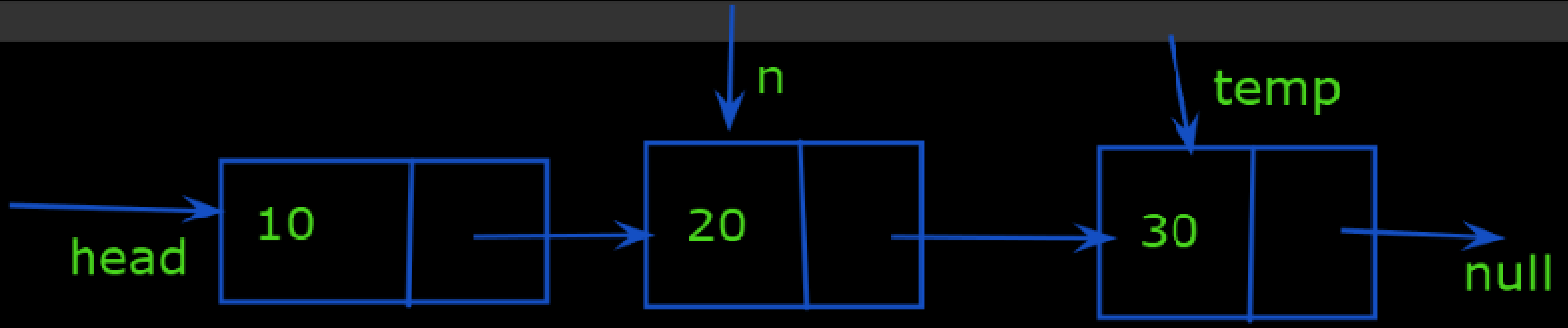
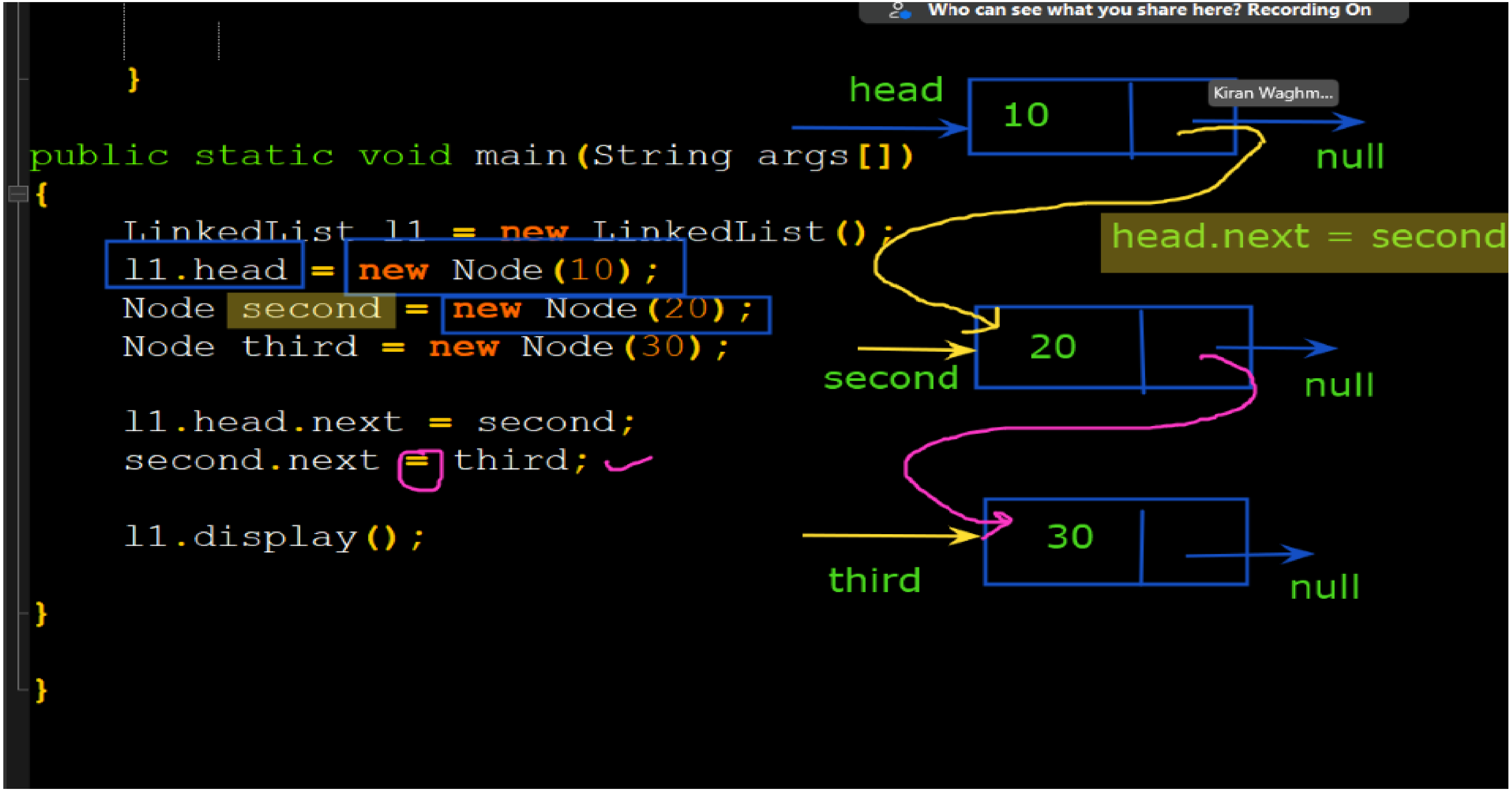
Algorithms & Data Structure
Day 4: Linked List

