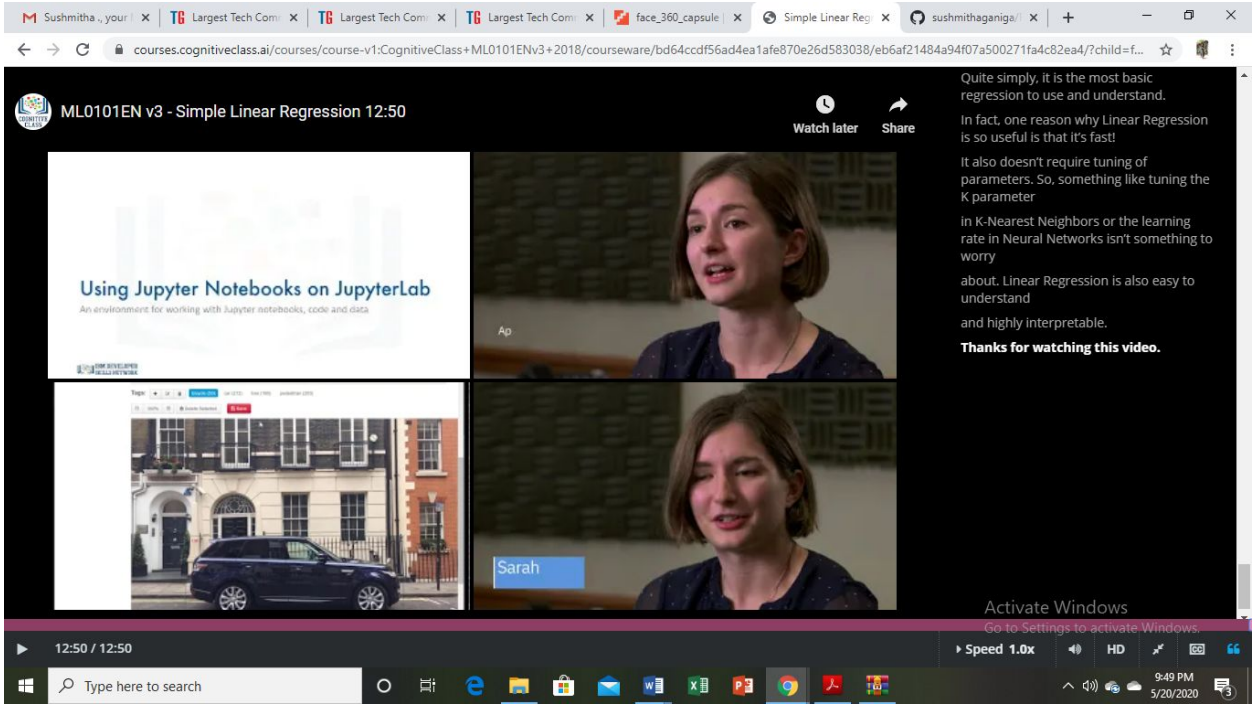
Online Test Details: (Attach the snapshot and briefly write the report for the same)

