

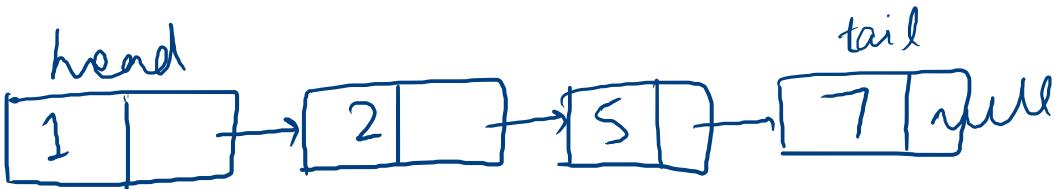
DSA-301

LECTURE #10

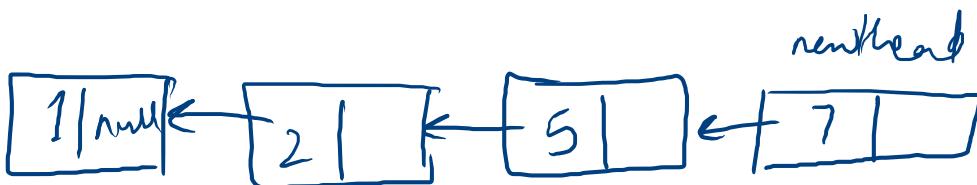
TOPIC : LINKED LIST -4

- Add 1 to Linked List
- Add 2 Linked Lists
- Introduction to Doubly Linked Lists

# Add 1 to Linked List



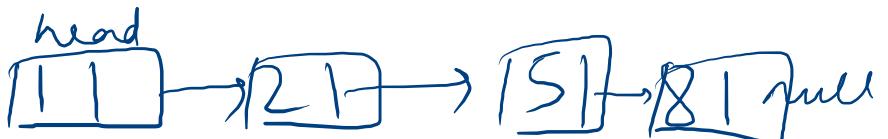
Step 1) Reverse the Linked



Step 2) Add 1 to head node & check for carry



Step 3) Reverse the LL



Variant 1

$$\begin{array}{r} 1257 \\ + 1 \\ \hline 1258 \end{array}$$

Linked List

$$\text{carry} = \text{node.data} / 10$$

$$\begin{array}{r} 1259 \\ + 1 \\ \hline \end{array}$$

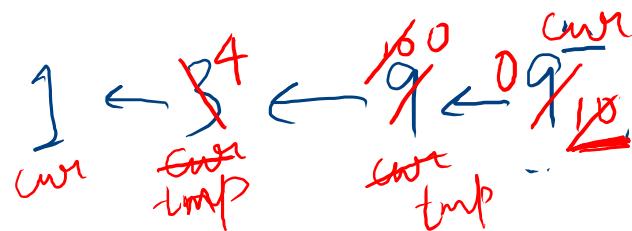
10

$$\text{node.data} = \text{node.data \% 10}$$

i/p

1 → 3 → 9 → 9

Step 1)



carry 0  
1

Step 2) var cur = head  
var carry = 0

cur.data = cur.data + 1

\* while (cur != null && (cur.data > 9 or carry > 0))

{

    tmp = cur

cur.data = cur.data + carry

Carry = cur.data / 10 -

cur.data = cur.data % 10

cur = cur.next

}

Step 3) Reverse the LL

1 → 4 → 0 → 0

Variant 2

head

$$\begin{array}{r} 1 \ 3 \ 9 \ 9 \\ + \ 1 \\ \hline 18 \end{array}$$

{ carry = node.data / 10

node.data = node.data % 10

} node = node.next

i/p : 9 → 9 → 9

Step 1)  $9 \leftarrow 9 \leftarrow 9 \text{ (10)}$  head

Step 2) var cur = head  
var carry = 0

cur.data = cur.data + 1

\* while (cur != null & & (cur.data > 9 || carry > 0))