Kiran Waghmare



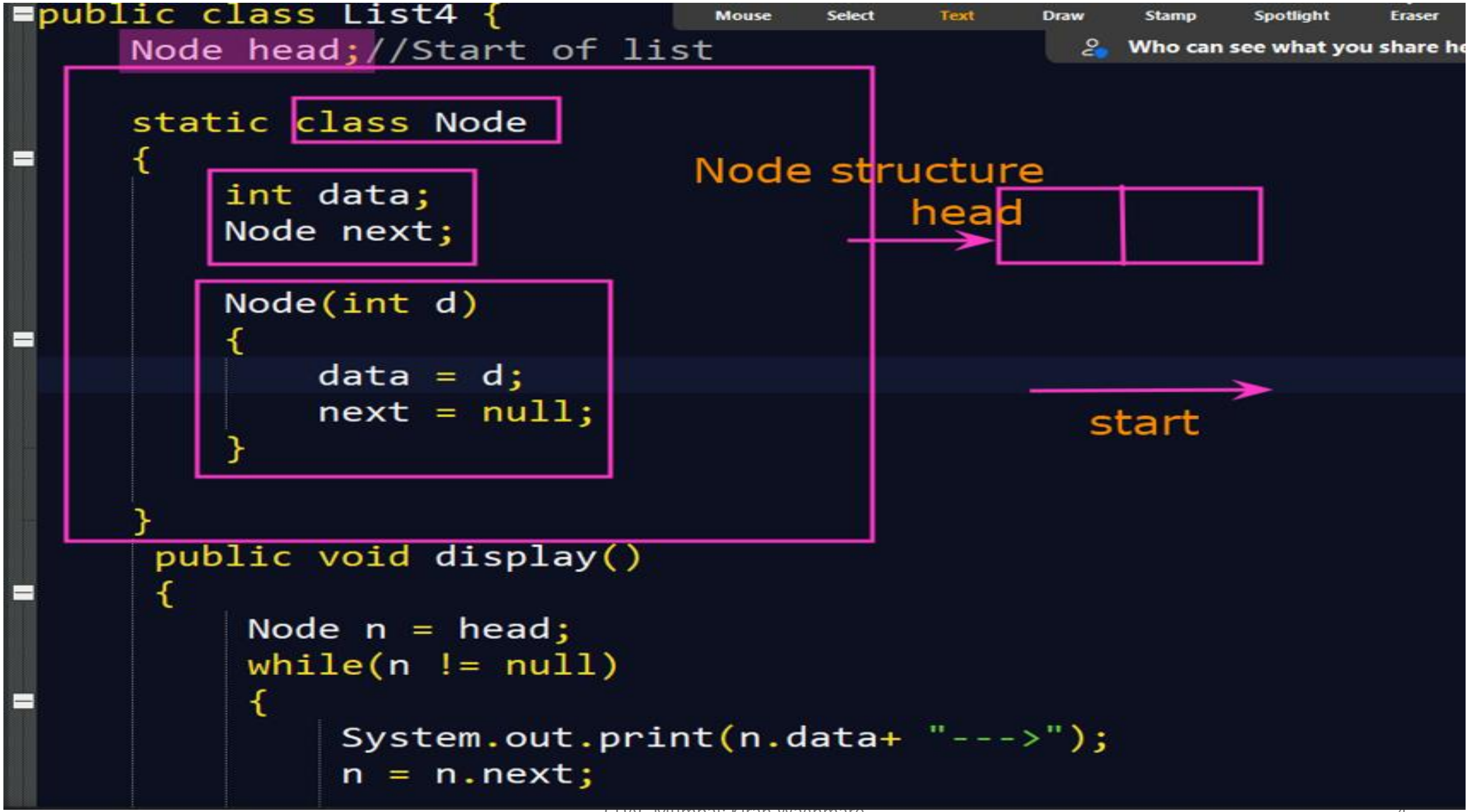
```
Node head, n;
head=n;

n=head.next

int num = temp.data

num=30
```

```
n=n.next

head=n.next

temp=n.next

head.next.next.data
n.next.data
```

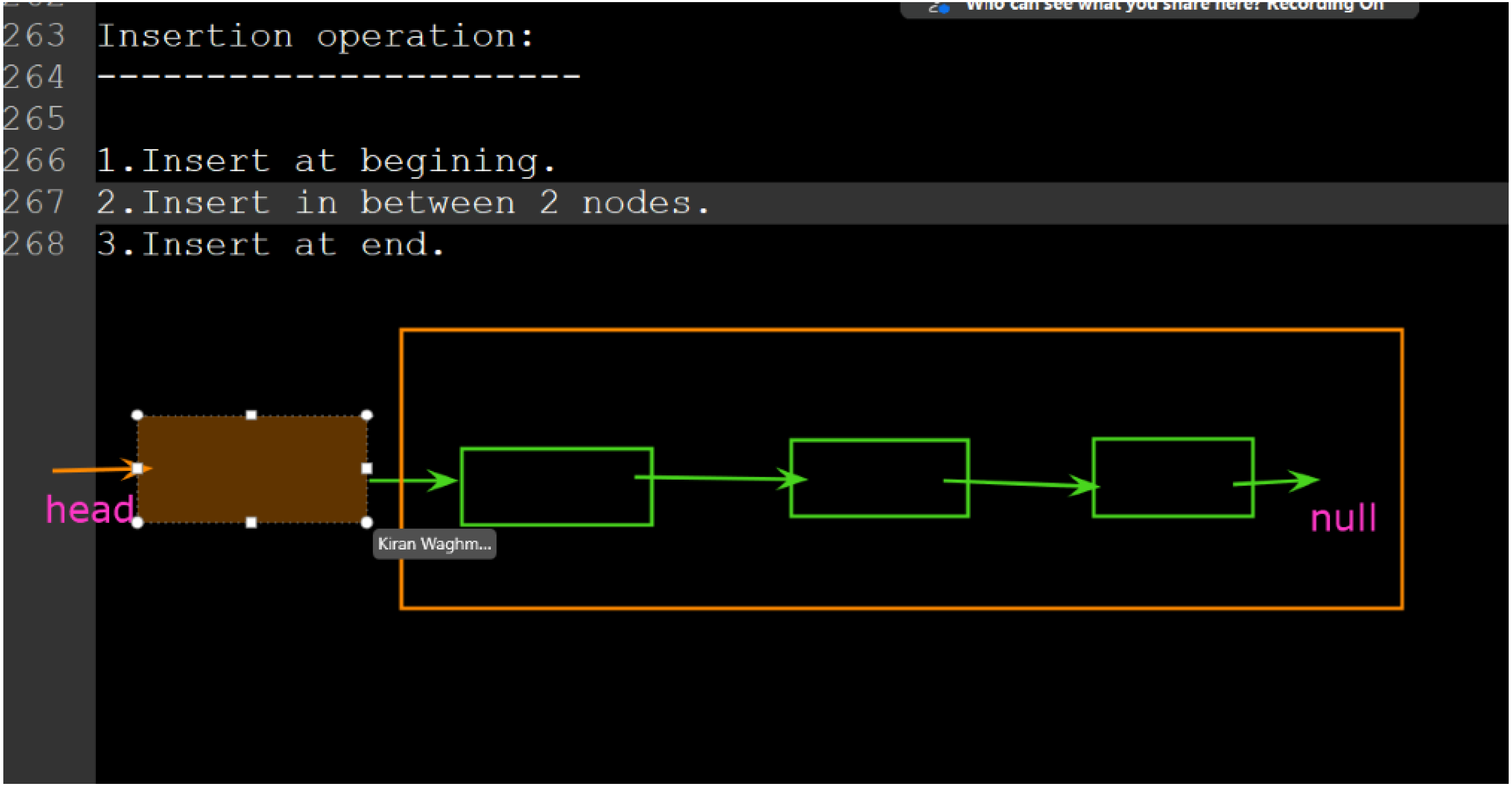


```
    }

public static void main(String args[])
{
    LinkedList l1 = new LinkedList();
    l1.head = new Node(10);
    Node second = new Node(20);
    Node third = new Node(30);

    l1.head.next = second;
    second.next = third;

    l1.display();

}

}
```

