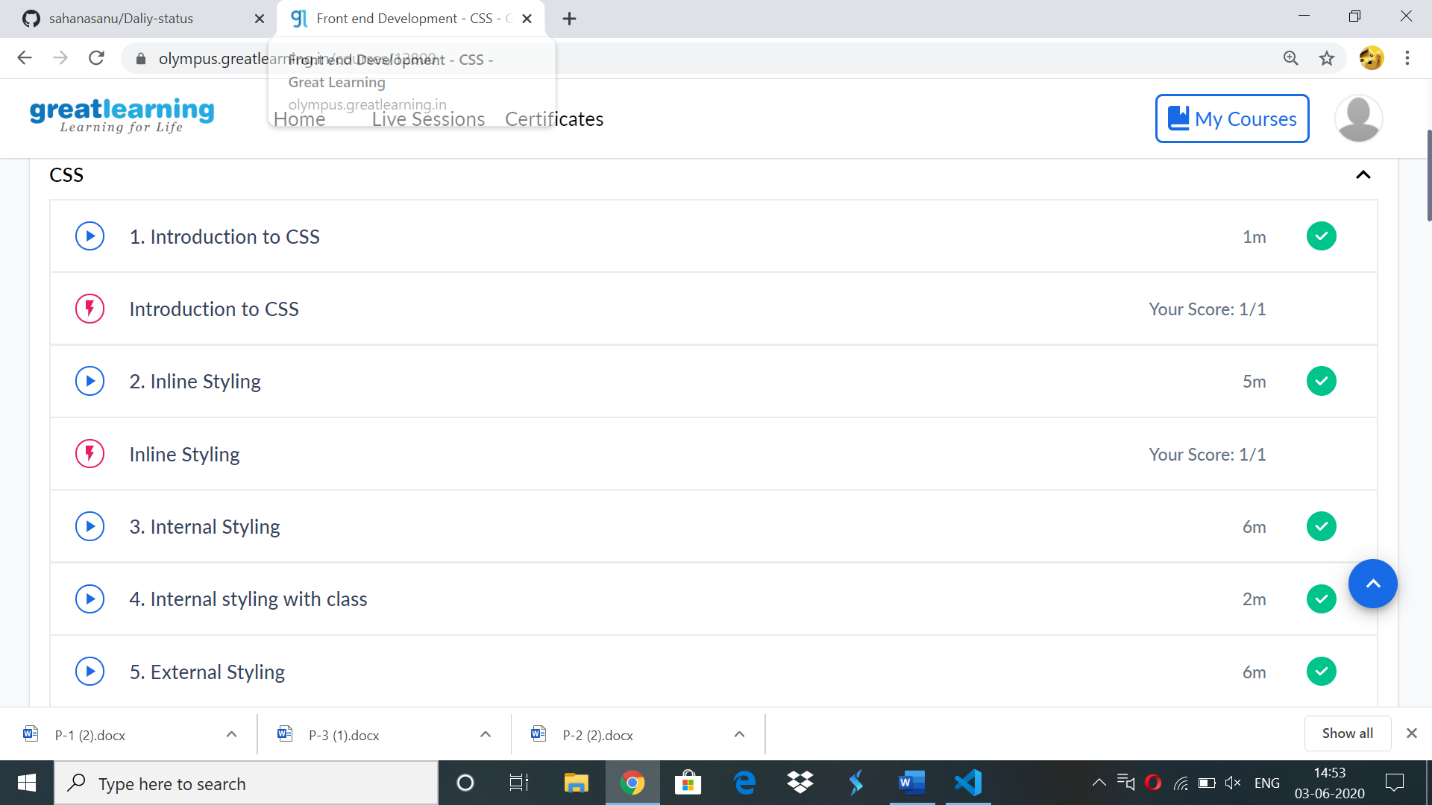
