

Remove Operation from Double Linked List & Circular linked list

Double-Linked List, Removal (Deletion) Operation.

There are **four** cases, which can occur while **removing** the node.

Note: If List has only one node, which is indicated by the condition, that the **head points** to the **same node as the tail**, the removal is quite simple by **set both head and tail to null**.

Void Remove first

if (head == null)

return;

else

n -- ;

if (head . after == null)

head = null ;

Tail = null ;

else

head = head . after ;

head . before = null ;

// or it can be as if (head == tail)
or it can be as if (tail . before == null)

Void Remove last

if (head == null)

return;

else

n -- ;

if (head . after == null)

head = null ;

Tail = null ;

else

Tail = tail . before ;

Tail . after = null ;

// or it can be as if (head == tail)
or it can be as if (tail . before == null)

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

else

If (loc >=1 && loc <= n)

Remove first()

```
// call remove first function
```

```
if (loc == n)
```

Remove last()

```
// call remove last function
```

n - - ;

P = head ;

```
For I = 1 ; I <= loc-2 ; I ++
```

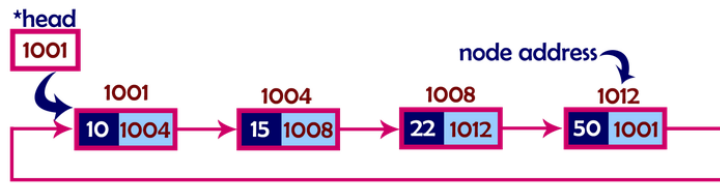
$p = p$. after ;

$$p \cdot \text{after} \cdot \text{after} \cdot \text{before} = p ;$$
$$p \cdot \text{after} = p \cdot \text{after} \cdot \text{after} ;$$

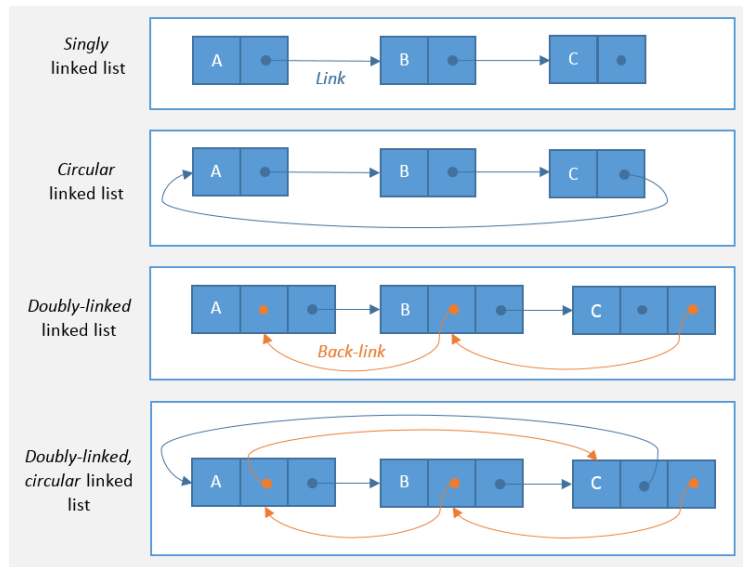
S.O.P (error value of loc);

In single linked list, every node points to its next node in the sequence and the last node points NULL. But in circular linked list, every node points to its next node in the sequence but the last node points to the first node in the list.

That means circular linked list is similar to the single linked list except that the last node points to the first node in the list



Circular linked list can be circular singly linked list or circular doubly linked list.



In a circular linked list, we perform the main following operations : (Insertion, Deletion, Display).

H. W.// Write a function for creation circular linked list which has n nodes .

Note: You can use the creation function of singly linked list, and finally you need to link the last node (let it p) with the first node (head) as:

$$P . next = head$$

Printing Circular Linked List

If head == null

S.O.P ("empty linked list")

else

P = head

While p . next != head

 S.O.P (p . data)

 P = p . next

S.O.P (p . data) // for last node

or

If head == null

S.O.P ("empty linked list")

Else

P = head

Do

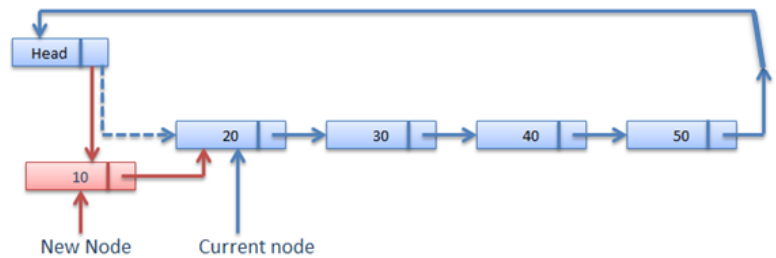
 S.O.P (p . data)