head.next = second



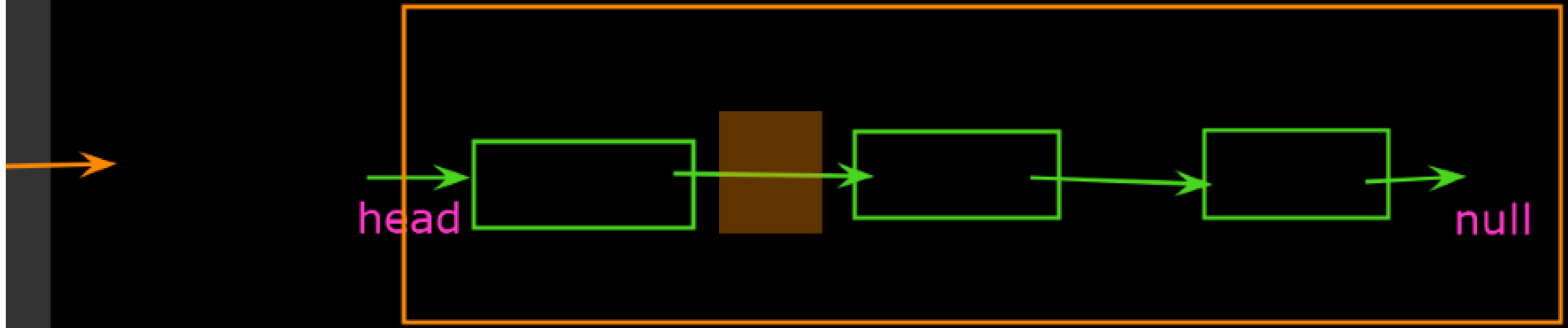
```
public class List4 {
    Node head;//Start of list

    static class Node
    {
        int data;
        Node next;

        Node(int d)
        {
            data = d;
            next = null;
        }

    }
     public void display()
     {
         Node n = head;
         while(n != null)
         {
             System.out.print(n.data+ "--->");
             n = n.next;
```

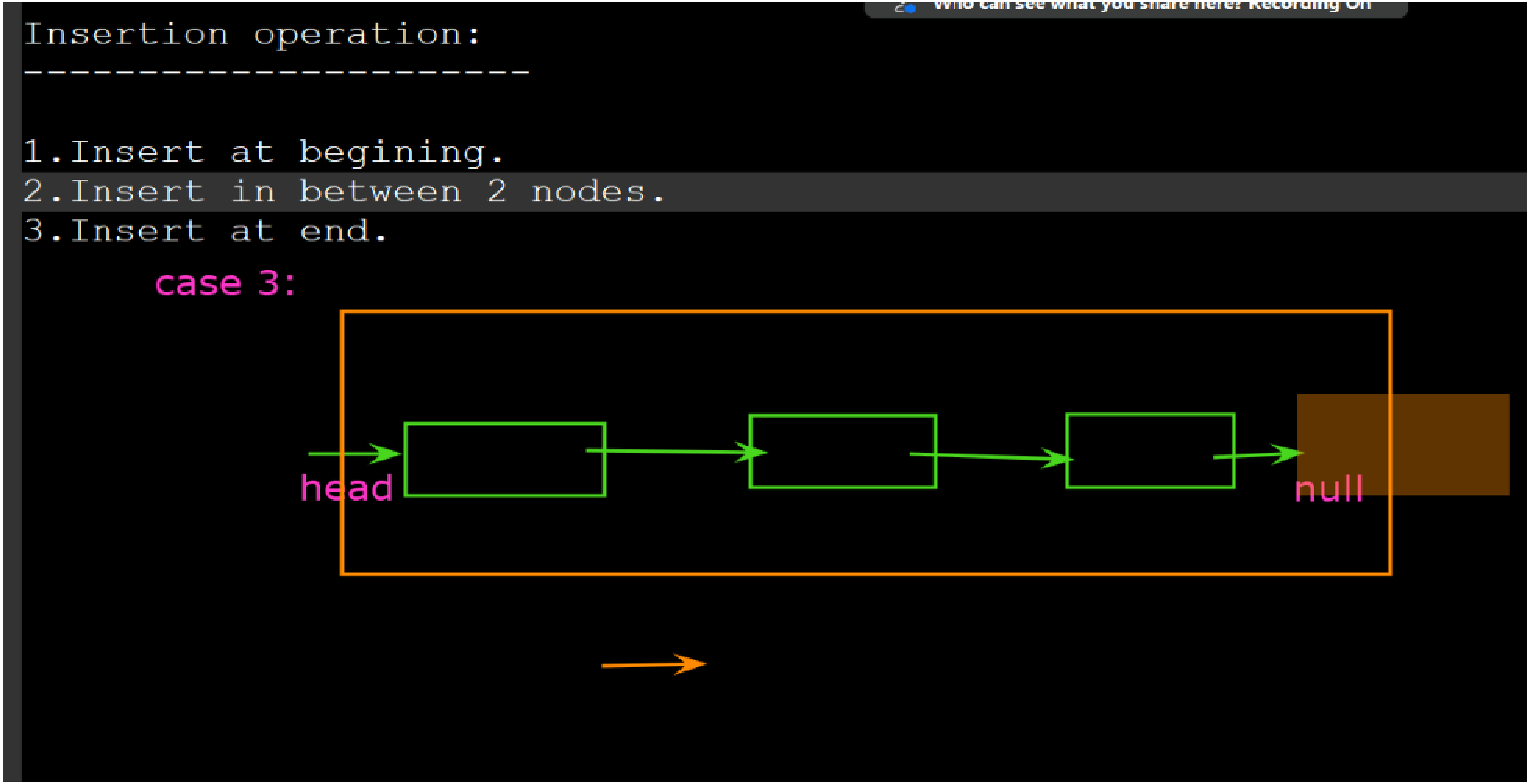


```
Insertion operation:
-----------------------

1.Insert at begining.
2.Insert in between 2 nodes.
3.Insert at end.
```

```
Insertion operation:
----------------------

1.Insert at begining.
2.Insert in between 2 nodes.
3.Insert at end.
```

case 2:

