# DAILY ONLINE ACTIVITIES SUMMARY

| | | | |
|---|---|---|---|
| **Date:** | **19-05-2020** | **Name:** | **M.C Suchithra Heggade** |
| **Sem & Sec** | **VI A** | **USN:** | **4AL17CS047** |

## Online Test Summary

| | | | |
|---|---|---|---|
| **Subject** | **CGV IA Test** | | |
| **Max. Marks** | **60** | **Score** | **57** |

## Certification Course Summary

| | | | |
|---|---|---|---|
| **Course** | **Python for Machine learning** | | |
| **Certificate Provider** | **GreatLearning** | **Duration** | **5hr** |

## Coding Challenges

**Problem Statement:**

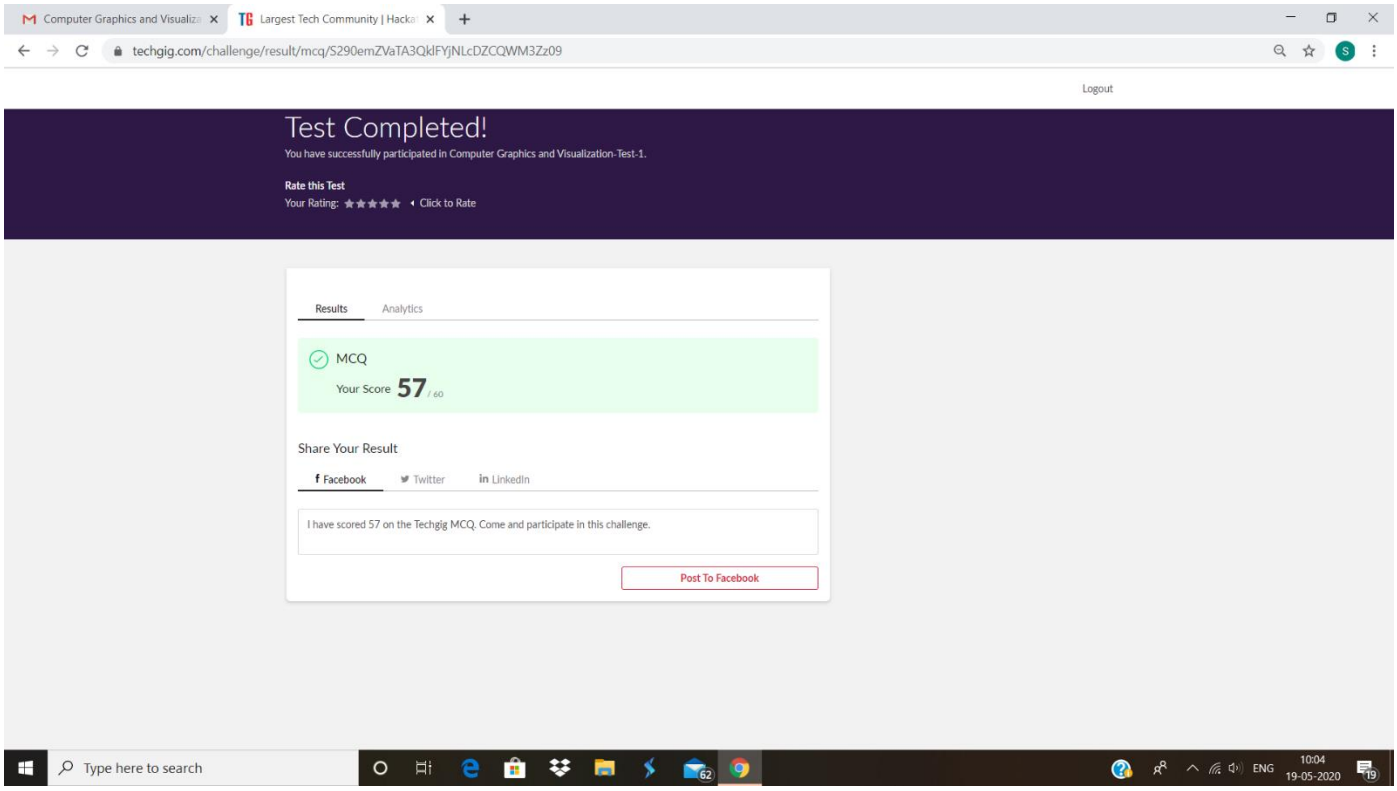
1. We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome For example we take "S": S will be the shortest palindrome string. If we take "xyz": zyxyz will be the shortest palindrome string So we need to add some characters to the given string or character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

