Collection

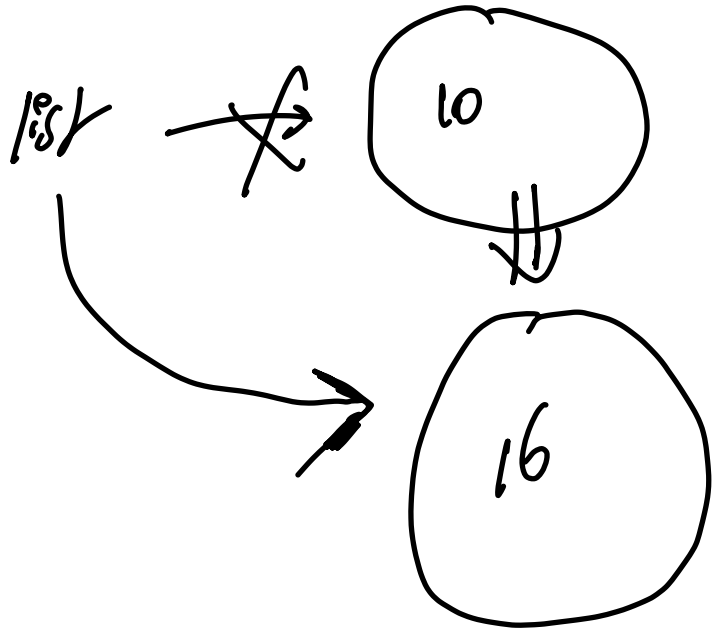
- list
- set
- queue

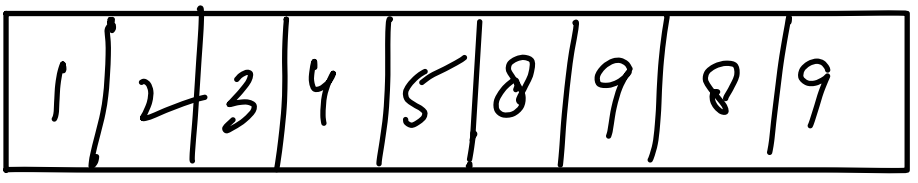
Todays topic

1 → list Interface
- → ArrayList
- → LinkedList
- → Vector (Legacy)