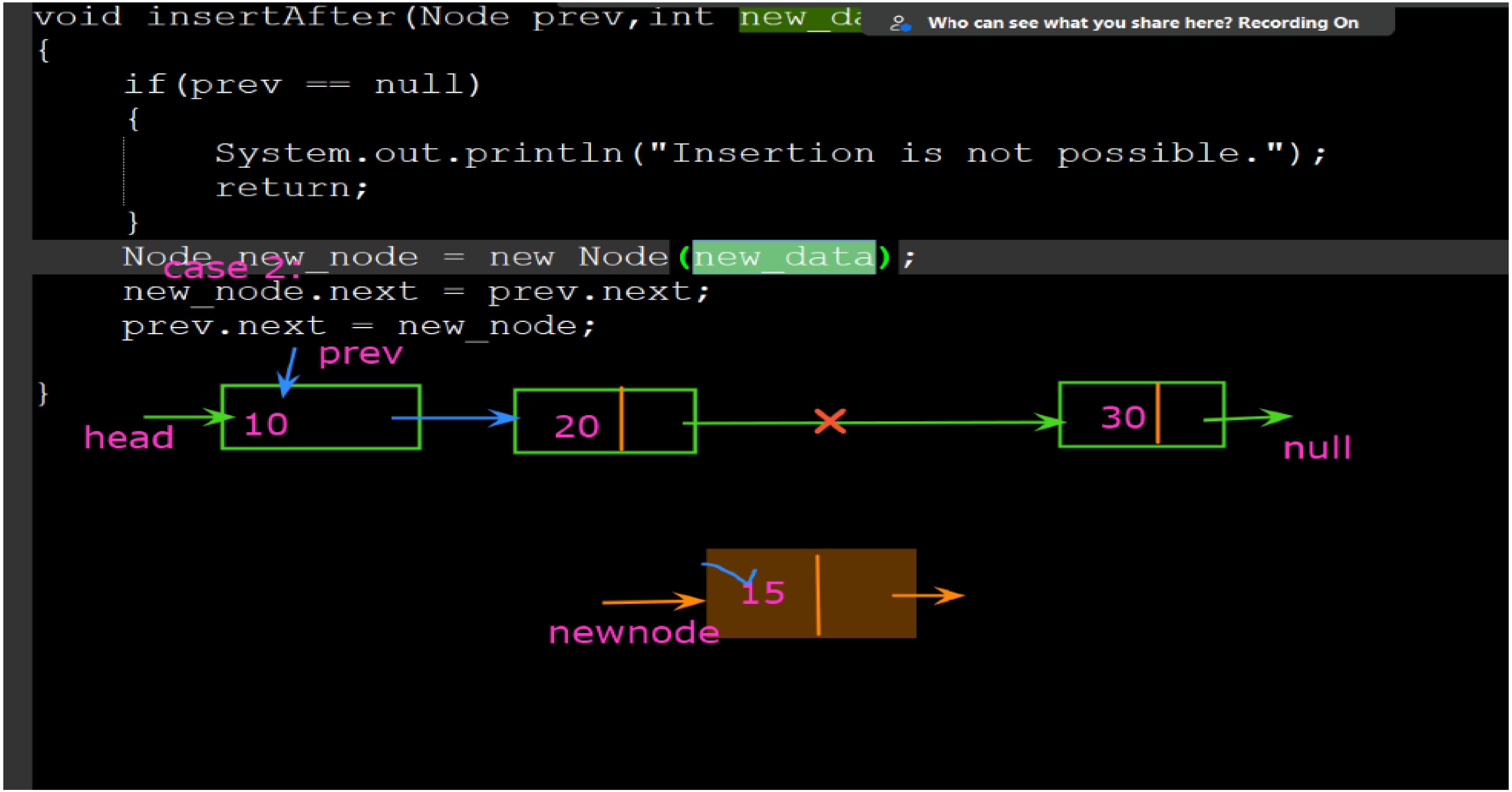




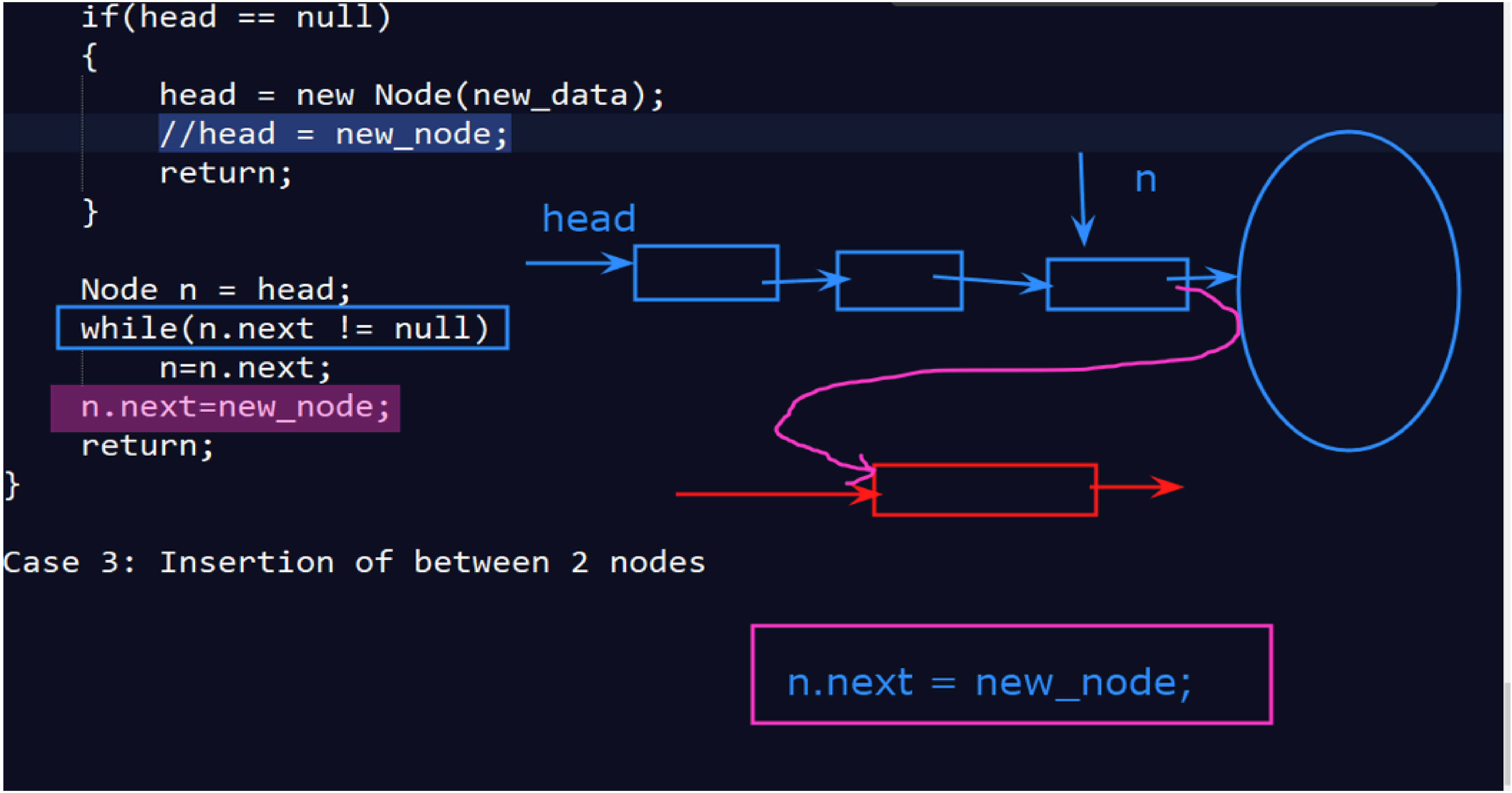
Who can see what you share here? Recording On

```
Insertion operation:
------------------------

1.Insert at begining.
2.Insert in between 2 nodes.
3.Insert at end.
```