{

tmp = cur

cur.data = cur.data + carry

→ carry = cur.data / 10

cur.data = cur.data % 10

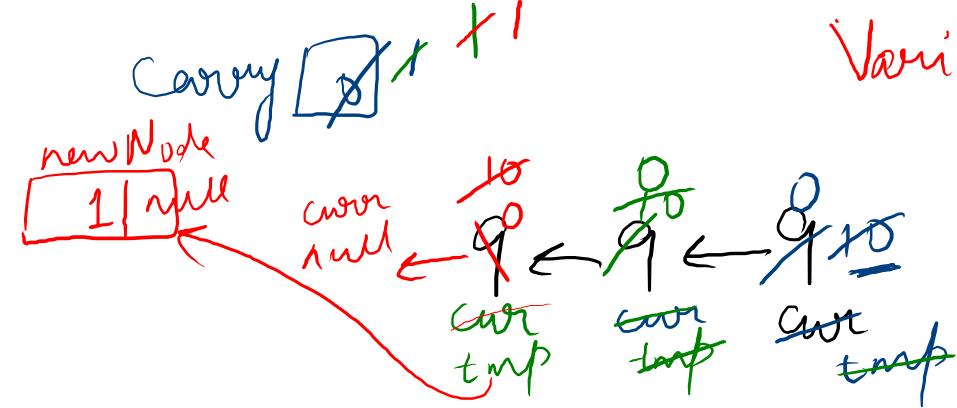
} cur = cur.next

if (carry > 0)

newNode = new Node(carry)