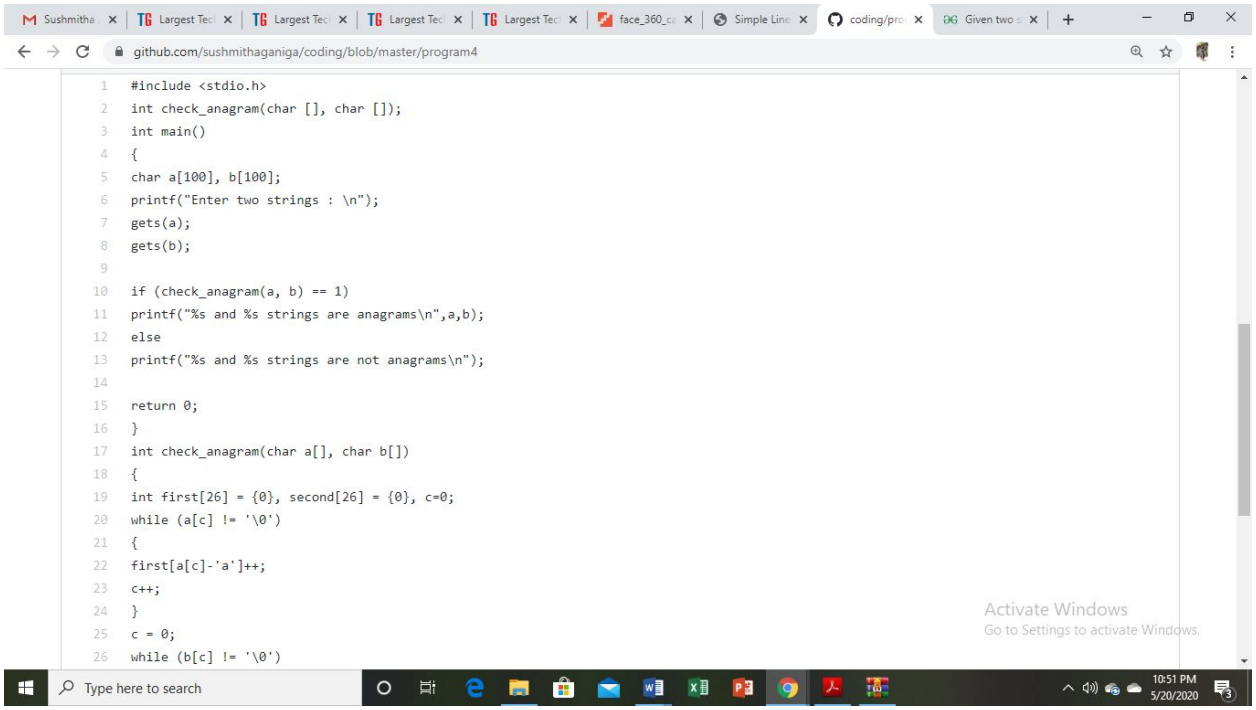
Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

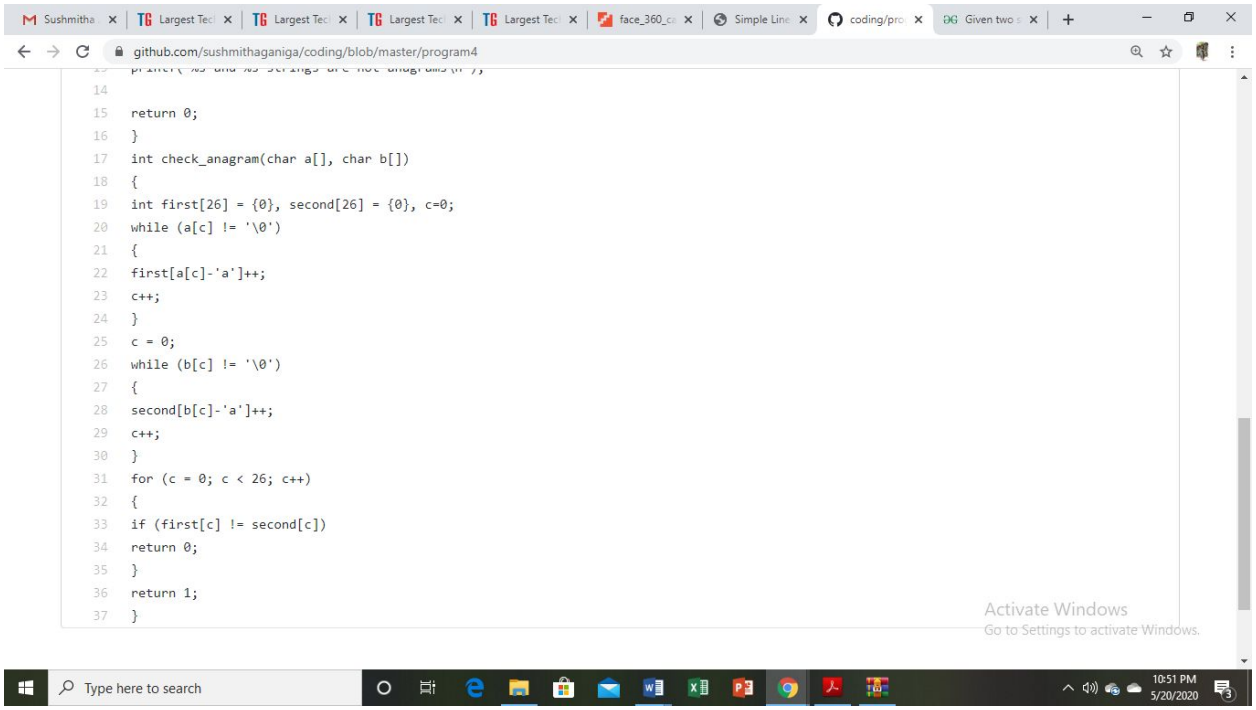



ML0101EN v3 - Intro to Regression 4:52
Watch later
Share
Sarah
these
and many other fields or domains, such as finance, healthcare, retail, and more.
We have many regression algorithms.
Each of them has its own importance and a specific condition to which their application
is best suited.
And while we've covered just a few of them in this course, it gives you enough base knowledge
for you to explore different regression techniques.
Thanks for watching!
Activate Windows
Go to Settings to activate Windows.
4:52 / 4:52
Speed 1.0x
HD
Type here to search
9:11 PM
5/20/2020