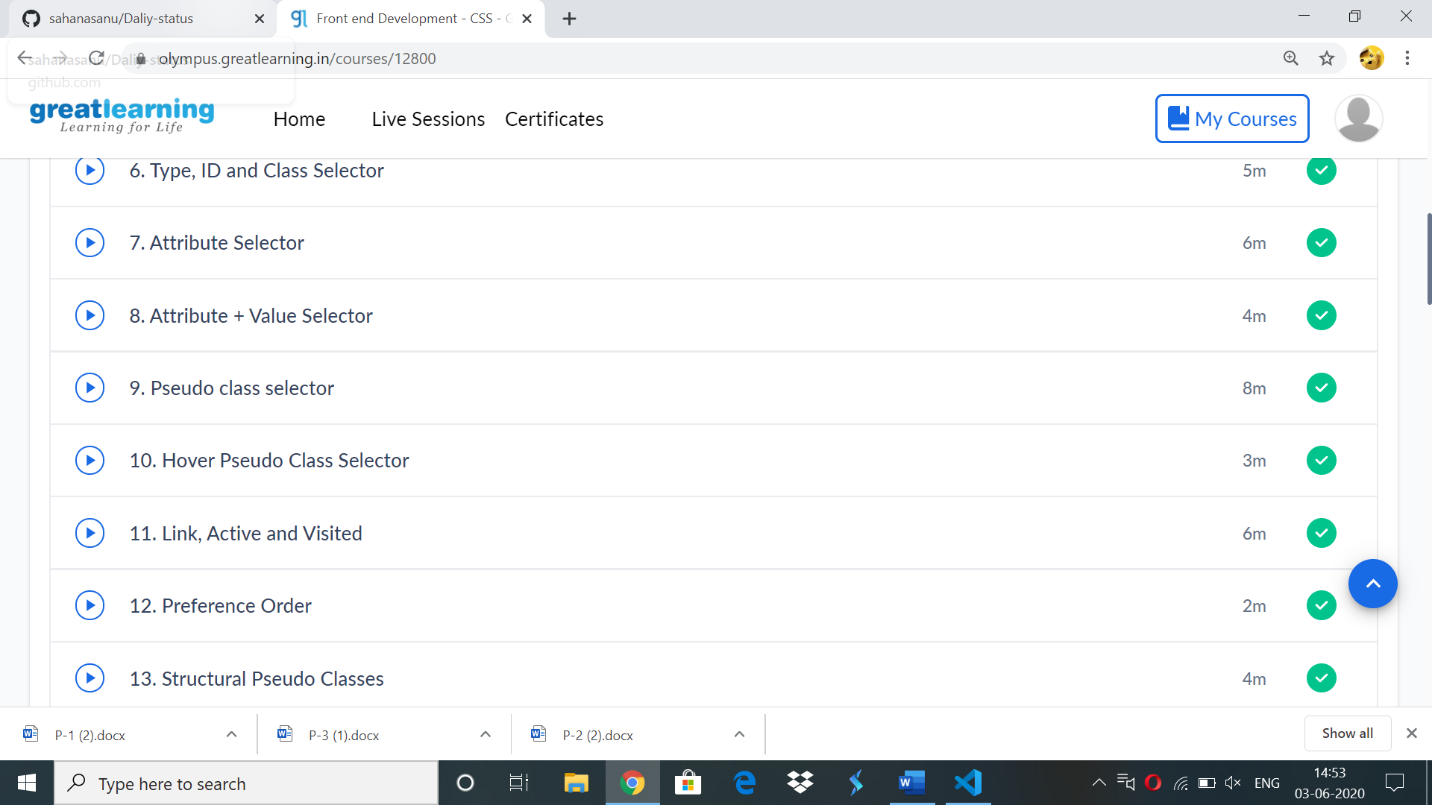
**DAILY ONLINE ACTIVITIES SUMMARY**