tmp.next = newNode

Step 3) Reverse  $\Rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0$



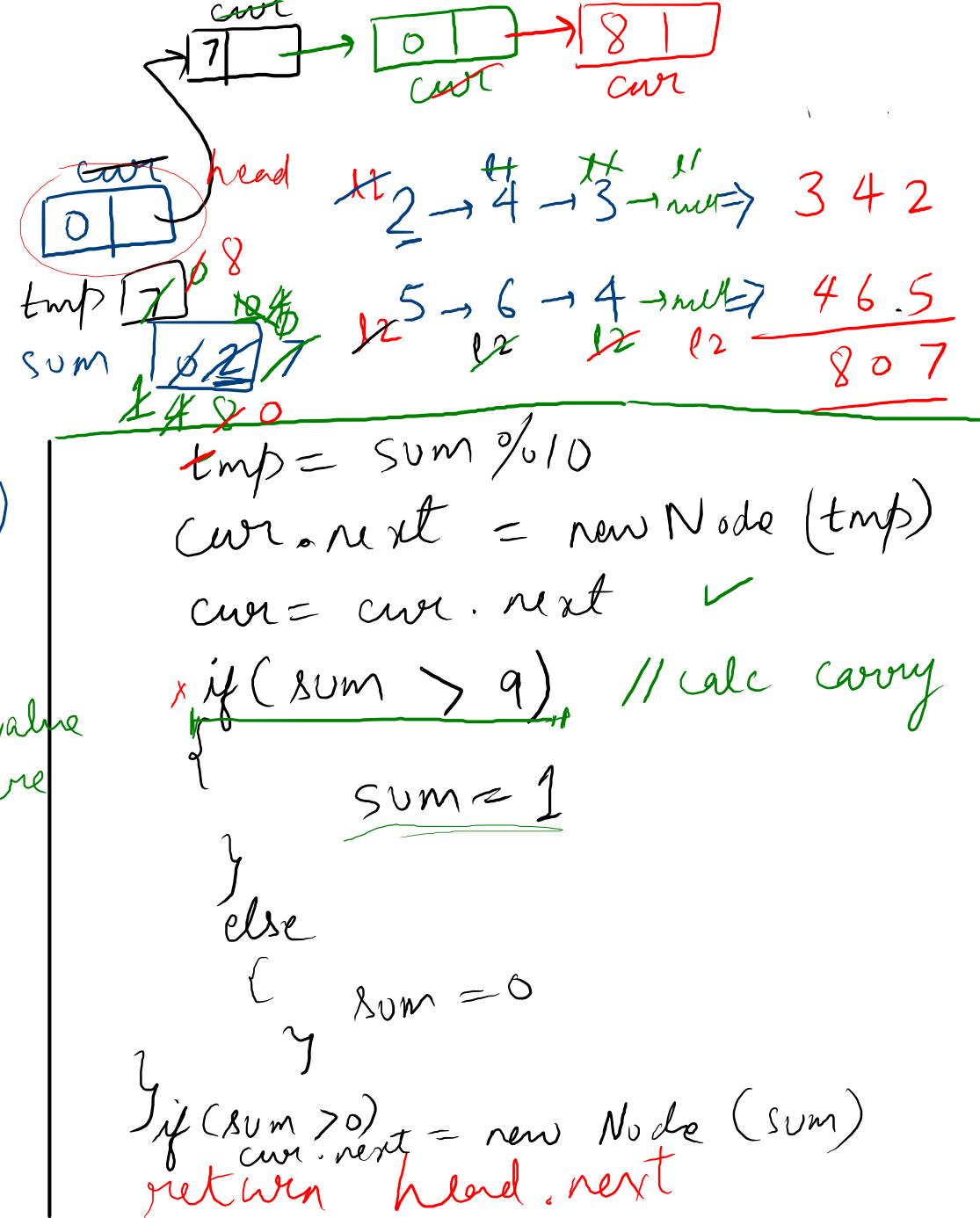
Variant 3

printLL()

op: 1 0 0 0

# Add 2 Linked Lists

```
function AddTwoLL(l1, l2)
{
    var sum = 0
    cur = new Node(0)
    head = cur
    while(l1 != null || l2 != null)
    {
        if(l1 != null)
        {
            sum += l1.data // carry value used here
            l1 = l1.next
        }
        if(l2 != null)
        {
            sum += l2.data
            l2 = l2.next
        }
    }
}
```



```
var sum = 0  
cur = new Node(0)  
head = cur
```

while(~~l1~~ < null || ~~l2~~ != null)

{

if (~~l1~~ != null)

sum += l1.data

if (~~l2~~ == null)

→ tmp = sum % 10  
cur.next = new Node(tmp)

cur = cur.next

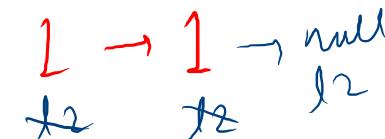
if (sum > 9)  
    sum = 1

else  
    sum = 0

~~ret head.next~~



342



$$\begin{array}{r} 11 \\ - 353 \\ \hline \end{array}$$

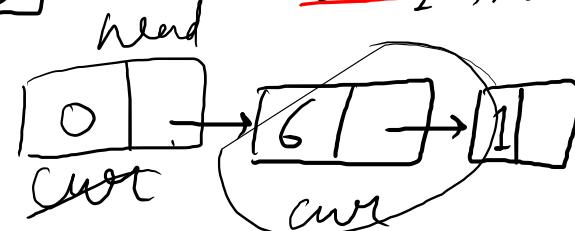


~~l1~~  
8 → null

~~l2~~ 8 →



16  
6 → 1 → null



Node(tmp)

if (sum > 0)

cur.next = new Node(sum)

ret head.next

```
Node  
{  
    const v(x)  
    {  
        data = x  
        next = null  
    }  
}
```



add a new node with data = 5

1) create a node

var node1 = new Node(5)

2) cur.next = node1



cur.next = new Node(5)

