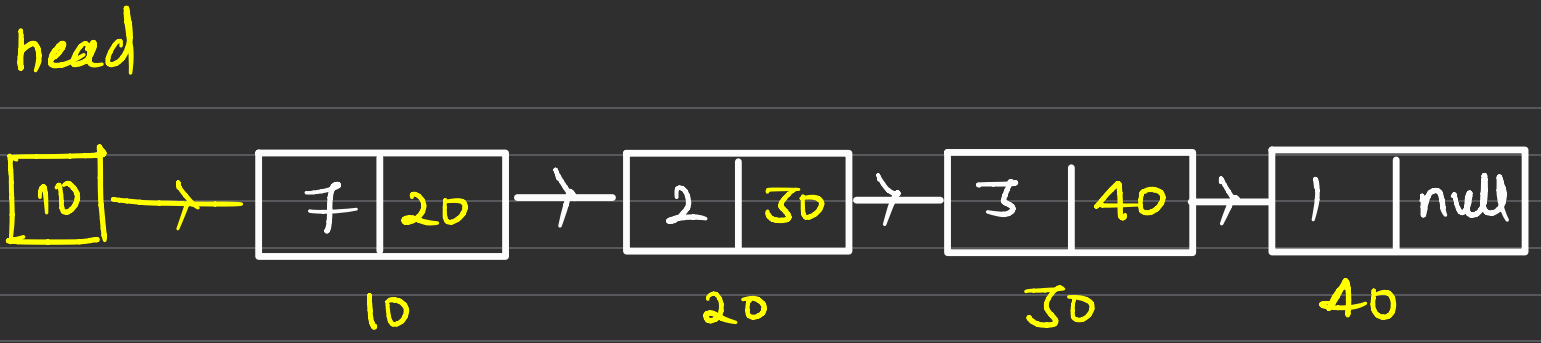
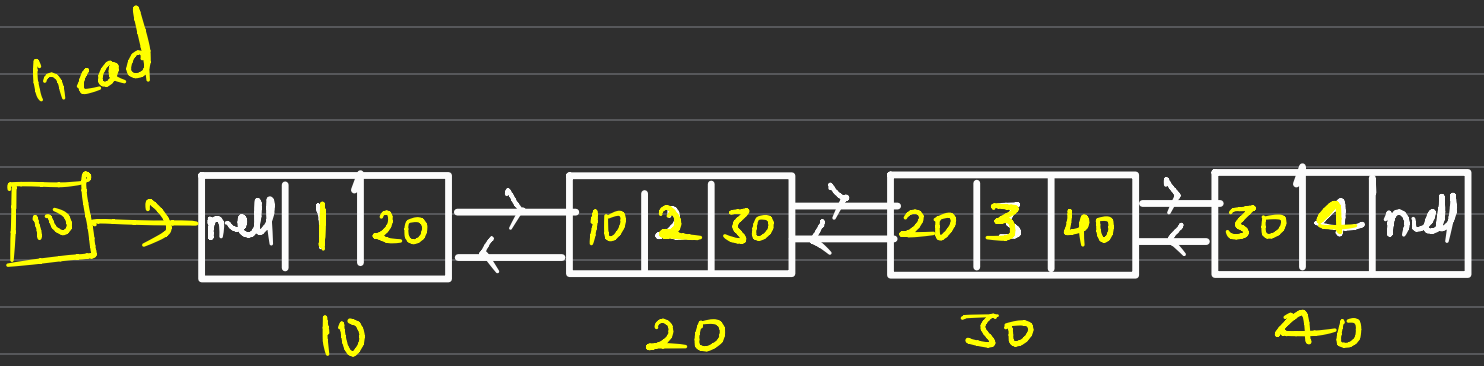
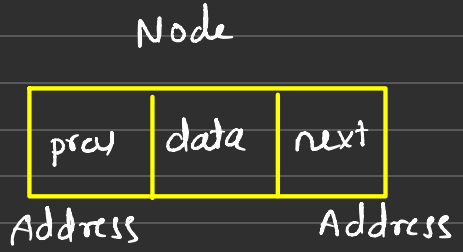