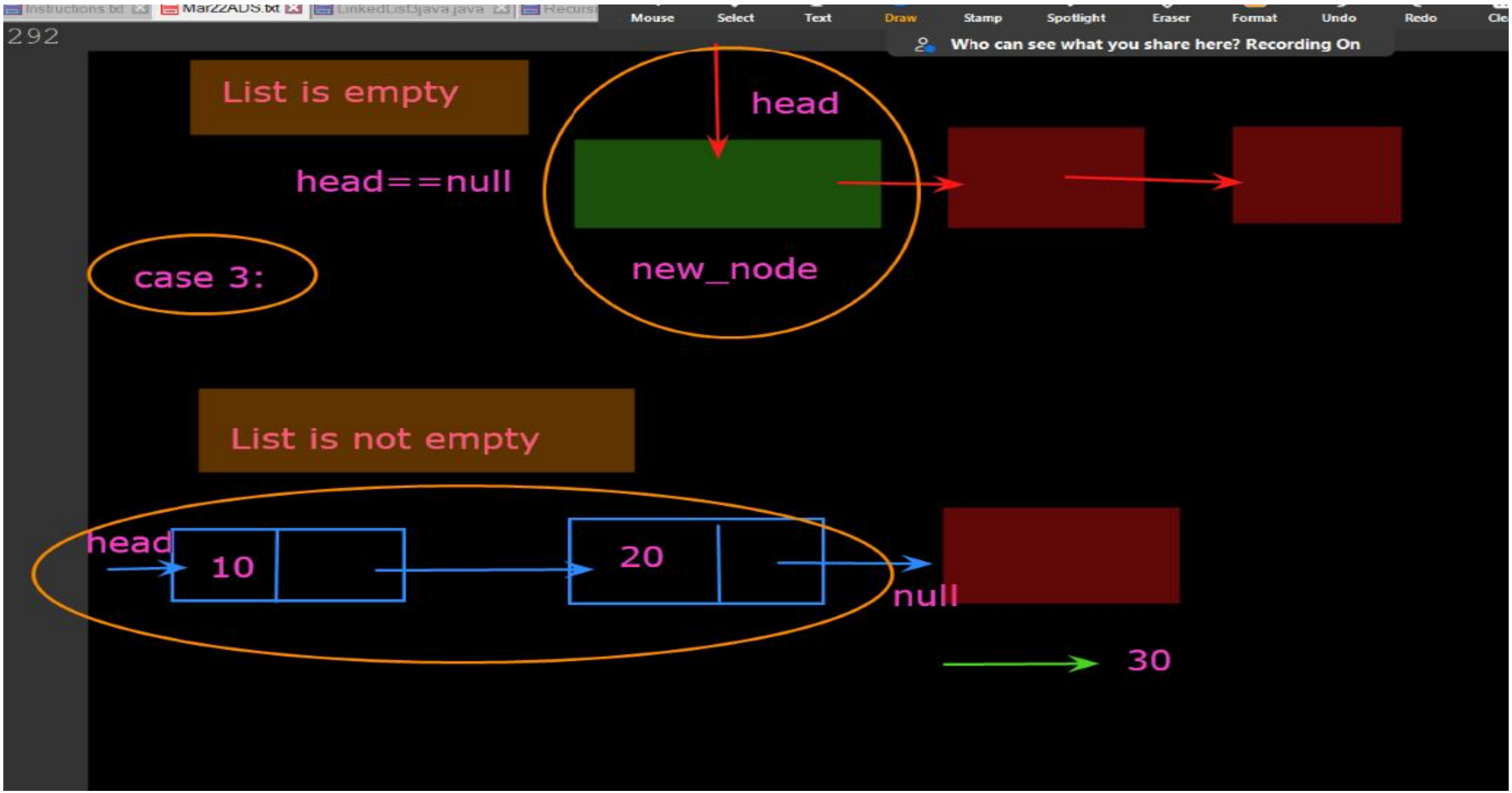
case 3:



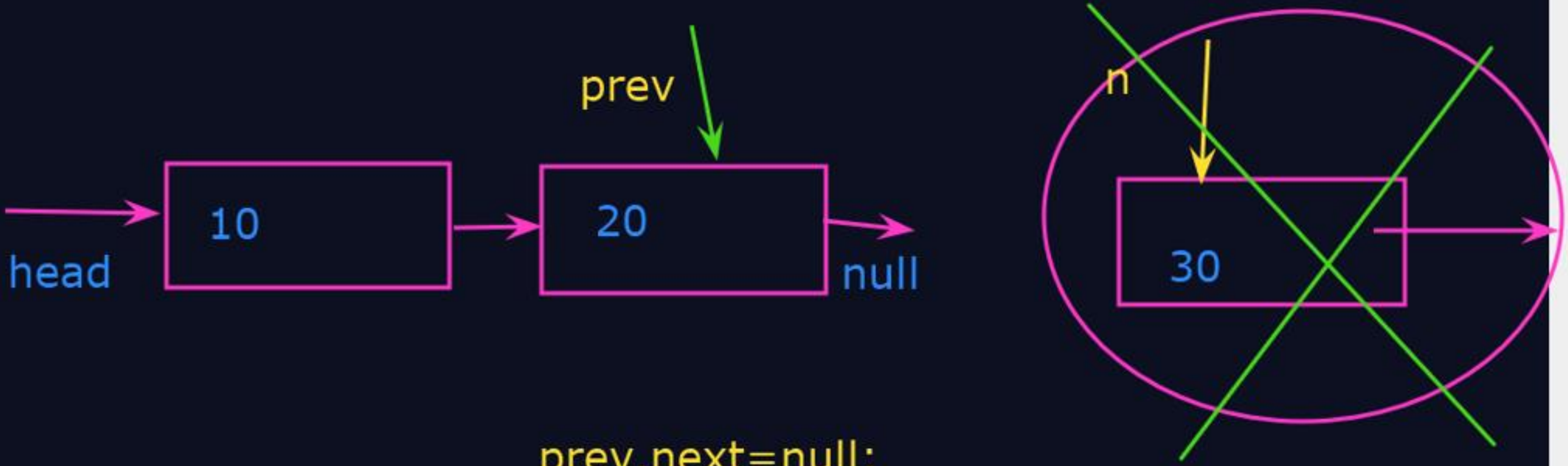
```
void insertAfter(Node prev,int new_da
{
     if(prev == null)
     {
          System.out.println("Insertion is not possible.");
          return;
     }
     Node new_node = new Node(new_data);
     new_node.next = prev.next;
     prev.next = new_node;

}
```



```
    if(head == null)
    {
        head = new Node(new_data);
        //head = new_node;
        return;
    }

    Node n = head;
    while(n.next != null)
        n=n.next;
    n.next=new_node;
    return;
}

Case 3: Insertion of between 2 nodes
```