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.

| | |
|---|---|
| **Status:Completed,executed** | |
| **Uploaded the report in Github** | **Yes** |

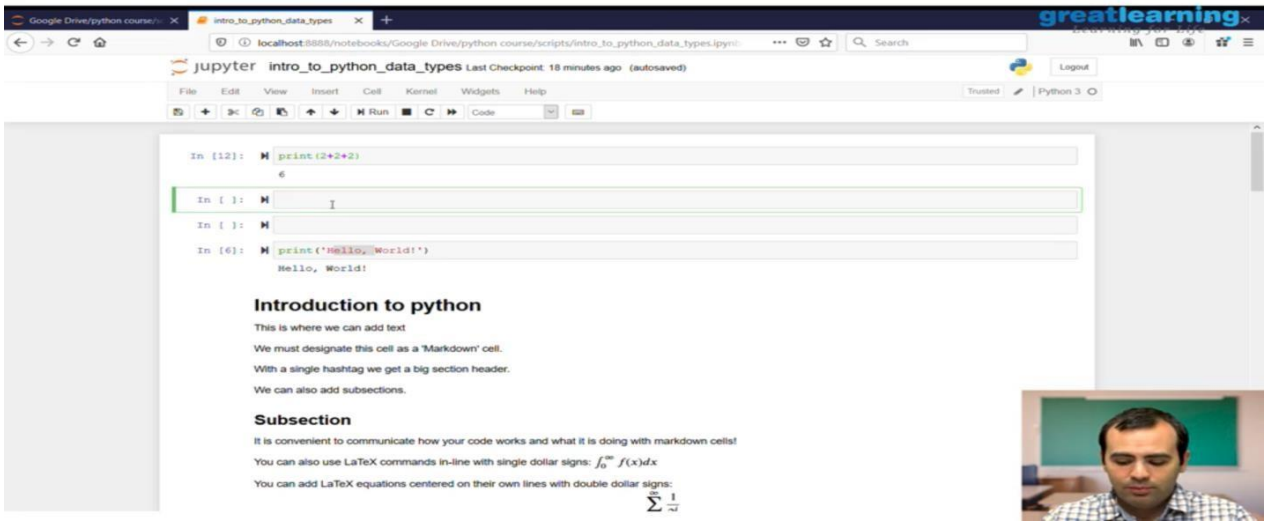
| | |
|---|---|
| **If yes Repository name** | https://github.com/Suchitraheggade/certification-and- online-coding |
| **Uploaded the report in slack** | **Yes** |