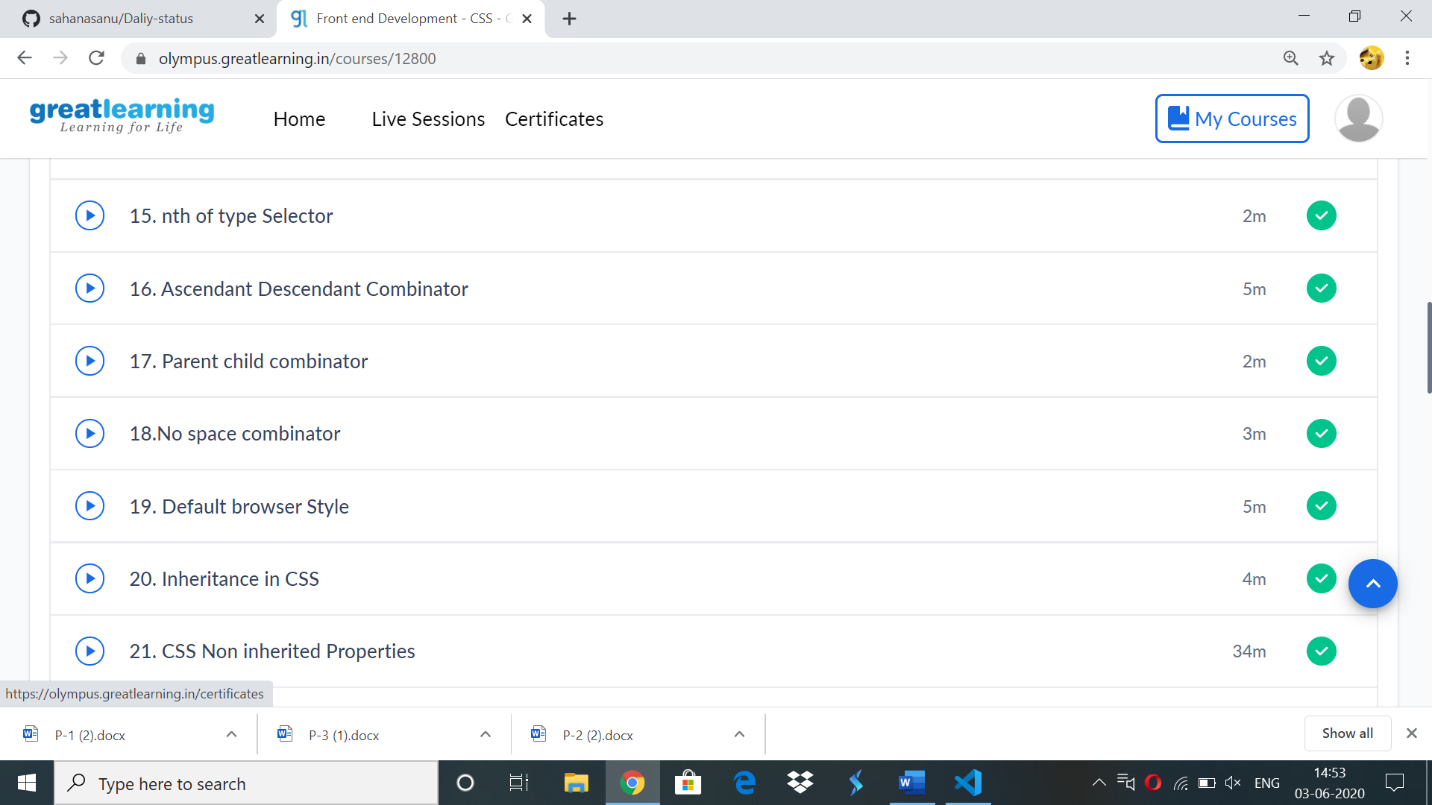
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **02-06-20** | | | | **Name:** | **SAHANA C** | |
| **Sem & Sec** | **VI B** | | | | **USN:** | **4AL17CS116** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **No test** | | | | | |
| **Max. Marks** | |  | | **Score** | |  | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **FRONT END DEVELOPMENT-CSS** | | | | | | |
| **Coding Challenges**  1)Python program to return a list containing first and last element using list slicing  2)Java program to check if given linked list has a loop or not.  3) C Program to find inversion count of array | | | | | | | |
| **Certificate Provider** | | | **Great learning** | **Duration** | | | **3 days** |
| **Status:Completed** | | | | | | | |
| **Uploaded the report in Github** | | | | **Yes** | | | |
| **If yes Repository name** | | | | **https://github.com/sahanasanu/Daliy-status** | | | |
| **Uploaded the report in slack** | | | | **Yes** | | | |