list Interface

ArrayList = (Capacity * 3/2) + 1





RandomAccess <- AL, Vector

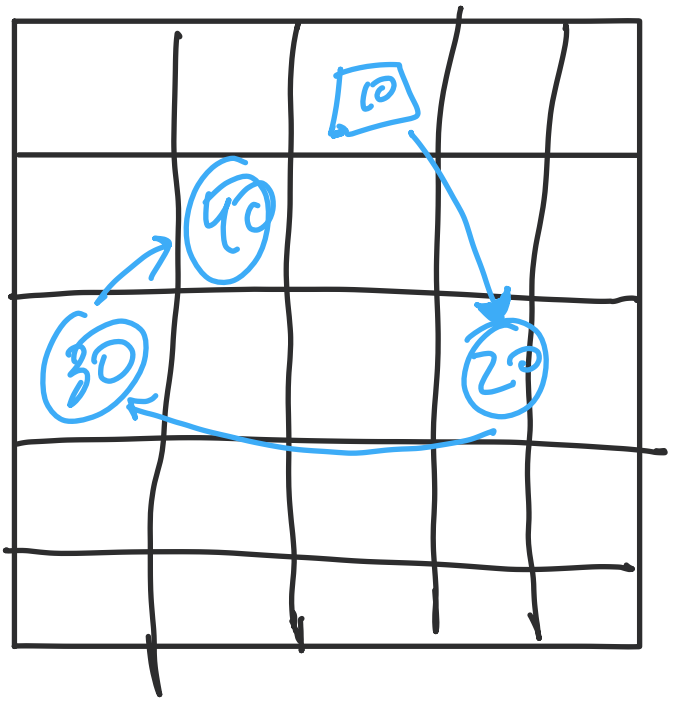
↳ marker Interface

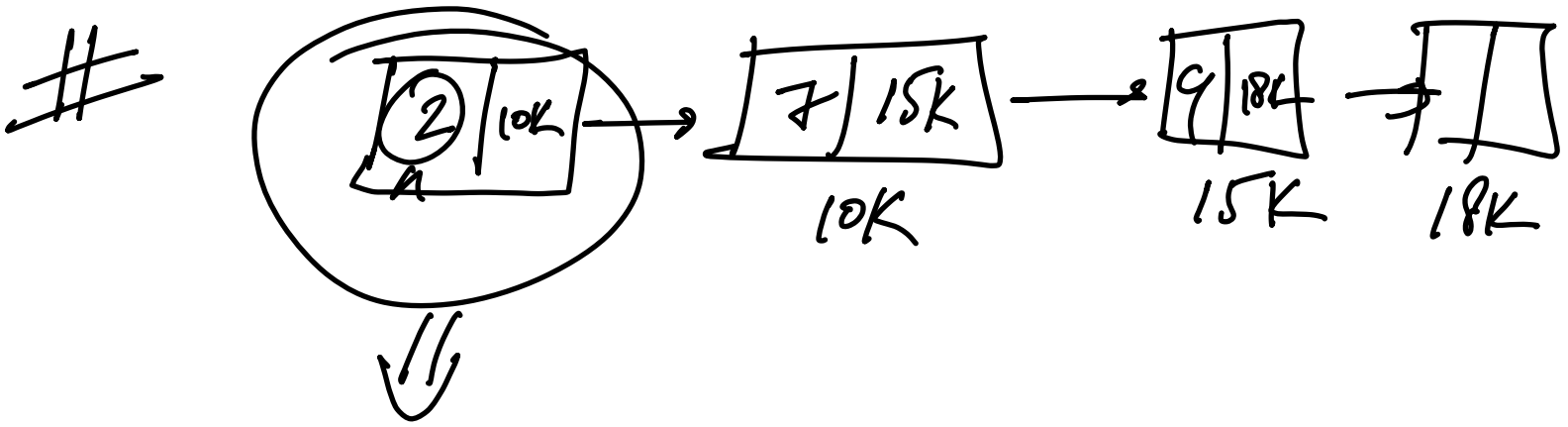