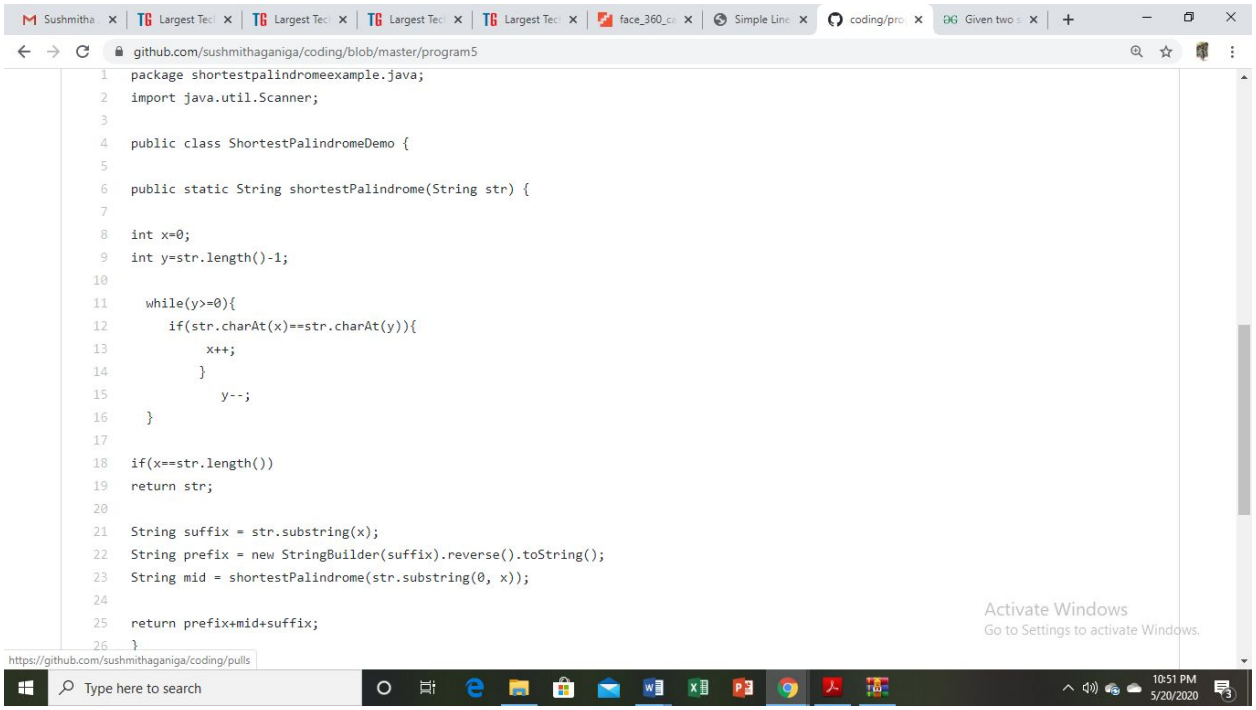

ML0101EN v3 - Simple Linear Regression 12:50
Watch later
Share
Using Jupyter Notebooks on JupyterLab
An environment for working with Jupyter notebooks, code and data
Sarah
Quite simply, it is the most basic regression to use and understand.
In fact, one reason why Linear Regression is so useful is that it's fast!
It also doesn't require tuning of parameters. So, something like tuning the K parameter
in K-Nearest Neighbors or the learning rate in Neural Networks isn't something to worry
about. Linear Regression is also easy to understand
and highly interpretable.
Thanks for watching this video.
Activate Windows
Go to Settings to activate Windows.
12:50 / 12:50
Speed 1.0x
HD
Type here to search
9:49 PM
5/20/2020