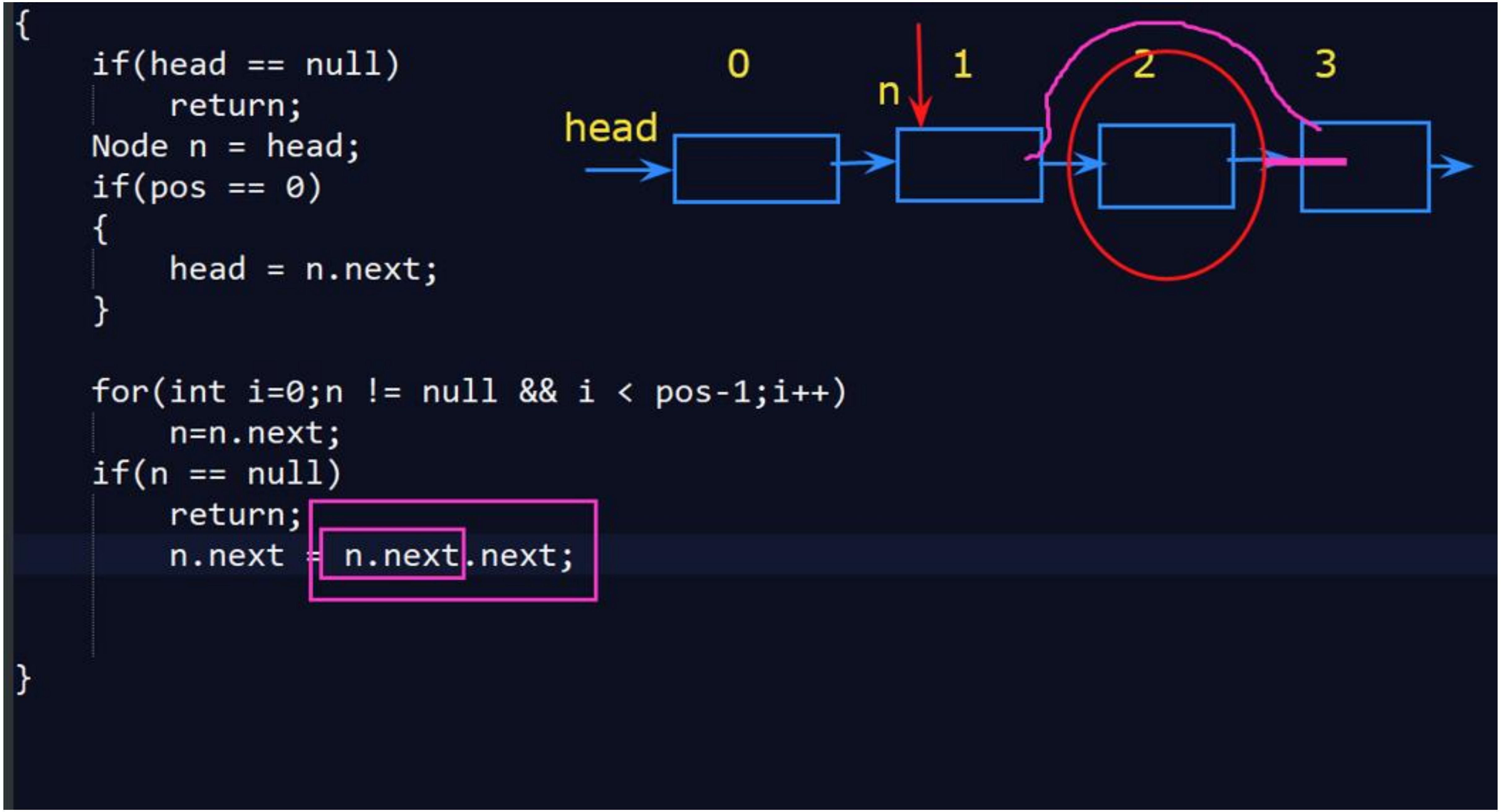

Instructions.txt
Mar22ADS.txt
LinkedList3java.java
Recurs
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Cle
Who can see what you share here? Recording On
292
List is empty
head
head==null
case 3:
new_node
List is not empty
head
10
20
null
30

-Linked List Operations
    -Delete

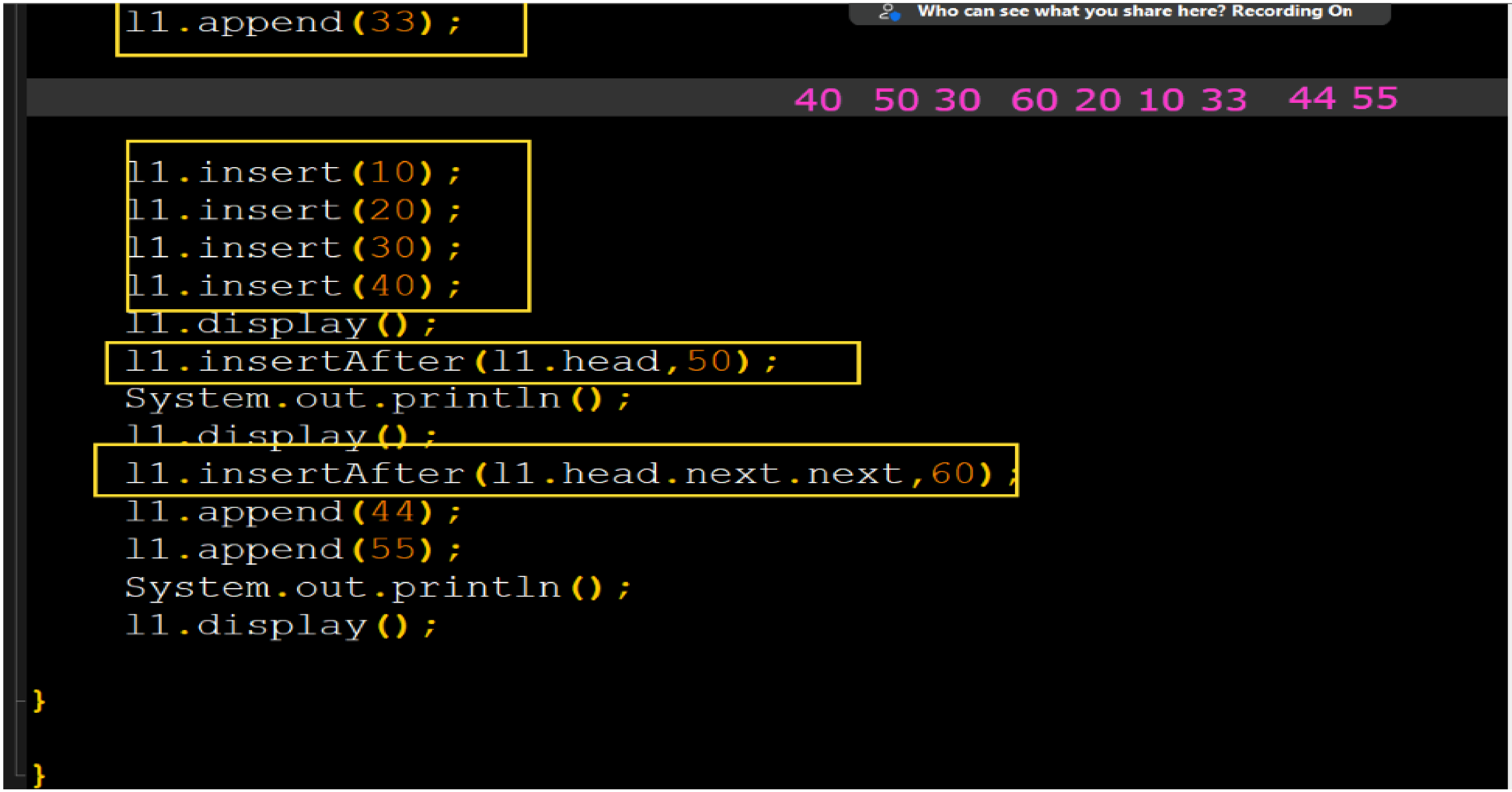


prev.next=null;

prev.next = n.next;



```
{
    if(head == null)
        return;
    Node n = head;
    if(pos == 0)
    {
        head = n.next;
    }

    for(int i=0;n != null && i < pos-1;i++)
        n=n.next;
    if(n == null)
        return;
        n.next = n.next.next;


}
```



```
    l1.append(33);
```

```
40  50 30  60 20 10 33  44 55
```

```
    l1.insert(10);
    l1.insert(20);
    l1.insert(30);
    l1.insert(40);
    l1.display();
    l1.insertAfter(l1.head,50);
    System.out.println();
    l1.display();
    l1.insertAfter(l1.head.next.next,60);
    l1.append(44);
    l1.append(55);
    System.out.println();
    l1.display();

}

}
```

