

## **DAILY ONLINE ACTIVITIES SUMMARY**

<b>Date:</b>	01-07-2020	<b>Name:</b>	Chethana j
<b>Sem &amp; Sec</b>	6 <sup>th</sup> & A	<b>USN:</b>	4al17cs022
<b>Online Test Summary</b>			
<b>Subject</b>			
<b>Max. Marks</b>	-	<b>Score</b>	-
<b>Pre-placement Training Summary</b>			
<b>Topic</b>			
<b>Faculty</b>		<b>Duration</b>	
<b>Coding Challenges</b>			
<b>Problem Statement:</b> 2 programs			
<b>Status:</b> Solved			
<b>Uploaded the report in Github</b>		yes	
<b>If yes Repository name</b>		<a href="https://github.com/Jchethana1990/online-course">https://github.com/Jchethana1990/online-course</a> <a href="https://github.com/Jchethana1990/Machine-learning-workshop">https://github.com/Jchethana1990/Machine-learning-workshop</a>	
<b>Uploaded the report in slack</b>		yes	

# ONLINE CODING

## 1. Python Program to Create a Linked List & Display the Elements in the Lists

```
class Node:

    def __init__(self, data):

        self.data = data

        self.next = None

class LinkedList:

    def __init__(self):

        self.head = None

        self.last_node = None

    def append(self, data):

        if self.last_node is None:

            self.head = Node(data)

            self.last_node = self.head

        else:

            self.last_node.next = Node(data)

            self.last_node = self.last_node.next

    def display(self):

        current = self.head

        while current is not None:

            print(current.data, end = ' ')

            current = current.next

a_llist = LinkedList()

n = int(input('Enter the elements would you like to add: '))
```

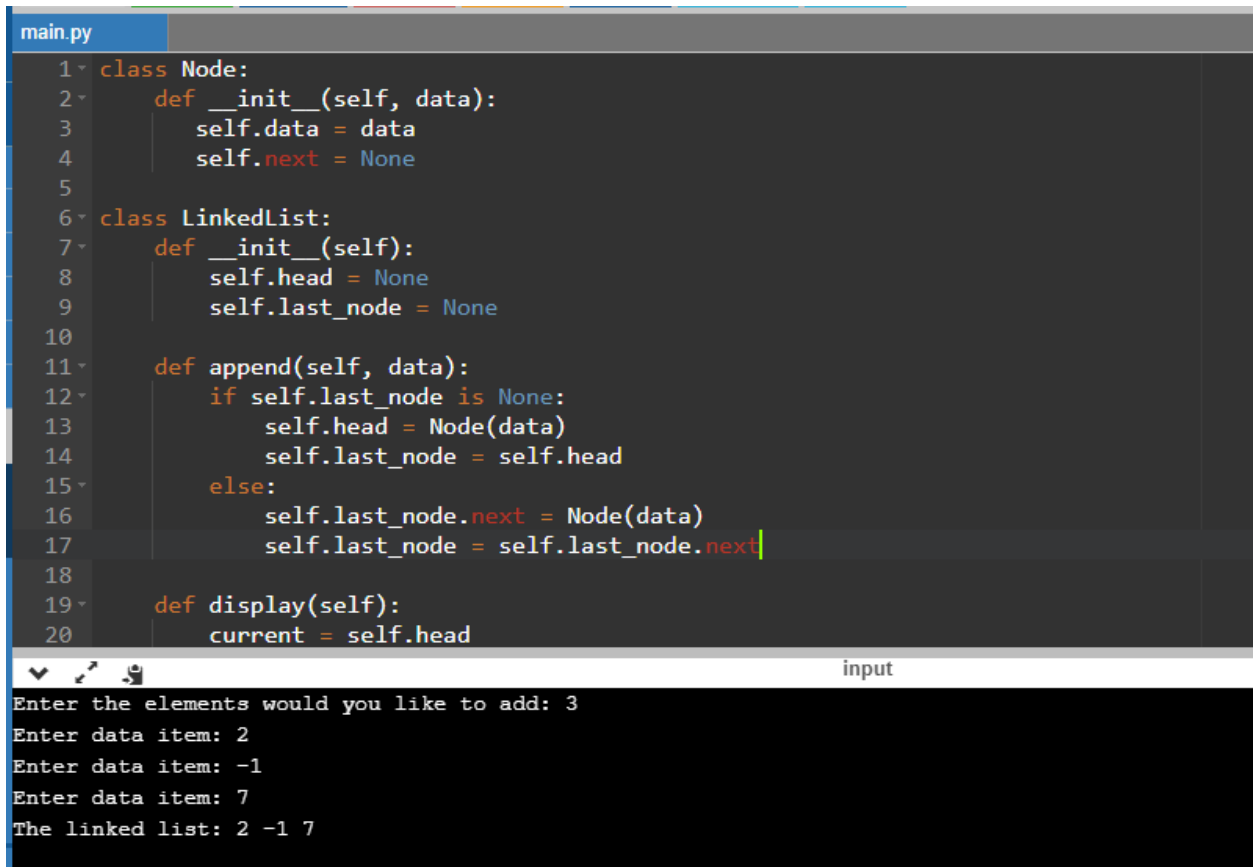
```
for i in range(n):
```

```
    data = int(input('Enter data item: '))
```

```
    a_llist.append(data)
```

```
print("The linked list: ", end = ")
```

```
a_llist.display()
```