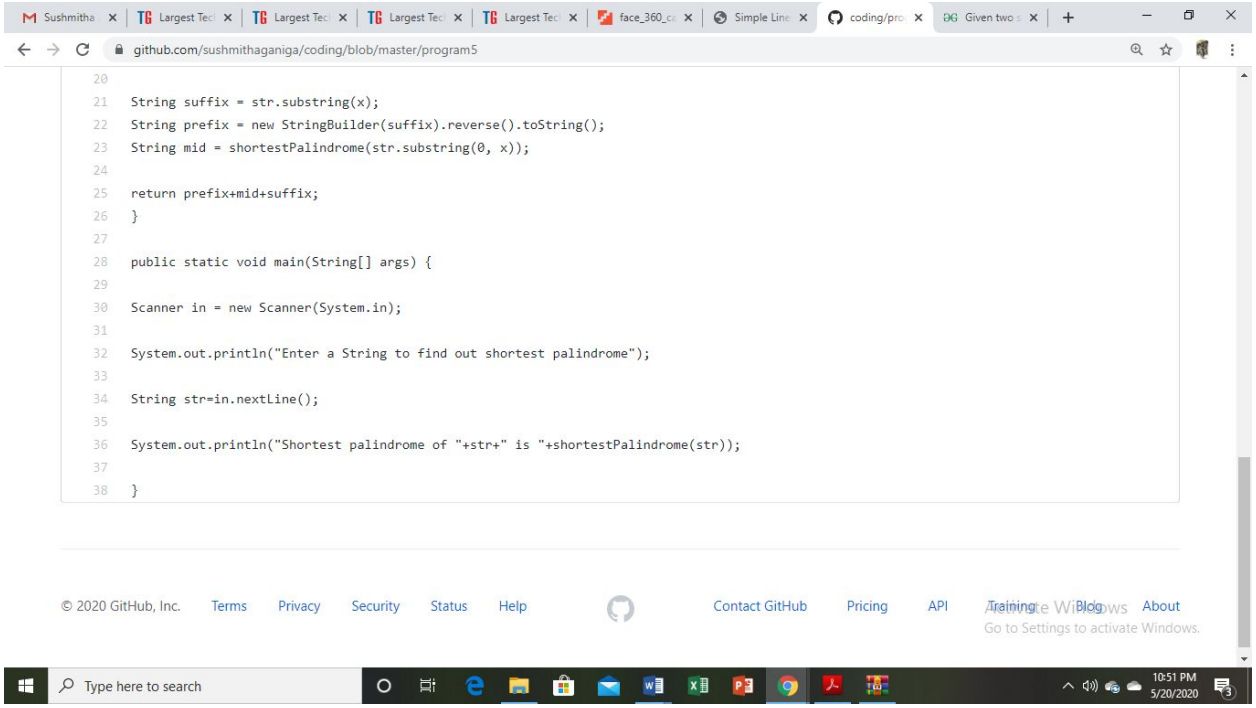
```
#include <stdio.h>
int check_anagram(char [], char []);
int main()
{
char a[100], b[100];
printf("Enter two strings : \n");
gets(a);
gets(b);

if (check_anagram(a, b) == 1)
printf("%s and %s strings are anagrams\n",a,b);
else
printf("%s and %s strings are not anagrams\n");

return 0;
}
int check_anagram(char a[], char b[])
{
int first[26] = {0}, second[26] = {0}, c=0;
while (a[c] != '\0')
{
first[a[c]-'a']++;
c++;
}
c = 0;
while (b[c] != '\0')
```