**IA MARKS DETAILS:no test**

**Online Certification Details:**

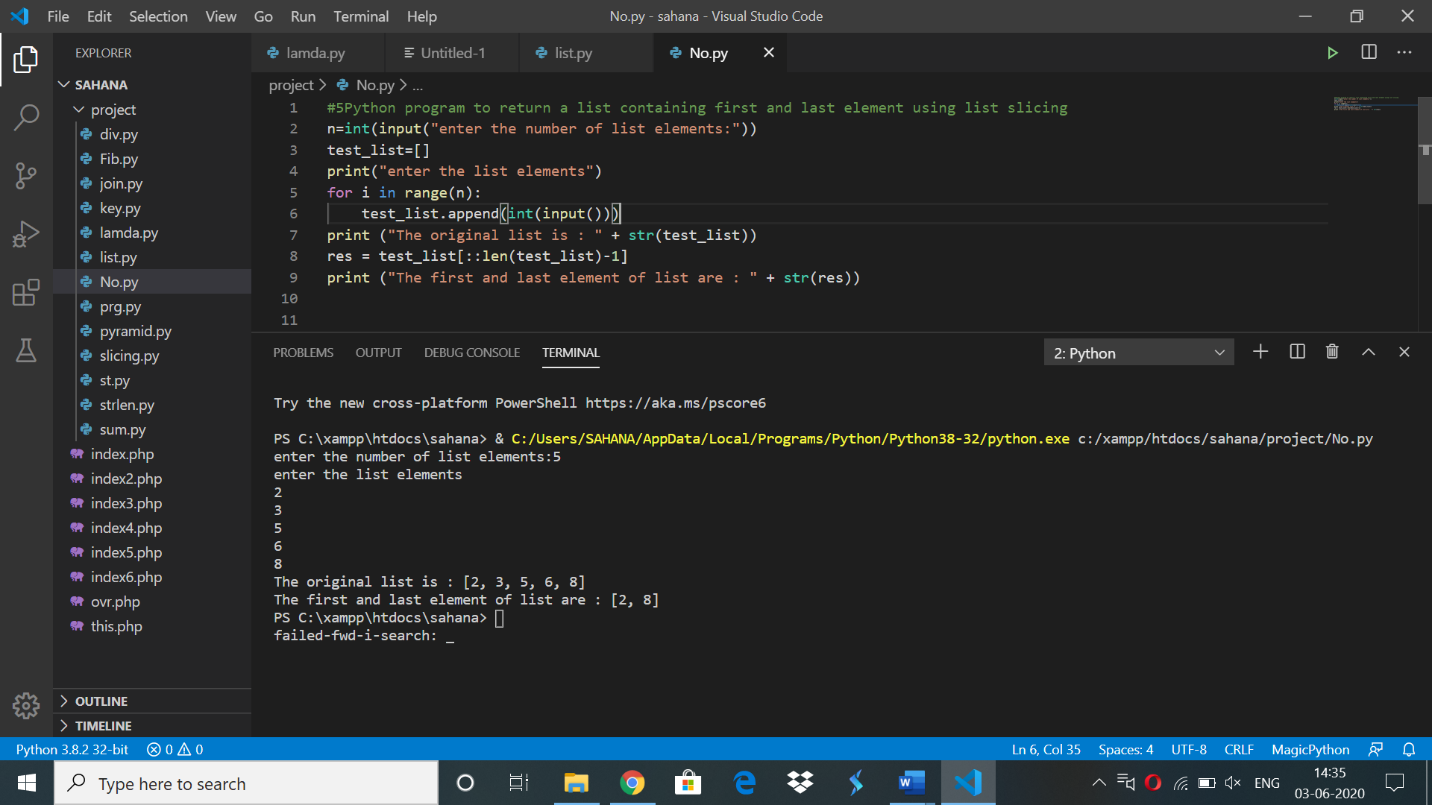
****

****

****

**Online coding:**

**1)**

****

**2)**

**Java**  program to check if given linked list has a loop or not.

**package** Tuesday;

**public** **class** LinkedList {

**private** Node head;

**private** **static** **class** Node {

**private** **int** value;

**private** Node next;

Node(**int** value) {

**this**.value = value;

}

}

**public** **void** addToTheLast(Node node) {

**if** (head == **null**) {

head = node;

} **else** {

Node temp = head;

**while** (temp.next != **null**)

temp = temp.next;

temp.next = node;

}

}

**public** **void** printList() {

Node temp = head;

**while** (temp != **null**) {

System.***out***.format("%d ", temp.value);

temp = temp.next;

}

System.***out***.println();

}

**public** **boolean** ifLoopExists() {

Node fastPtr = head;

Node slowPtr = head;

**while** (fastPtr != **null** && fastPtr.next != **null**) {

fastPtr = fastPtr.next.next;

slowPtr = slowPtr.next;

**if** (slowPtr == fastPtr)

**return** **true**;

}

**return** **false**;

}

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

LinkedList list = **new** LinkedList();

// Creating a linked list

Node loopNode=**new** Node(7);

list.addToTheLast(**new** Node(5));

list.addToTheLast(**new** Node(6));

list.addToTheLast(loopNode);

list.addToTheLast(**new** Node(1));

list.addToTheLast(**new** Node(2));

list.printList();