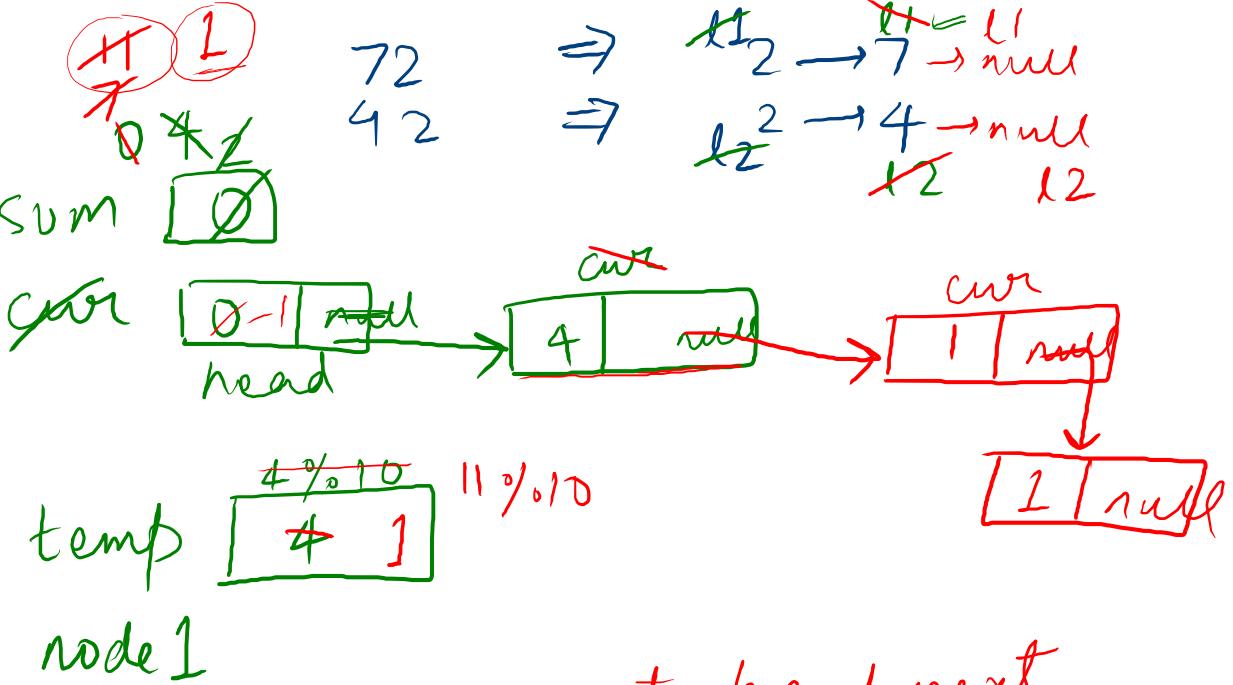
```

function AddTwoLL( l1, l2)
{
    var sum=0
    cur = new Node(0)// any data 0 or -1
    var ans = cur //save the starting of the sum LL
    → while( l1 != null || l2!= null)
    {
        if( l1 != null)
        {
            sum += l1.data
            l1 = l1.next
        }

        if( l2 !=null)
        {
            sum+= l2.data
            l2 = l2.next
        }

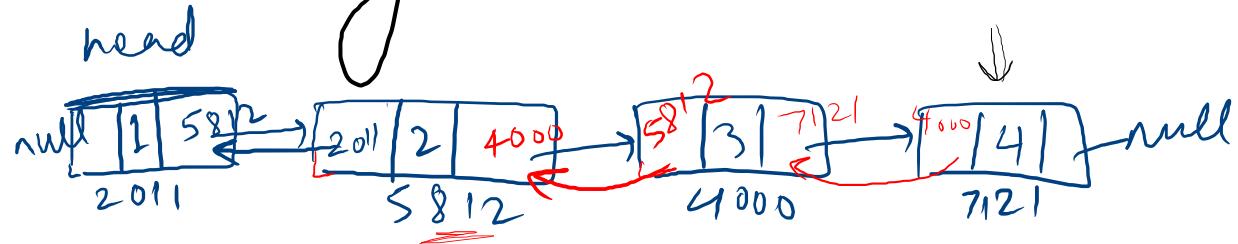
        temp = sum% 10
        node1 = new Node(temp) ✓
        cur.next = node1
        cur = cur.next
        // calculate carry & save it in sum var
        if( sum >9) ✗
            sum=1 // new carry
        else
            sum =0
    } // while end
    if( sum >0)
        cur.next = new Node(sum)
    return ans.next
}

```



$$\begin{array}{r}
 72 \\
 42 \\
 \hline
 114
 \end{array} \Rightarrow 4 \rightarrow 1 \rightarrow 1 \quad \checkmark$$