Who can see what you share here? Recording On

```
          null

}

void delete(int key)
{
    Node temp = head, prev =null
    if(temp.data == key && temp !=null)
    {                    20
        head = temp.next;
        return;
    }
```

case 1

```
    while(temp !=null && temp.data != key)
    {
        prev = temp;
        temp = temp.next;
    }
```

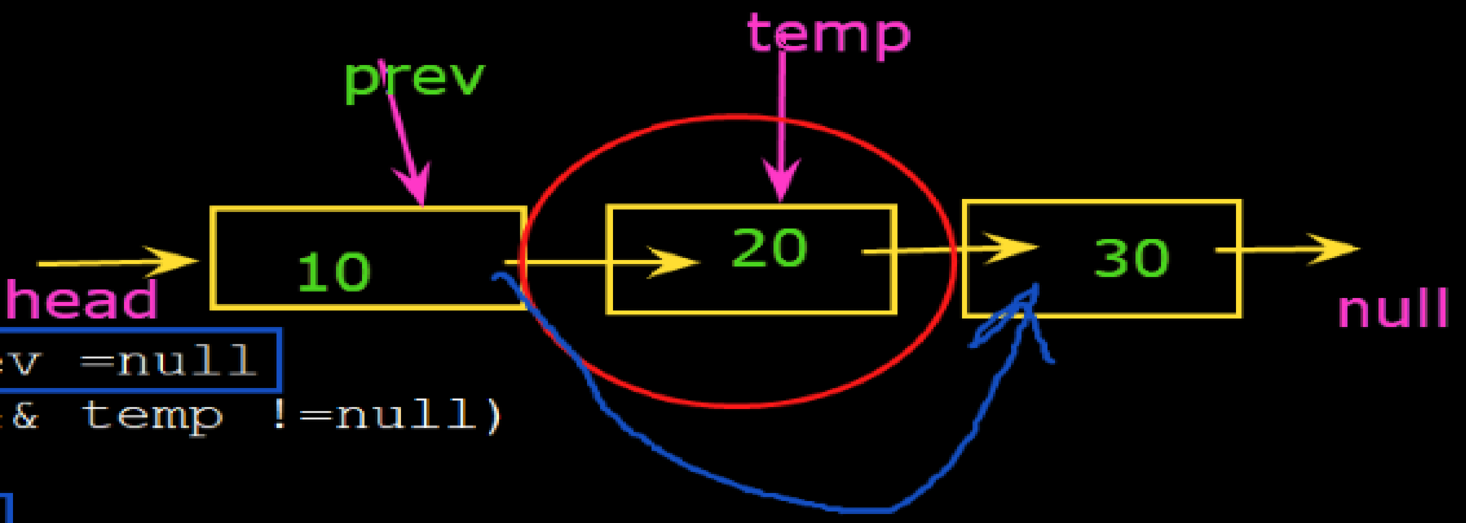
case 2

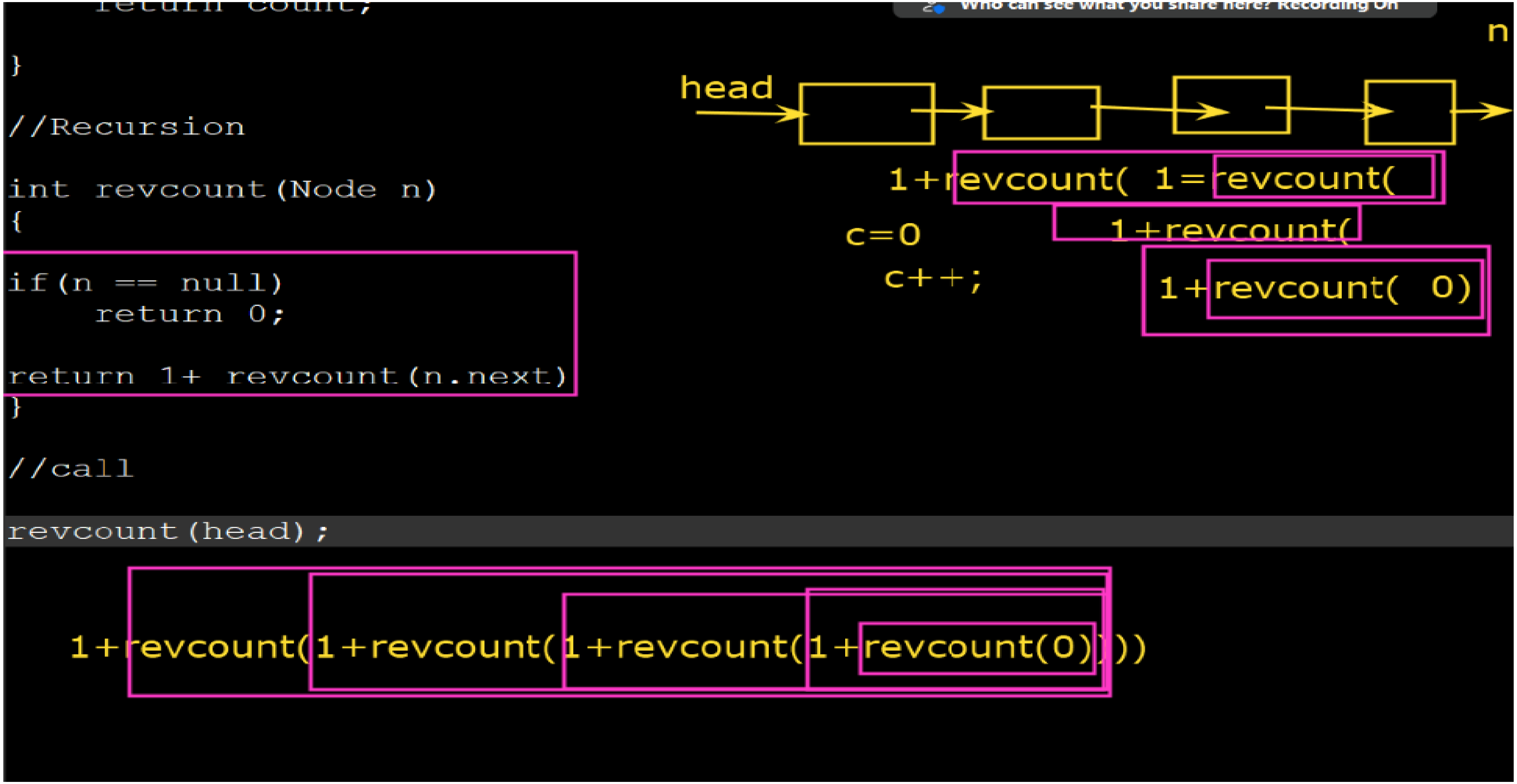
```
    if(temp==null)
        return;
```