Clonable Interface

LinkedList

↳ DLL

↳ CL

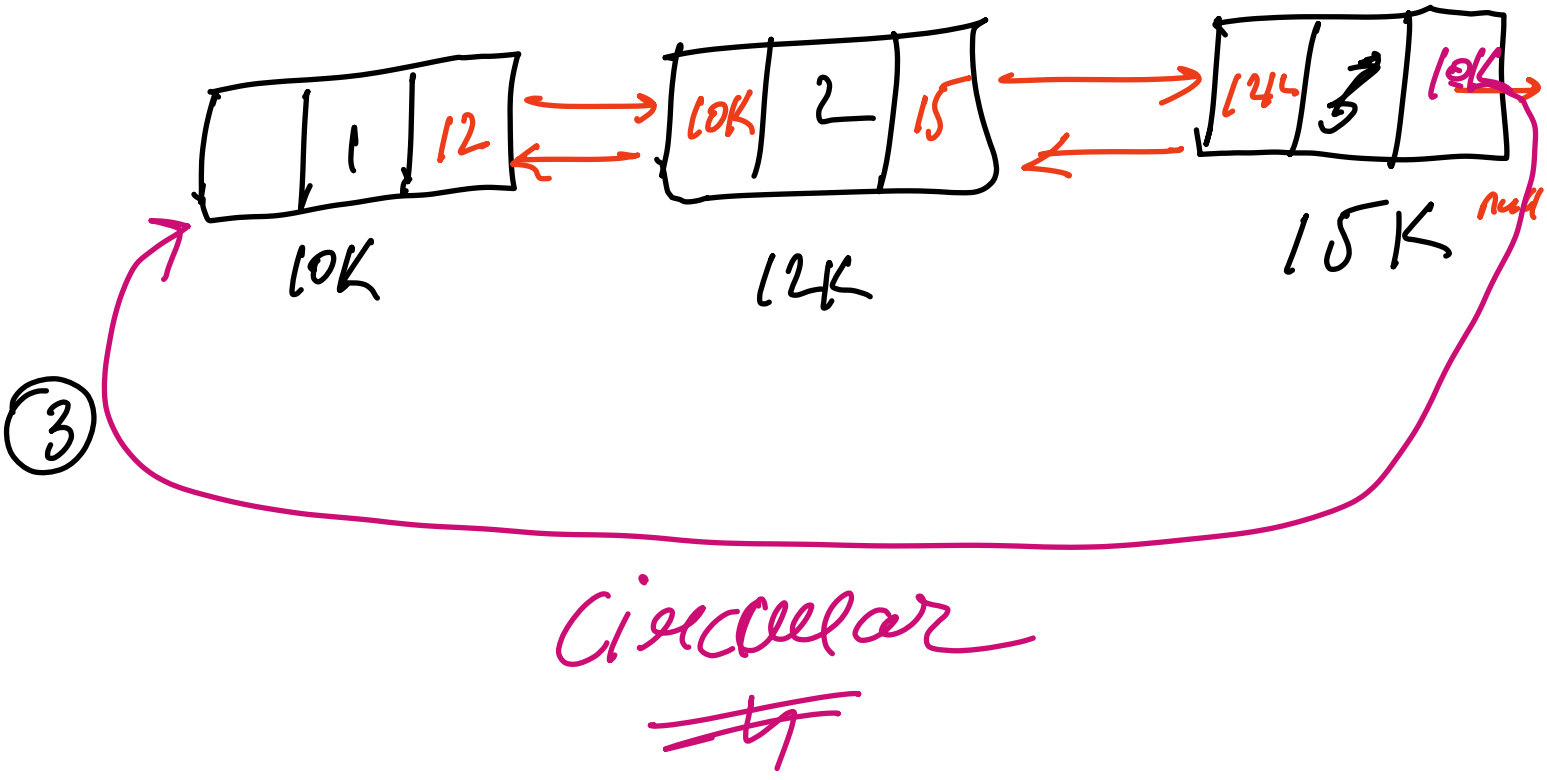




① data
② Address of the next Node

① Single LL

② Double LL