# Online Tests



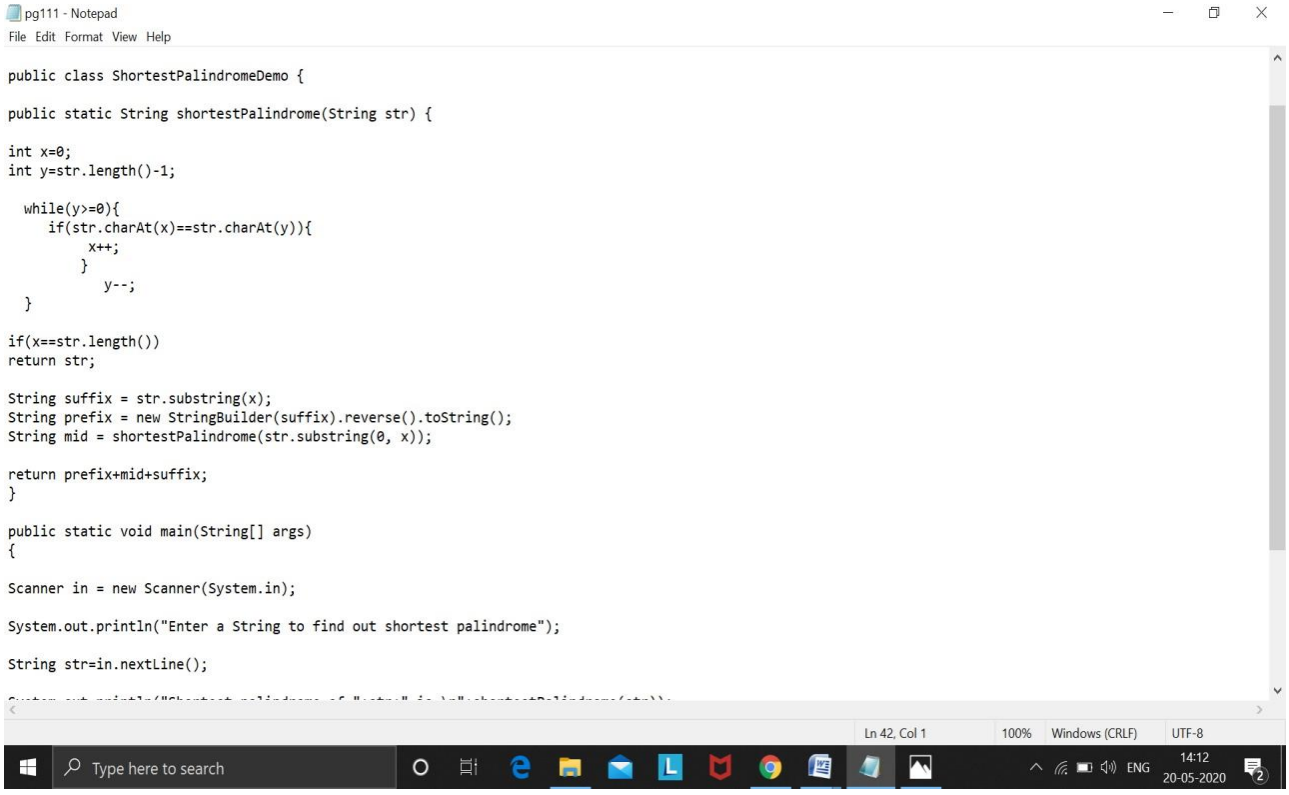

## Online Certification Details

Modules completed:
-Why python, python vs R,
-Anaconda installation,Intro to Jupyter Notebook-2
-Jupiter Notebook Shortcut-2