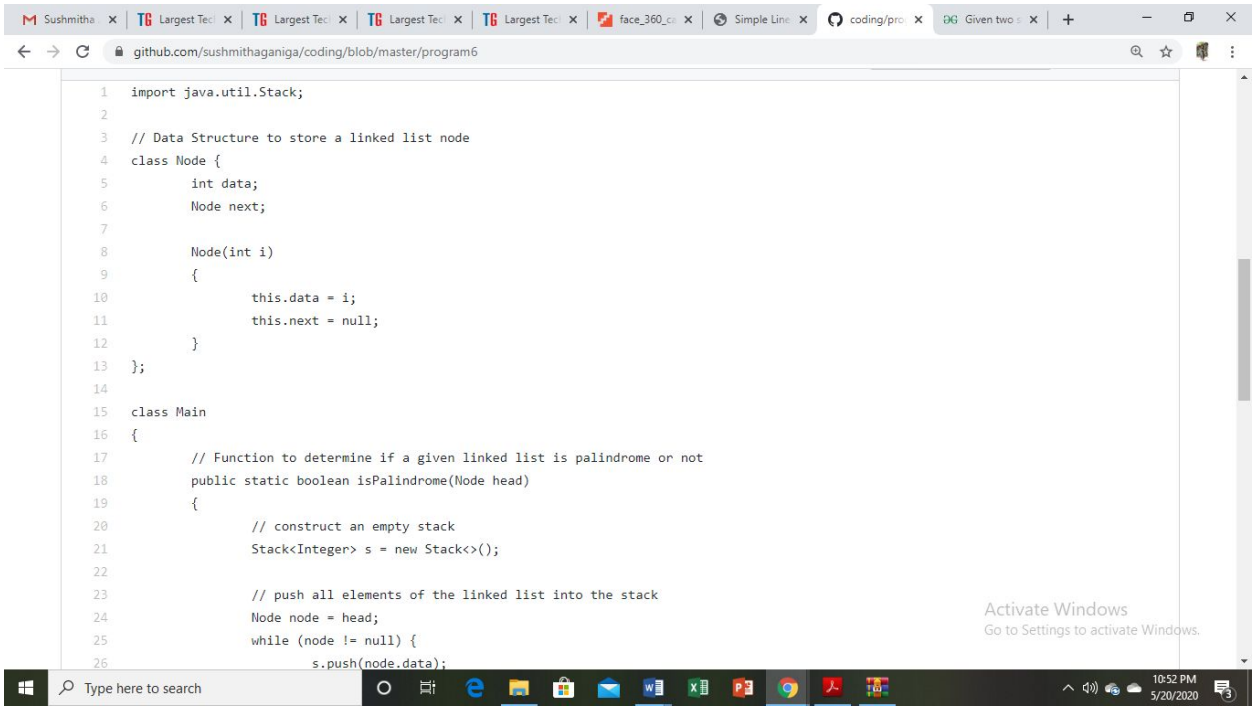
```

return 0;
}
int check_anagram(char a[], char b[])
{
int first[26] = {0}, second[26] = {0}, c=0;
while (a[c] != '\0')
{
first[a[c]-'a']++;
c++;
}
c = 0;
while (b[c] != '\0')
{
second[b[c]-'a']++;
c++;
}
for (c = 0; c < 26; c++)
{
if (first[c] != second[c])
return 0;
}
return 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;
}
```

```
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));

}
```