Circular

⇒ addFirst()

⇒ addLast()

class Node {

int data;

Node next;

⇒ Node(int data) {

this.data = data;

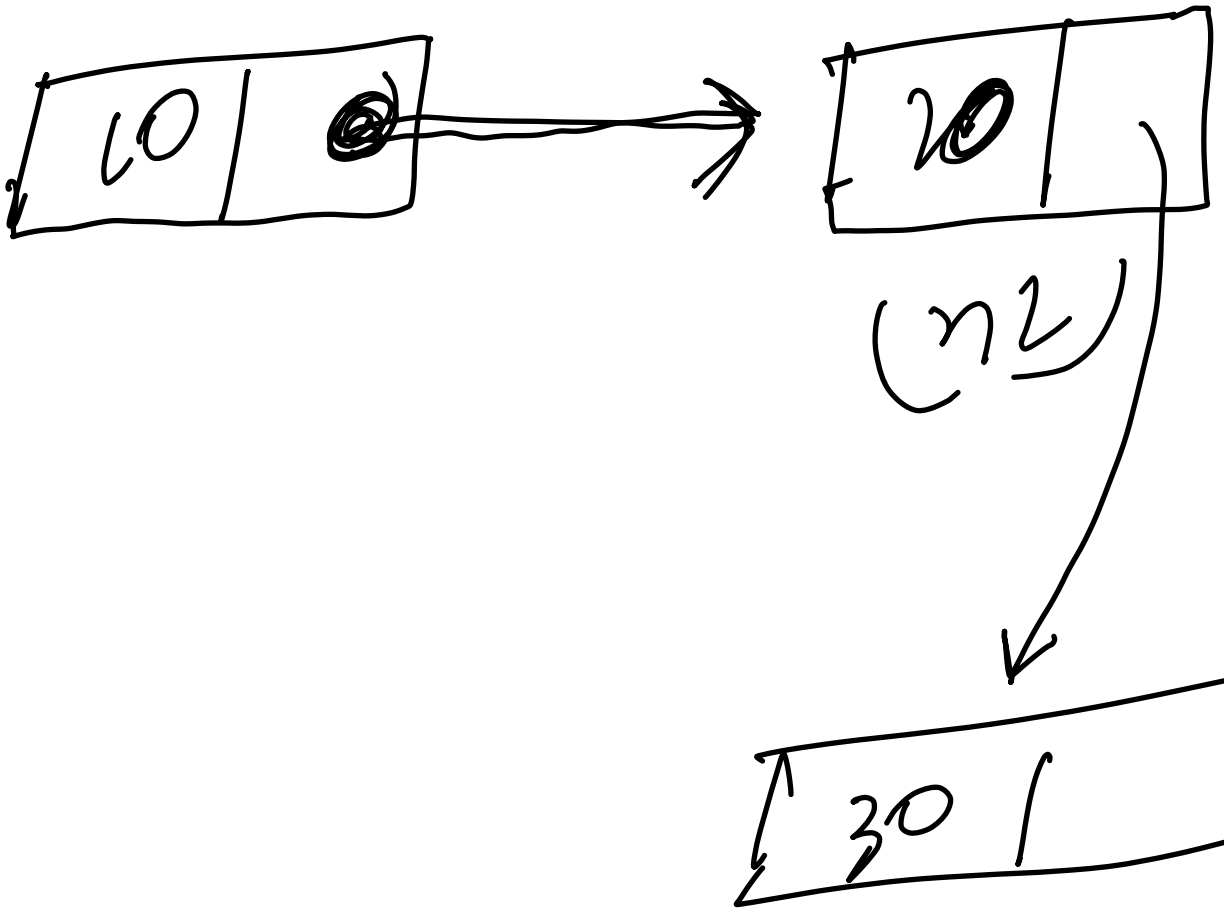
this.next = null;