case 3

```
    prev.next = temp.next;
}
```

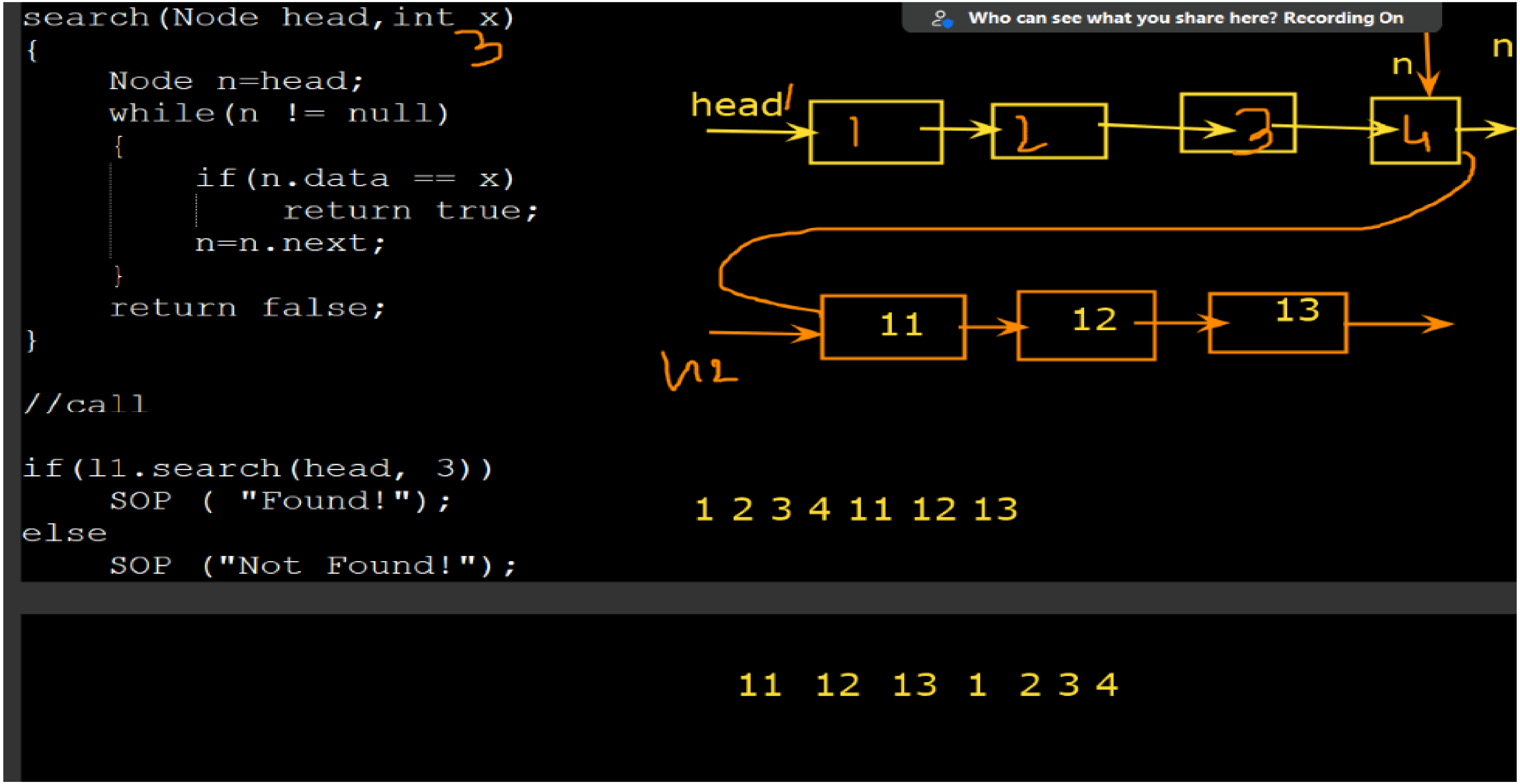
$\rightarrow$ connection establish





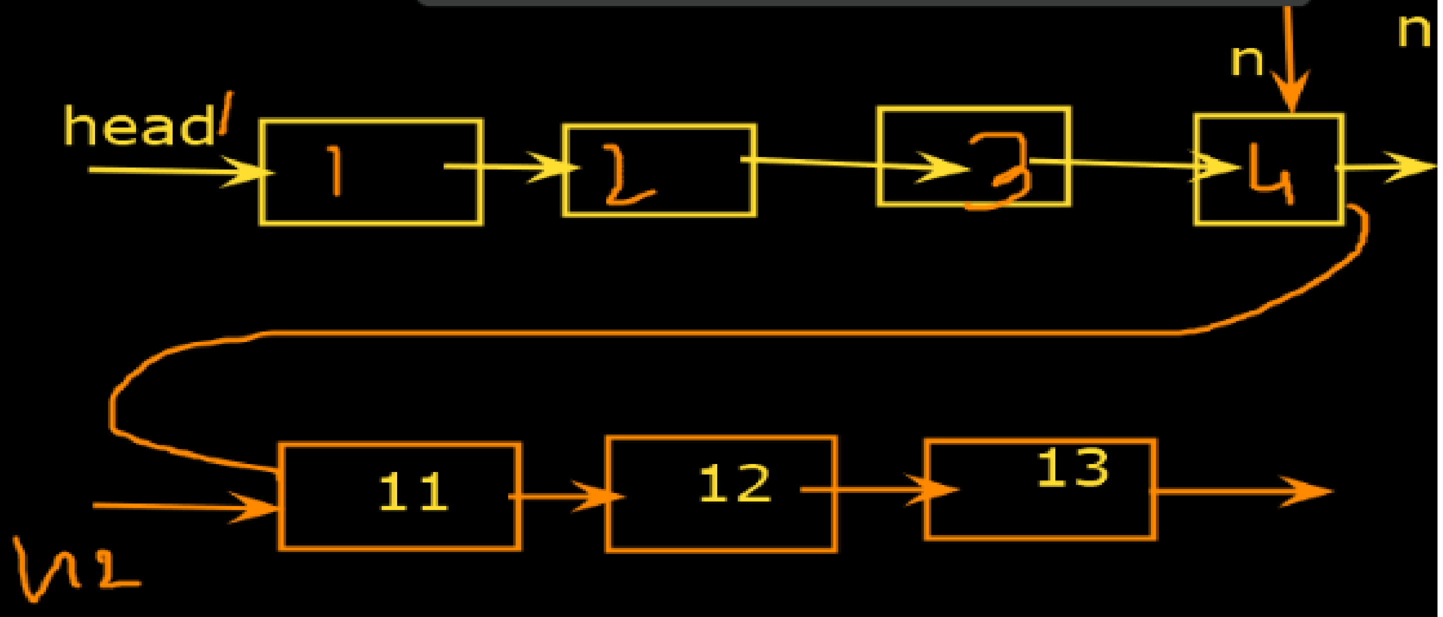
```
    return count;

}

//Recursion

int revcount(Node n)
{

if(n == null)
    return 0;

return 1+ revcount(n.next)
}

//call

revcount(head);
```

$1+\boxed{revcount(\boxed{1+revcount(\boxed{1+revcount(\boxed{1+\boxed{revcount(0)}})})})}$



Who can see what you share here? Recording On

```
search(Node head,int x)
{
     Node n=head;
     while(n != null)
     {
          if(n.data == x)
               return true;
          n=n.next;
     }
     return false;
}

//call

if(l1.search(head, 3))
     SOP ( "Found!");
else
     SOP ("Not Found!");
```



1 2 3 4 11 12 13

11 12 13 1 2 3 4

Thanks