# Doubly Linked List

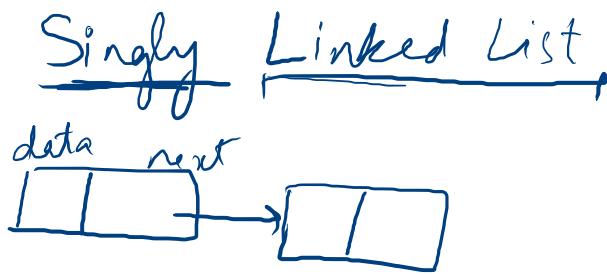


Double links



identify last node?

if (prev == null)  
→ head node



## Advantages

- 1) reverse easier
- 2) deletion easier
- 3) traverse in both direction

## Insertion at End

DLLNode

```
{ constr( x )
    data = x
    next = null
    y prev = null
```

```
if(this.head != null)
{ newNode = new DLLNode( 5 )
```



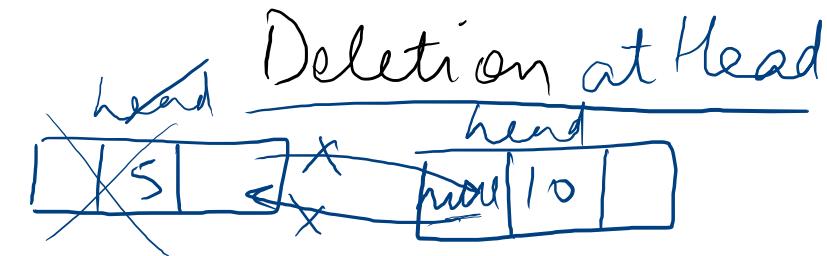
```
y head = newNode
```

```
else
```

```
{ node1 = newDLLNode( 10 )
```

```
✓ curr . next = node1
```

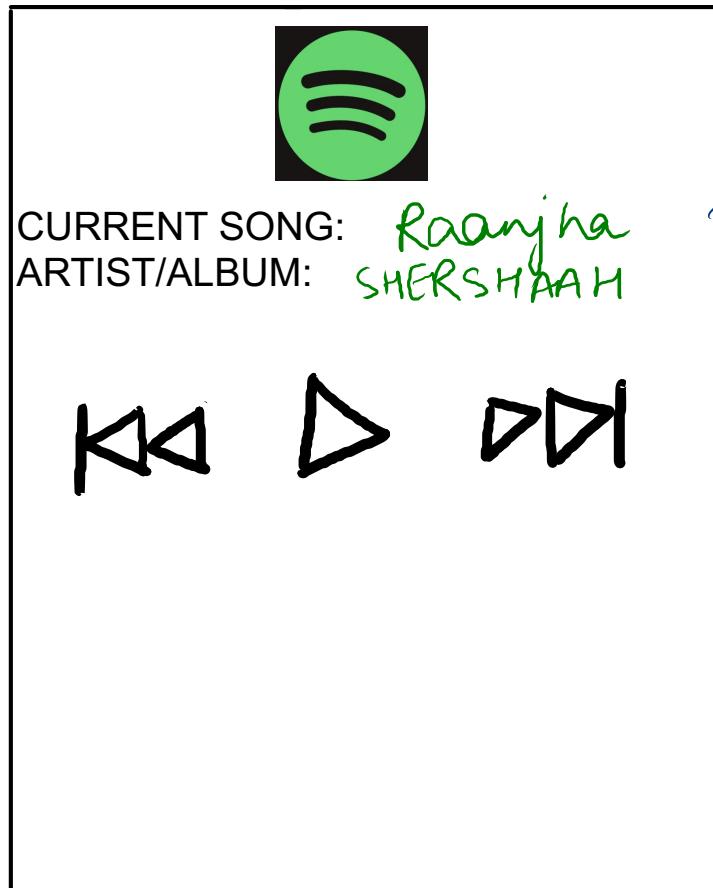
```
y node1 . prev = curr
```



head = head.next

head.prev = null

# MUSIC APP



Song 2

prev

next