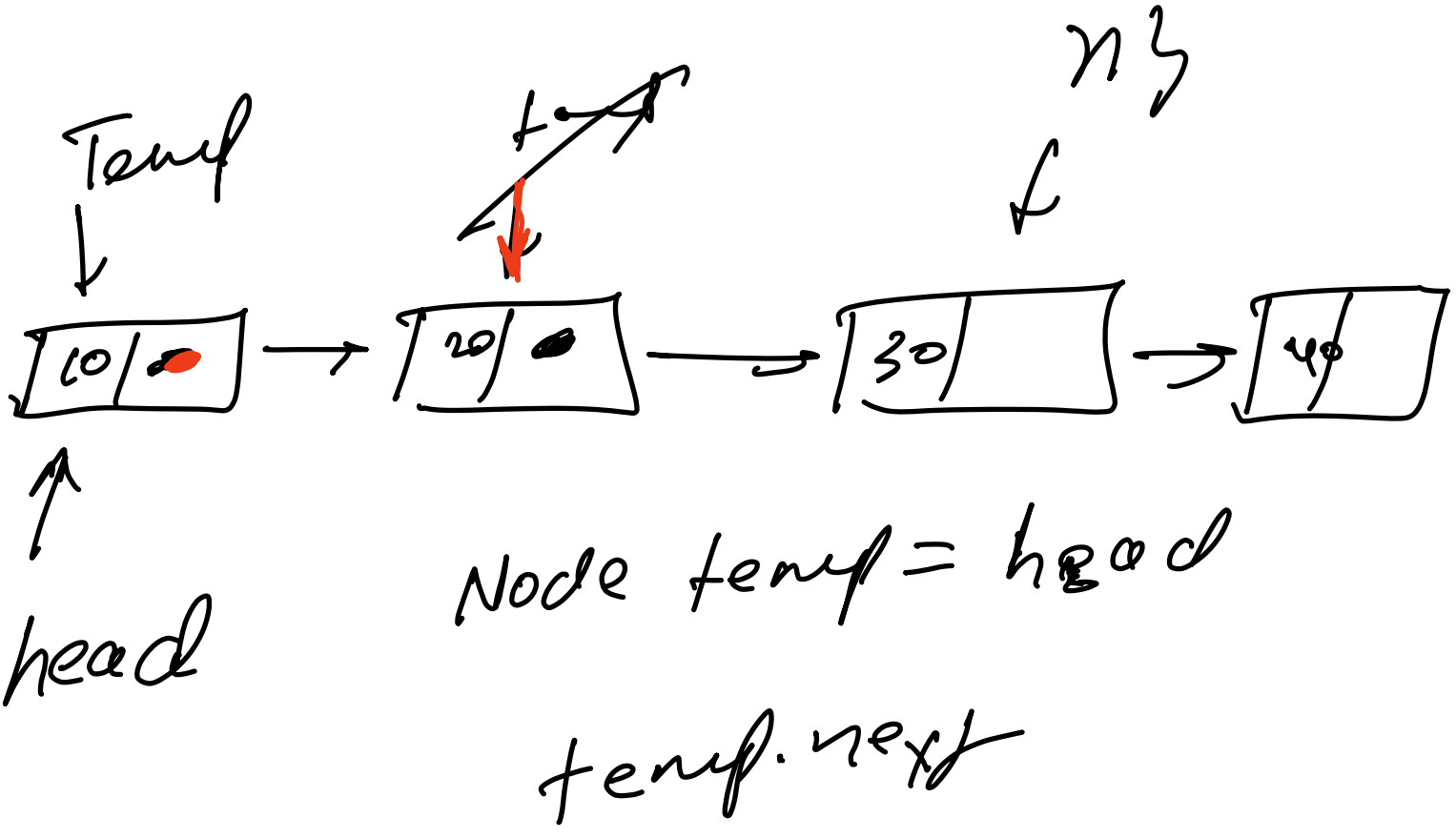
}

Node n1 = new Node(10);

Node n2 = new Node(20);

n1.next = n2





Node temp = head

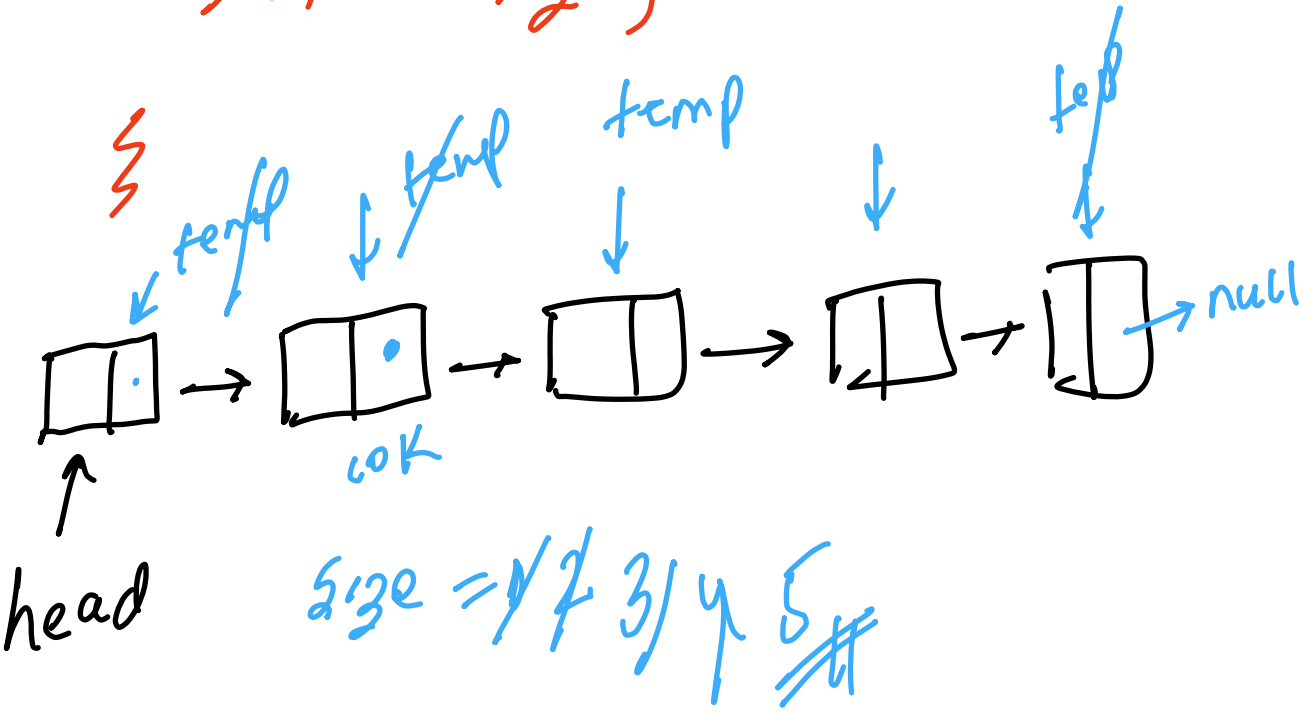
temp.next

```
# class LinkedListDemo {
    public static Node head;
    public static int sizeOfLL() {
        Node Temp = head
        int size = 0 ;

        while (temp != null) {
            size++ ;
        → temp = temp.next ;

        }

        return size ;
```