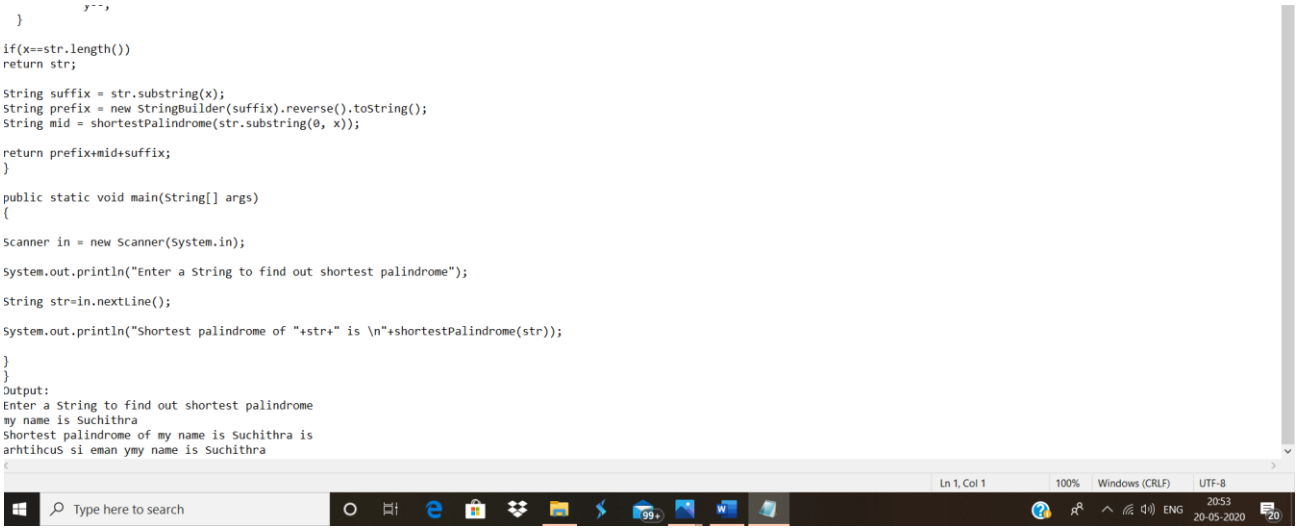

# Coding Challenge Details

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 by using simple java program.