// creating a loop

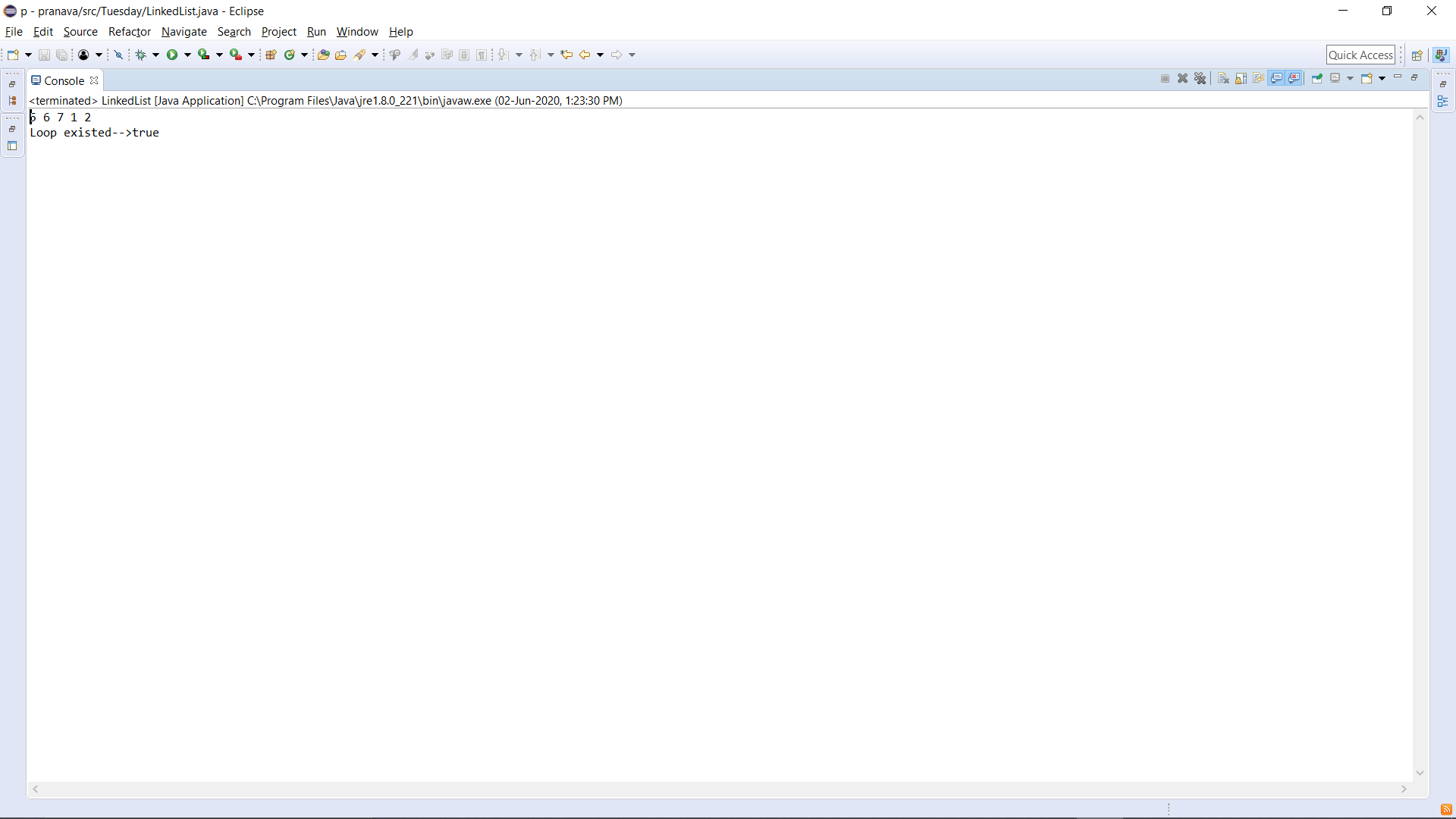
list.addToTheLast(loopNode);

// Test if loop existed or not

System.***out***.println("Loop existed-->" + list.ifLoopExists());

}

}



public static void main(String[] args) {

LinkedList list = new LinkedList();

// Creating a linked list

list.addToTheLast(new Node(5));

list.addToTheLast(new Node(6));

list.addToTheLast(new Node(7));

list.addToTheLast(new Node(1));