The screenshot shows a Python IDE with a file named 'main.py'. The code defines two classes: 'Node' and 'LinkedList'. The 'Node' class has an '\_\_init\_\_' method that takes 'data' as an argument and sets 'self.data' and 'self.next' (to None). The 'LinkedList' class has an '\_\_init\_\_' method that sets 'self.head' and 'self.last\_node' to None. It also has an 'append' method that adds a new node to the end of the list, and a 'display' method that prints the elements of the list. The execution output shows the user entering 3 elements (2, -1, 7) and the program printing 'The linked list: 2 -1 7'.

```
1 class Node:
2     def __init__(self, data):
3         self.data = data
4         self.next = None
5
6 class LinkedList:
7     def __init__(self):
8         self.head = None
9         self.last_node = None
10
11     def append(self, data):
12         if self.last_node is None:
13             self.head = Node(data)
14             self.last_node = self.head
15         else:
16             self.last_node.next = Node(data)
17             self.last_node = self.last_node.next
18
19     def display(self):
20         current = self.head
```

input

Enter the elements would you like to add: 3  
Enter data item: 2  
Enter data item: -1  
Enter data item: 7  
The linked list: 2 -1 7

## 2. Write a program to find given two trees are mirror or not.

```
class Node
```

```
{
```

```
int data;
```

```
Node left, right;
```

```

public Node(int data)
{
    this.data = data;
    left = right = null;
}

}

public class BinaryTree
{
    Node a, b;

    boolean areMirror(Node a, Node b)
    {
        if (a == null && b == null)
            return true;

        if (a == null || b == null)
            return false;

        return a.data == b.data
            && areMirror(a.left, b.right)
            && areMirror(a.right, b.left);
    }

    public static void main(String[] args)
    {
        BinaryTree tree = new BinaryTree();

        Node a = new Node(1);
        Node b = new Node(1);

        a.left = new Node(2);
        a.right = new Node(3);
        a.left.left = new Node(4);
    }
}

```

```
a.left.right = new Node(5);
```

```
b.left = new Node(3);
```

```
b.right = new Node(2);
```

```
b.right.left = new Node(5);
```

```
b.right.right = new Node(4);
```

```
if (tree.areMirror(a, b) == true)
```

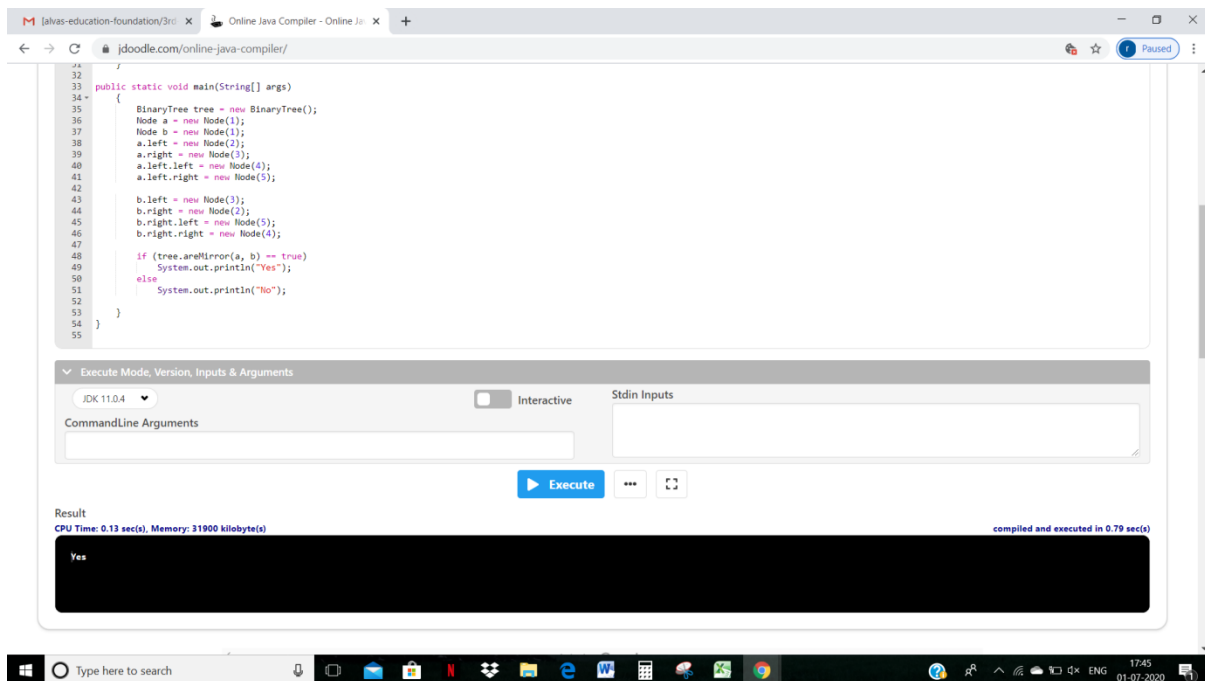
```
System.out.println("Yes");
```

```
else
```

```
System.out.println("No");
```

```
}
```

```
}
```



```
32  
33 public static void main(String[] args)  
34 {  
35     BinaryTreeNode tree = new BinaryTreeNode();  
36     Node a = new Node(1);  
37     Node b = new Node(1);  
38     a.left = new Node(2);  
39     a.right = new Node(3);  
40     a.left.left = new Node(4);  
41     a.left.right = new Node(5);  
42  
43     b.left = new Node(3);  
44     b.right = new Node(2);  
45     b.right.left = new Node(5);  
46     b.right.right = new Node(4);  
47  
48     if (tree.areMirror(a, b) == true)  
49         System.out.println("Yes");  
50     else  
51         System.out.println("No");  
52  
53 }  
54  
55 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Interactive

Stdin Inputs

CommandLine Arguments

Execute

Result

CPU Time: 0.13 sec(s), Memory: 31900 kilobyte(s)

compiled and executed in 0.79 sec(s)

Yes