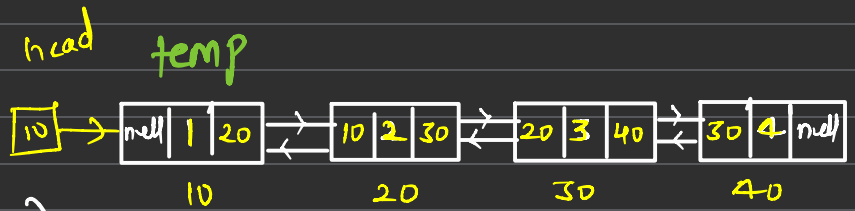



# Structure of Node in DLL

```
class Node
{
    int data
    Node * next
    Node * prev
};
```



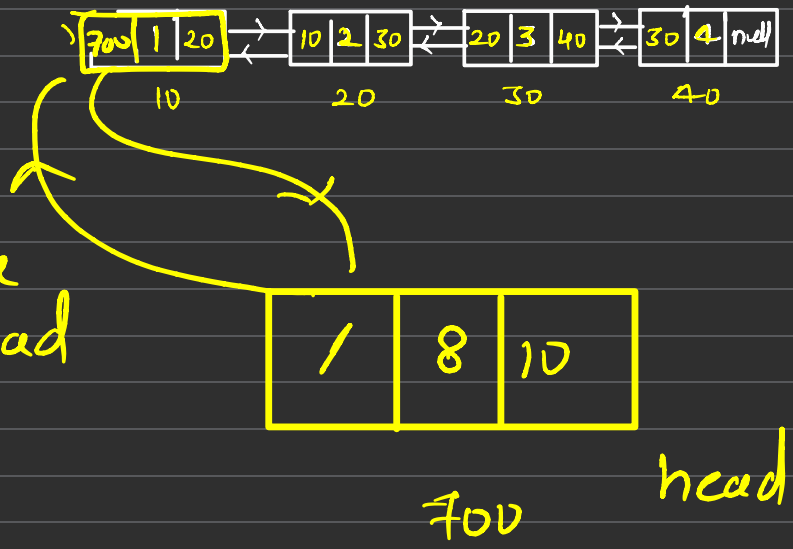
# * Traversal in DLL



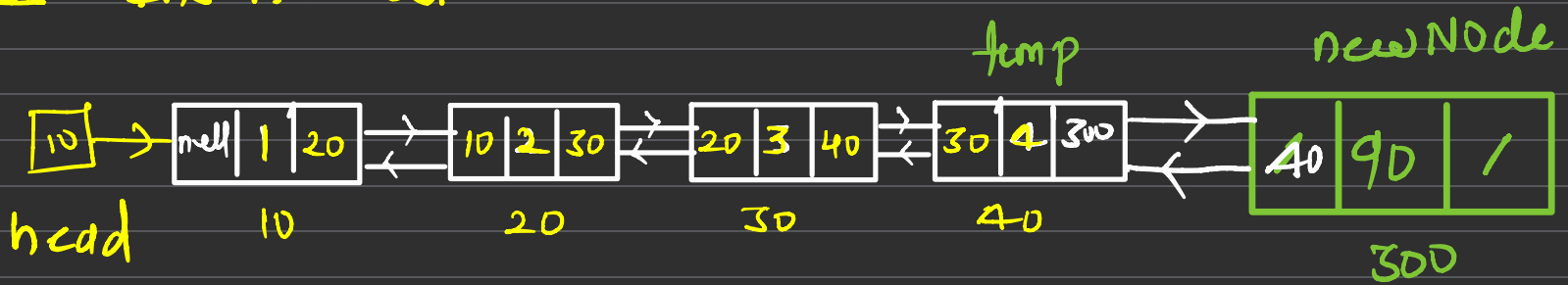

```
void Traverse(head)
{
    Node temp = head
    while (temp != null)
    {
        print(temp → data)
        temp = temp → next
    }
}
```

* Insertion operations → start

```
if (head == null)
{
    head = newNode
}
else{

    head → prev = newNode
    newNode → next = head
    head = newNode

}
```



# Insert at end:

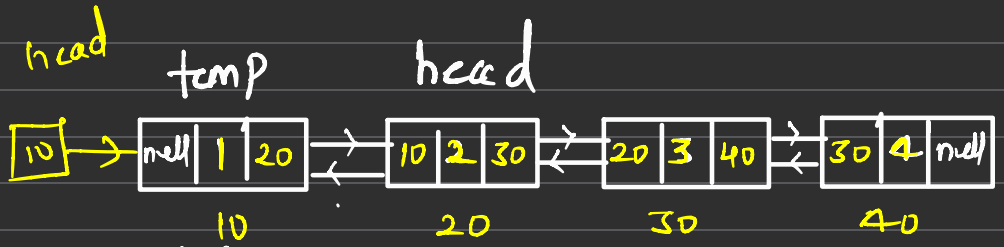


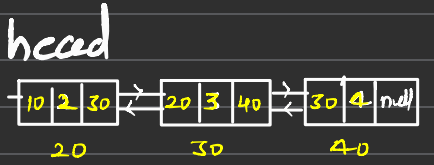

```
Node temp = head
while ( temp.next != null )
{
        temp = temp → next

}
    temp → next  = newNode
    newNode → prev = temp
```

```
if (head == null)
    head = newNode
```

# Deletion → Start

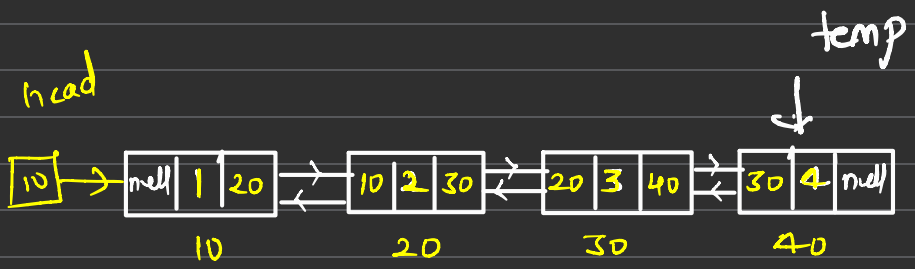
```
if (head = null)
{
    return
}
else if (head->next == null)
{
    Node temp = head
    head = NULL
    delete (temp)
    count --
    return
}
```




```
Node * temp = head
head = head -> next
head -> prev = Null
delete ( temp)
(ount--
```

# — Delete at end




```
Node temp = head
while ( temp → next != null)
 }     temp = temp → next
 >
    temp → prev → next = null
     delete ( temp )
```