 P = p . next

While p != head

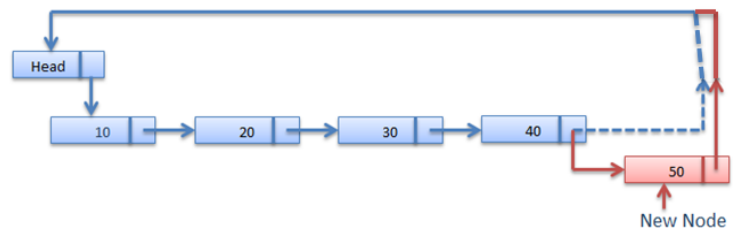
Inserting At Beginning of the list

```
P = head
While p . next != head
P = p . next
//read v
Node = new clink (v, head)
Head = node
P . next = head
```

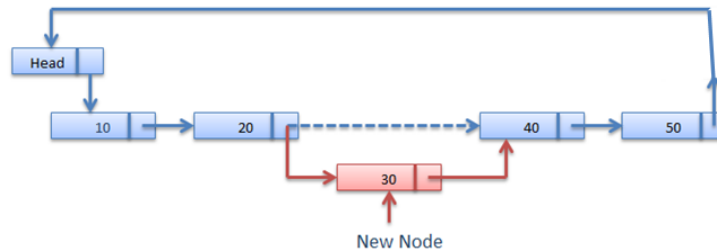


Inserting At Ending of the list

```
P = head
While p . next != head
P = p . next
//read v
Node = new clink (v, head)
P . next = node
```



H.W. // Write the function of inserting a node in circular linked list in any position.



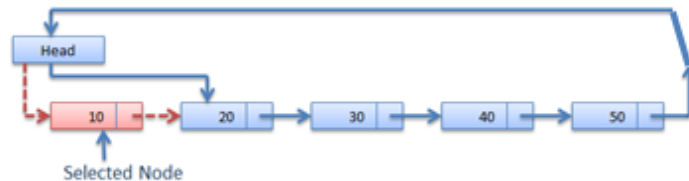
Delete From The Beginning of Circular Linked List

If head . next == head // There is only one node

Head = null

Else

```
P = head
While p . next != head
P = p . next
Head = head . next
P . next = head
```



Delete From The End of Circular Linked List

If $\text{head} \rightarrow \text{next} == \text{head}$

Head = null

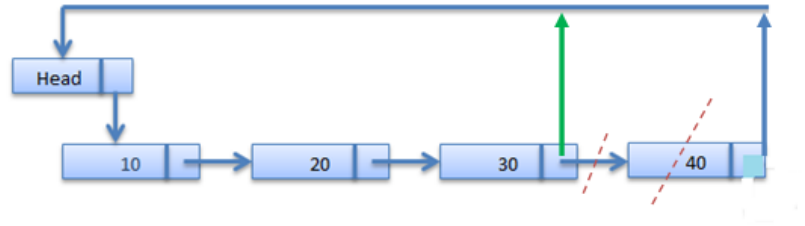
Else

P = head

While $p \rightarrow \text{next} \rightarrow \text{next} \neq \text{head}$

P = p . next

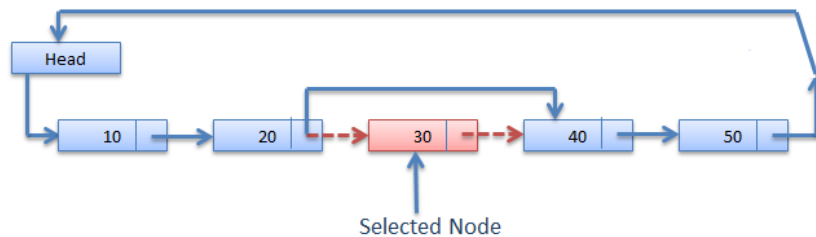
P . next = head



H. W.//

A- Write a function for deletion from circular linked list based on the position of a node.

B- Write a function for deletion from circular linked list based on the data value.



Examples Of Operations In Circular Doubly Linked List