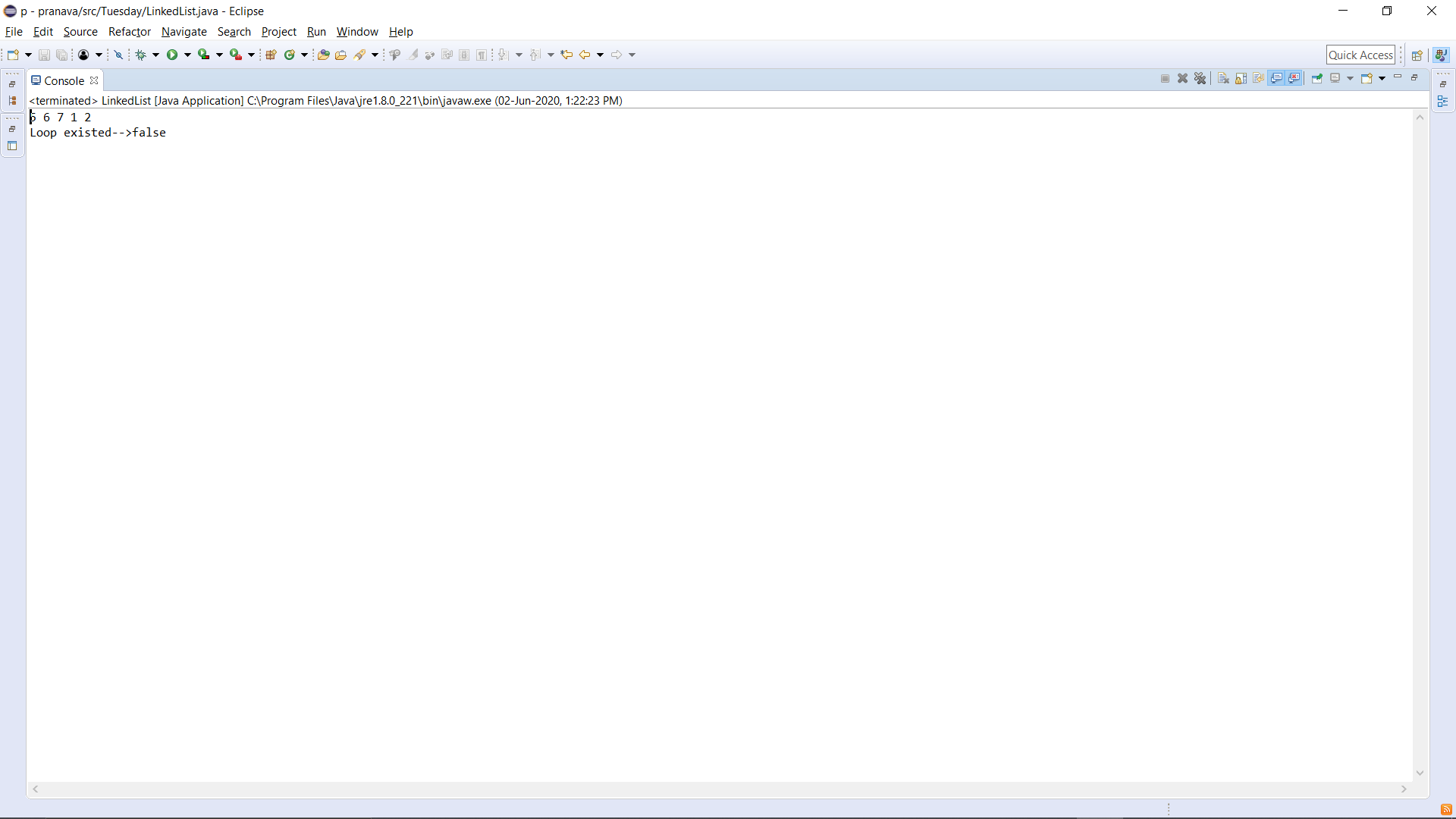
list.addToTheLast(new Node(2));

list.printList();

// Test if loop existed or not

System.out.println("Loop existed-->" + list.ifLoopExists());

}



3)

#include <stdio.h>

// Function to find Inversion count of the given array

int findInversionCount(int arr[], int n)

{

int inversionCount = 0,i,j;

for (i = 0; i < n - 1; i++)

{

for (j = i + 1; j < n; j++)

if (arr[i] > arr[j])

inversionCount++;

}

return inversionCount;

}

// main function

int main()

{

int n,arr[100],i;

printf("Enter the number of elements in the array\n");

scanf("%d",&n);

printf("Enter the elements in the array\n");

for(i=0;i<n;i++){

scanf("%d",&arr[i]);

}

printf("The inversion count of the array is %d", findInversionCount(arr, n));

return 0;

}

**OUTPUT**

