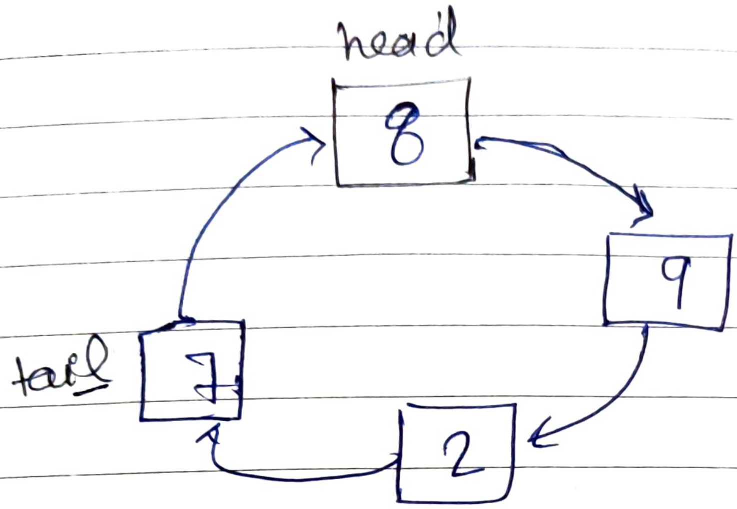
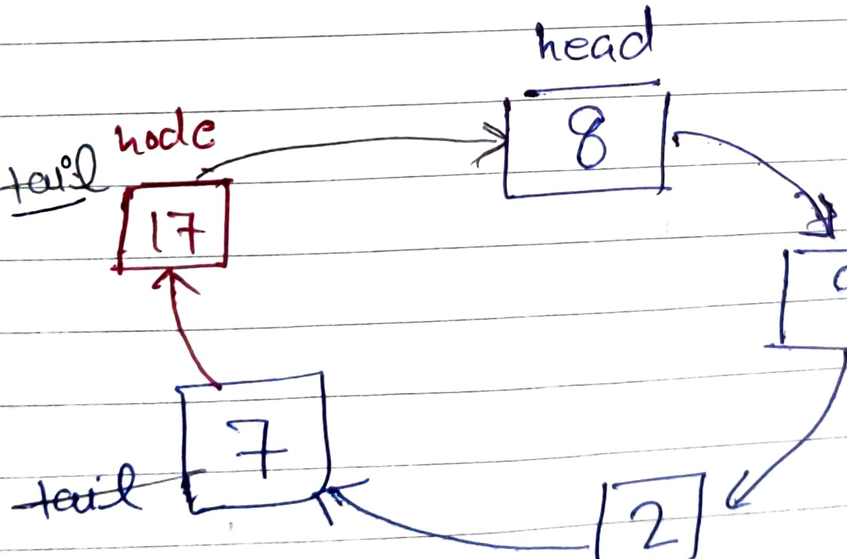
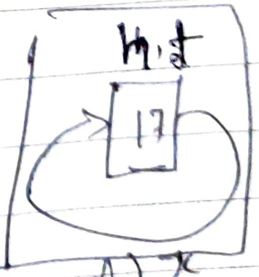
Circular linked list



```
class Node {
  int val;
  Node next;
}
```

So, here it is not null.
No one is pointing to null. until
linked list is ~~to~~ empty.





```
tail.next = node
```
$\boxed{\text{node.next = head}}$
```
tail = node
```

```
if (head == φ)
    head = node
    tail = node
```

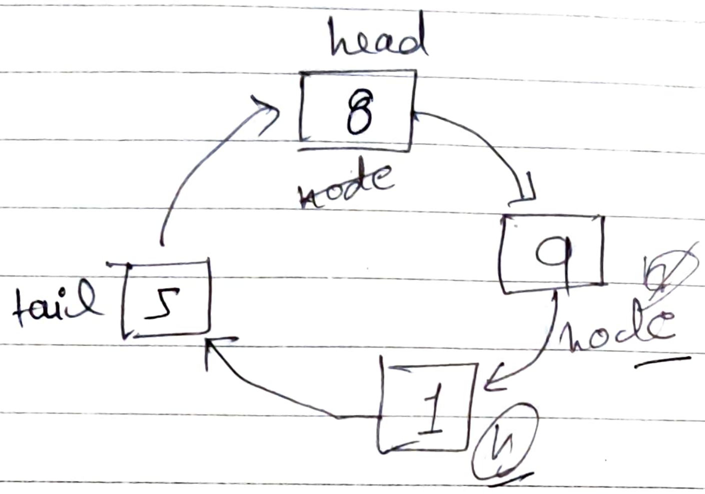
notes

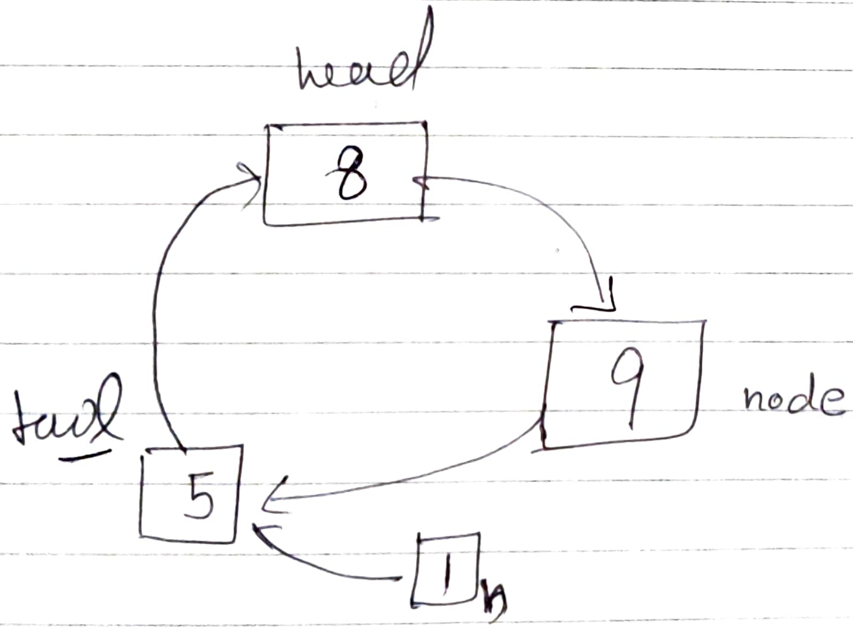
code :-
```
public class CLL {
        private Node head;
        private Node tail;

    public CLL() {
            this.head = null;
            this.tail = null;
        }
    public void insert(int val) {
            Node node = new Node(val);
            if (head == null) {
                head = node
                tail = node;
                return;
            }
            tail.next = node;
            node.next = head;
            tail = node;
        }
public void display() {
        Node node = head;
        if( head != null) {
            do {
              System.out.print(node.val+"->");
              if( node.next != null) {
                  node = node.next;
              }
            } while (node != head);
        }
        sout("HEAD");
    }
```

How do we delete :-

val = 1





O(n)

notes

Code for delete :-

```
public void delete (int val) {
        Node node = head;
       if (node == null) {
             return;
         }
    if( head == tail) {
          head = null;
           tail = null;
           return;
      }
     if (node.val == val) {
           head = head.next;
            tail.next = head;
             return;
     }
  do {
        Node n = node.next;
           if ( n.val == val) {
              node.next = n.next;
              break;
        }
          node = node.next;
       } while(node != head);
    }
```