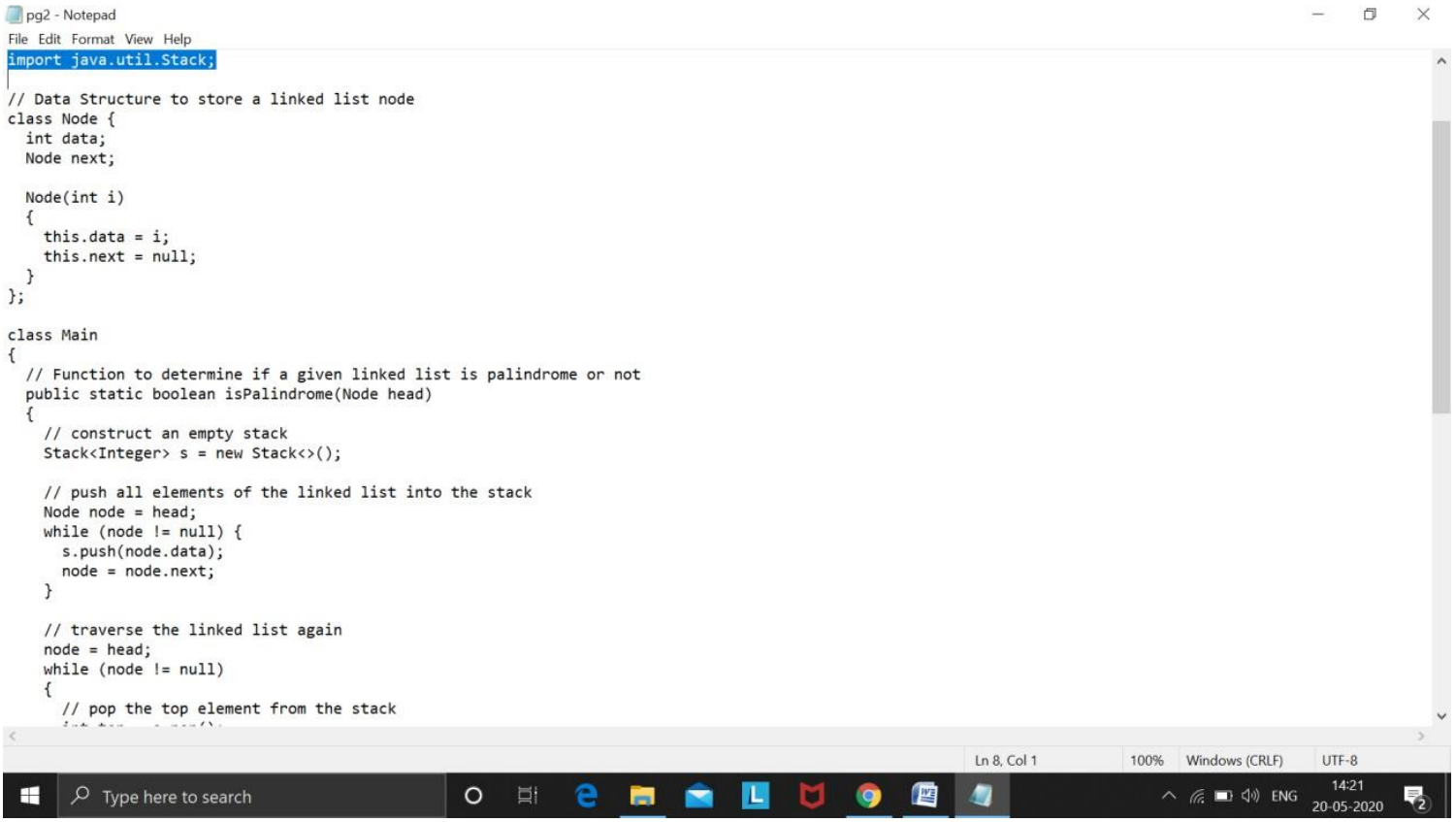


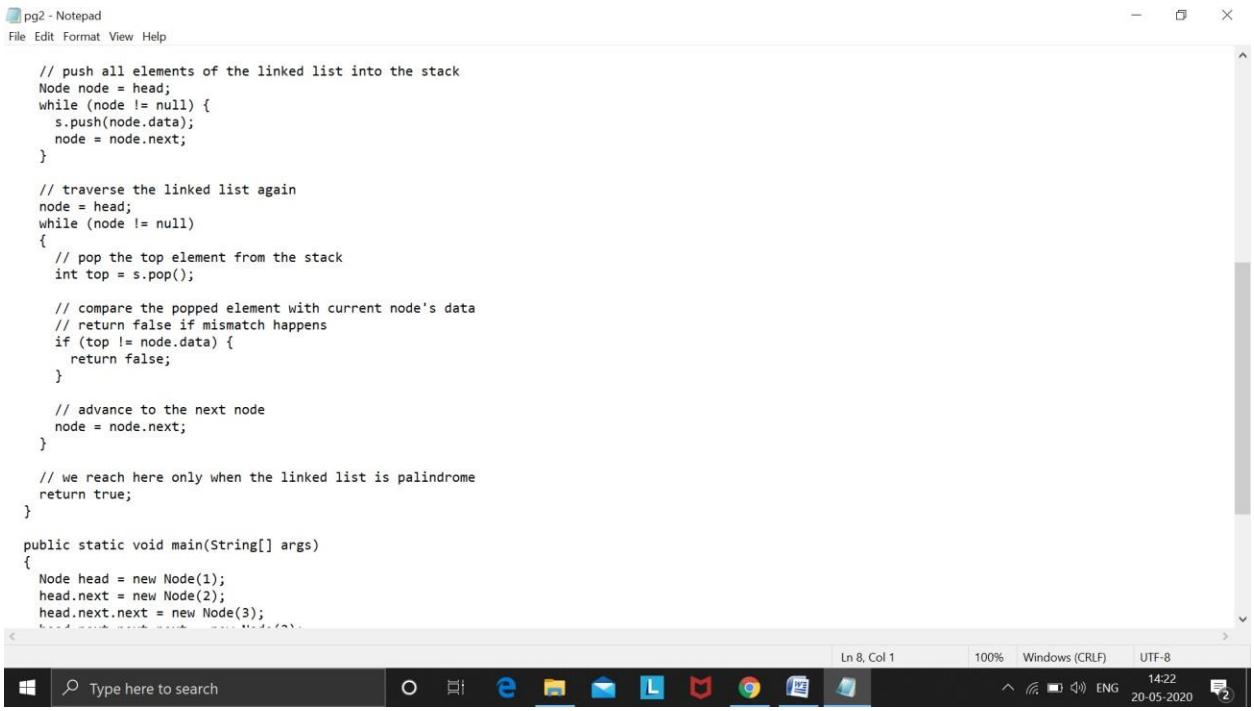

simple java program

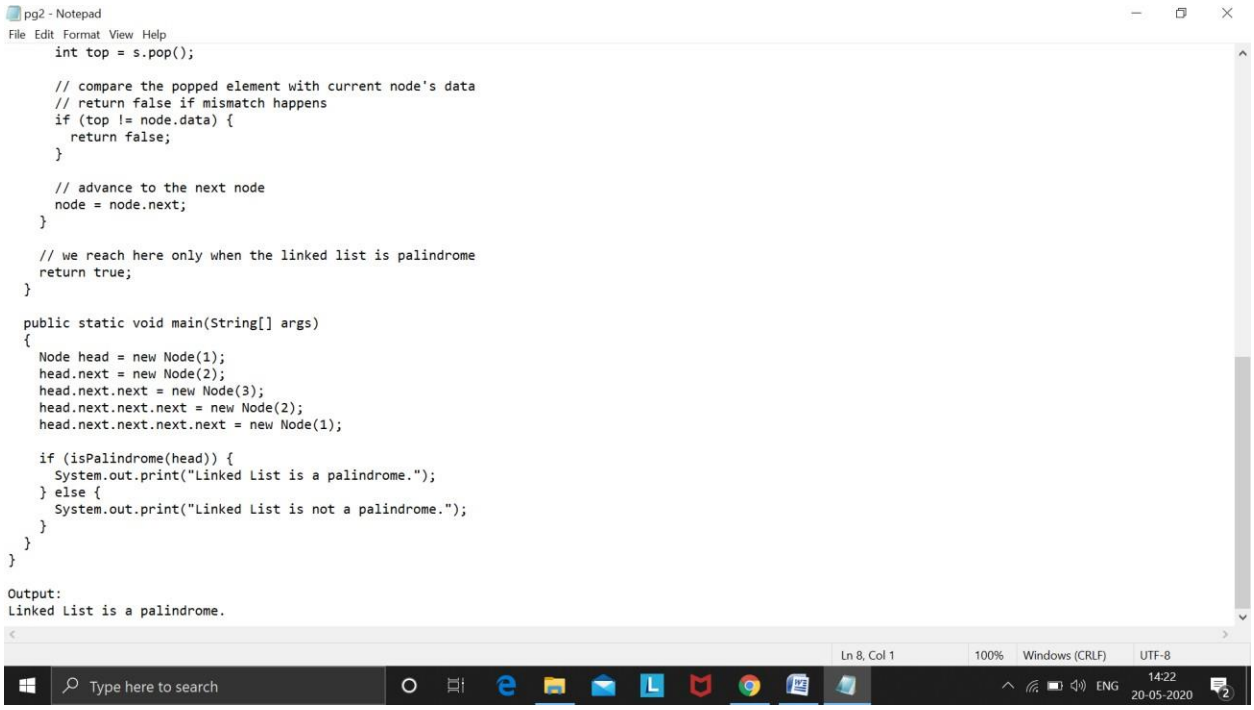
```
            y--;
  }

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 \n"+shortestPalindrome(str));

}
}
Output:
Enter a String to find out shortest palindrome
my name is Suchithra
Shortest palindrome of my name is Suchithra is
arhtihcuS si eman ymy name is Suchithra
```



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.




pg2 - Notepad
File Edit Format View Help
// push all elements of the linked list into the stack
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 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)
{
Node head = new Node(1);
head.next = new Node(2);
head.next.next = new Node(3);
Ln 8, Col 1
100%
Windows (CRLF)
UTF-8
Type here to search
ENG
14:22
20-05-2020


pg2 - Notepad
File Edit Format View Help
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)
{
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.");
}
}
}
Output:
Linked List is a palindrome.
Ln 8, Col 1
100%
Windows (CRLF)
UTF-8
Type here to search
ENG
14:22
20-05-2020