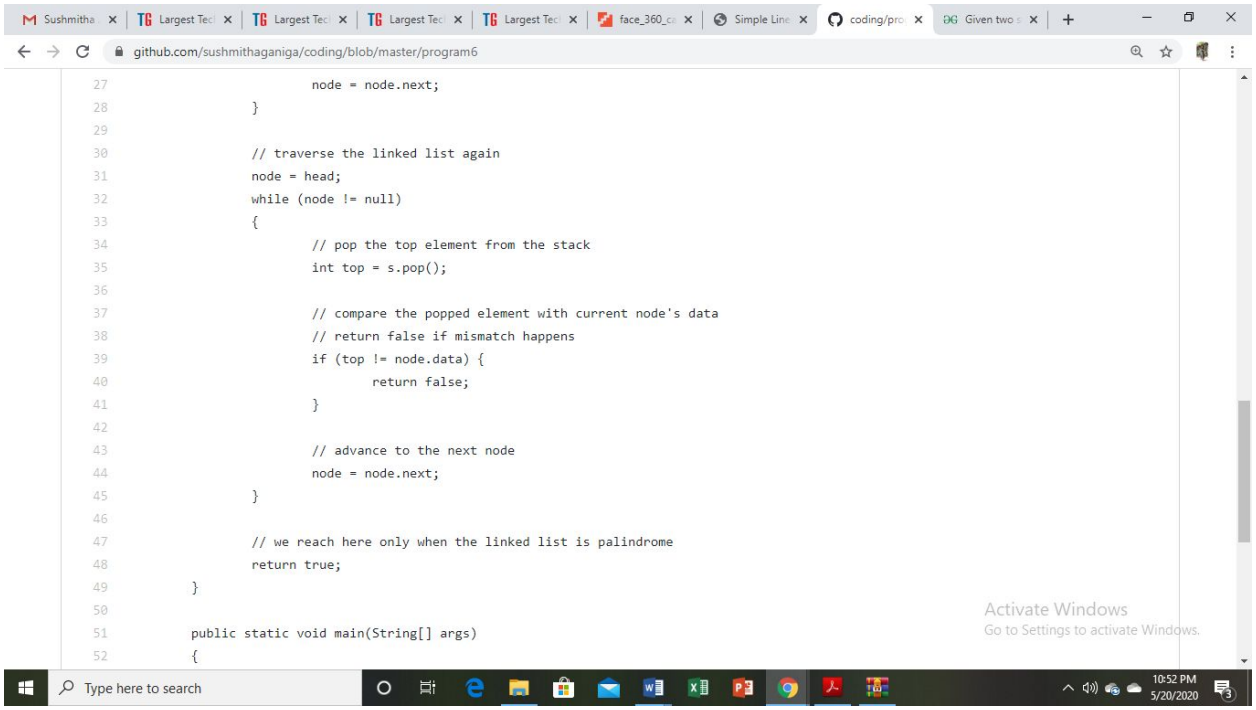
github.com/sushmithaganiga/coding/blob/master/program6

```
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 palindrome or not
        public static boolean isPalindrome(Node head)
        {
                // construct an empty stack
                Stack<Integer> s = new Stack<>();

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

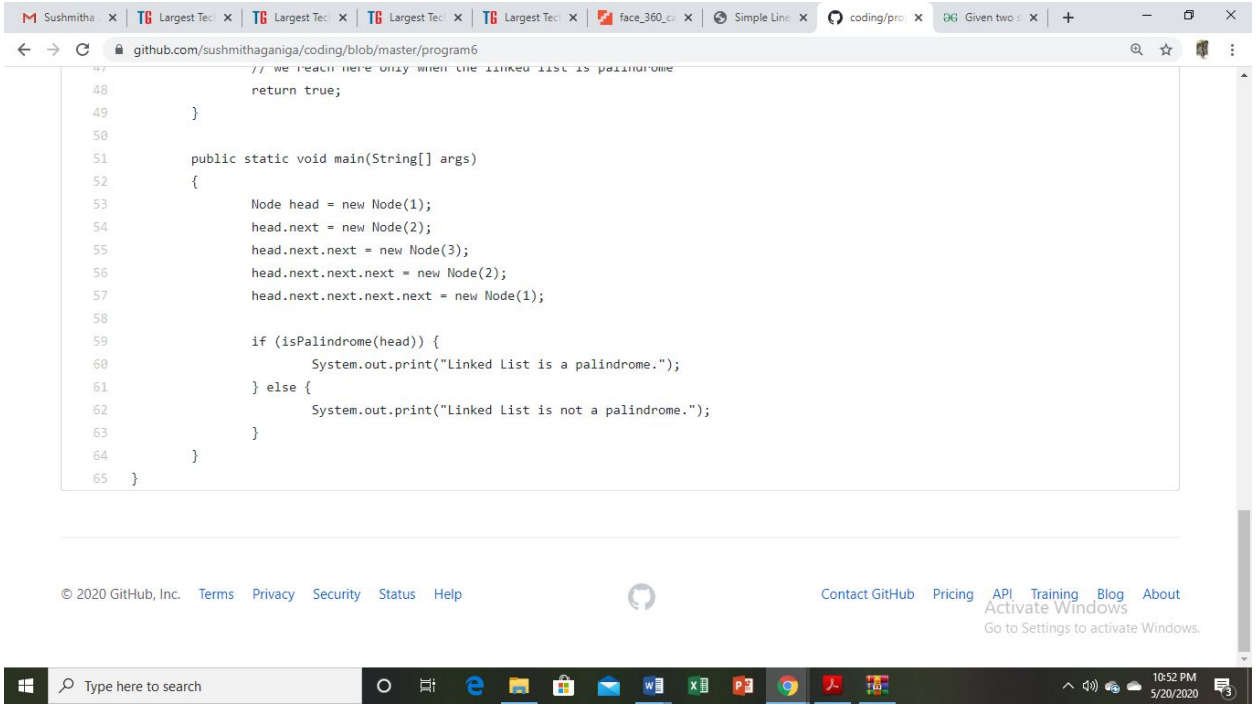
Activate Windows
Go to Settings to activate Windows.

Type here to search

10:52 PM
5/20/2020



```
                        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 data
                        // 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 palindrome
                return true;
        }

        public static void main(String[] args)
        {
```

```
                // we reach here only when the linked list is palindrome
                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.");
                }
        }
}
```