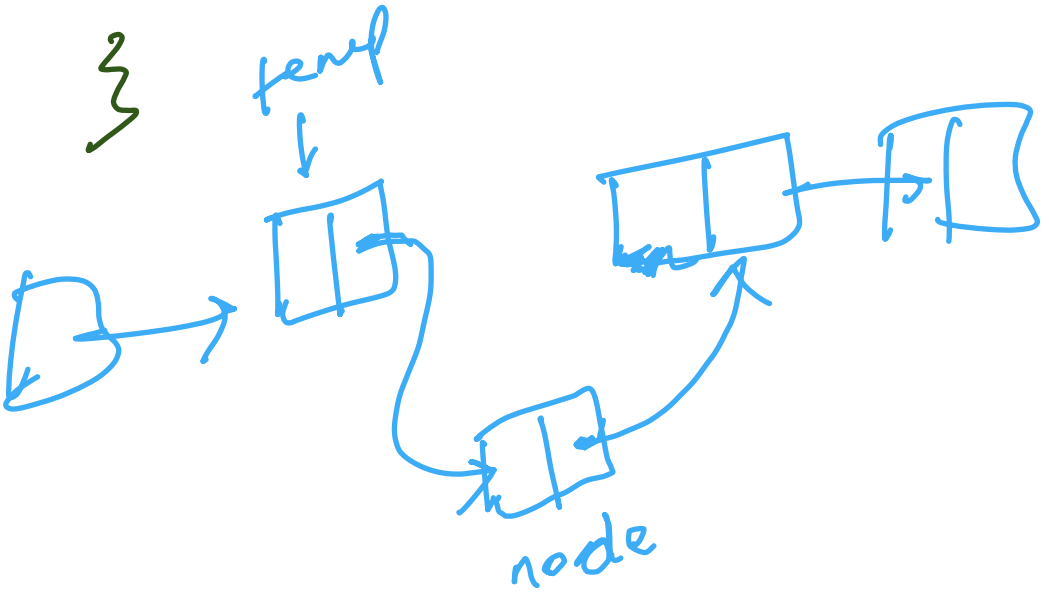
size = 4

```
public static void insertNode(int pos, int value){
    Node temp = head;
    int size = sizeofLL();
    Node node = new Node(value);
    if(pos < 1){
        return;
    }
    if(pos > size+1){
        return;
    }
    if(pos == 1){
        node.next = head;
        head = node;
    }
}
```

for(i=1; i<pos-1; i++){

temp=temp.next;

}

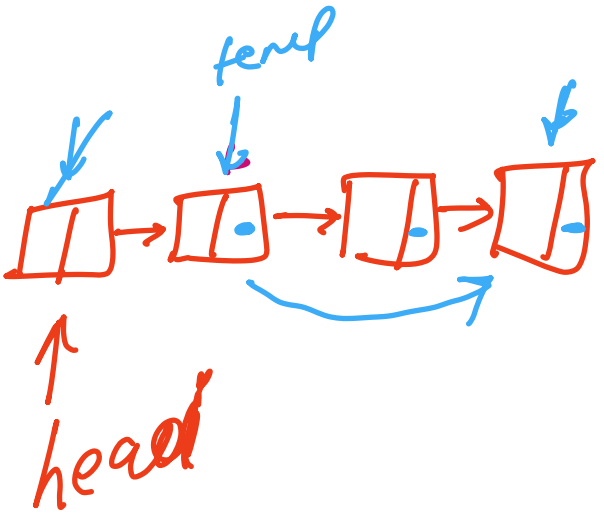


node.next = temp.next;

temp.next = node

}

```
public static void deleteNode(int pos){
    int size = sizeOfLL();
    if(pos<1){
        return
    }
    if(pos>size){
        return;
    }
    if(pos==1){
        head = head.next;
    }
    else{
        for(i=1; i<pos-1; i++){
            temp = temp.next;
        }
        temp.next = temp.next.next;
```



```
public static void printLL() {
    Node temp = head;
    while(temp.next != null) {
        System.out.println(temp.data + " ");
        temp = temp.next;
    }
    System.out.println(temp.data);
}
```