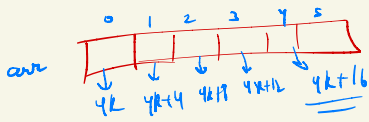
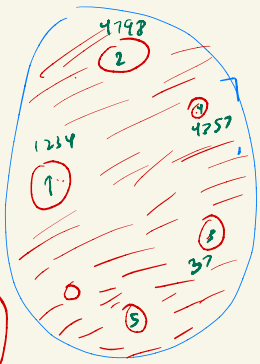
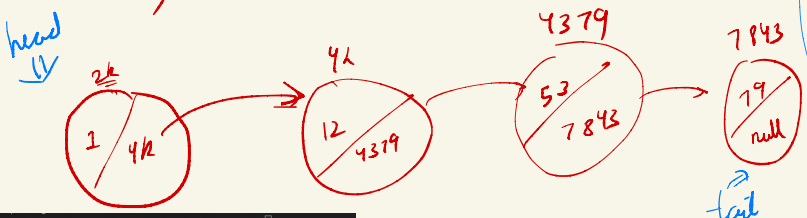
Linked List



y in memory

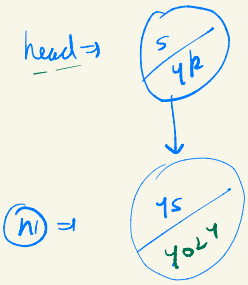
class Node {
int data;
Node next;
}






```java
public static void main(String[] args) {
    Node head=new Node(data: 5);

    Node n1=new Node(data: 45);

    head.next=n1;

    Node n2=new Node(data: 7);
    n1.next=n2;

    Node nextOfHead=head.next;
    System.out.println(n1+" fggfg "+nextOfHead);
    System.out.println(nextOfHead.data);
}
```

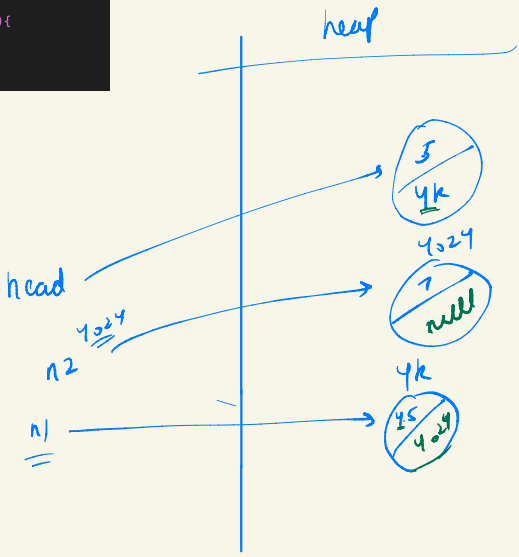
```java
class Node {
    int data;
    Node next;

    public Node(){

    }

    public Node(int data){
        this.data=data;
    }
}
```

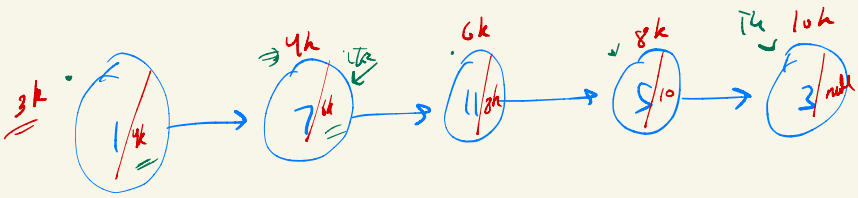


head.data
head.next = 4k

n1.data = 45
n1.next = 4024

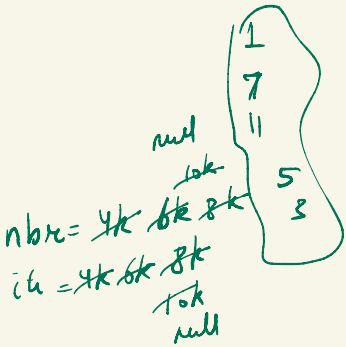
n1





head
(3k)

```java
public static void printLinkedList(Node head){
    System.out.println(x: "Printing linked list");
    Node itr=head;

    while(itr!=null){
        System.out.print(itr.data+" ");

        Node nbr=itr.next;
        itr=nbr;
    }
}
```



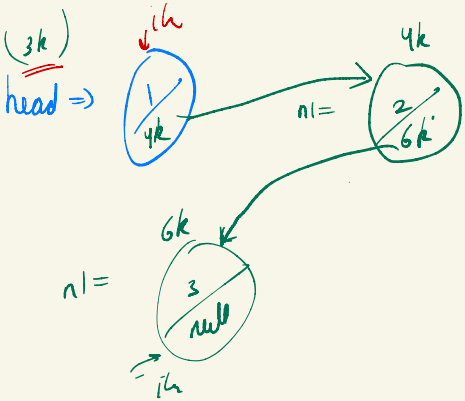
3 ⇒ 1, 2, 3

data = 2 3

f- l = 1

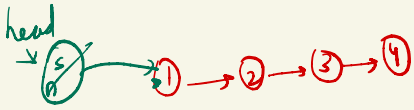
itr =

```java
    Scanner scn=new Scanner(System.in);
    System.out.println(x: "Enter number of inputs ");
    int n=scn.nextInt();

    int first_ele=scn.nextInt();

    Node head=new Node(first_ele);
    Node itr=head;

    for(int i=1; i<n; i++){
        int data=scn.nextInt();

        Node n1=new Node(data);
        itr.next=n1;

        Node nbr=itr.next;
        itr=nbr;
    }

    printLinkedList(head);
}
```

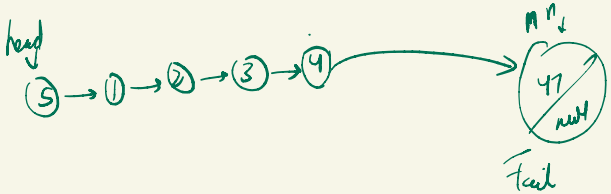


addFirst

val = 5

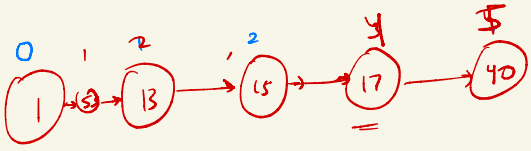



```
Node nn = new Node(val);
nn.next = head;
head = nn;
```



fail

```java
public static Node getTail(Node head){
    Node itr=head;

    while(itr.next!=null){
        // itr=itr.next;
        Node nbr=itr.next;
        itr=nbr;
    }

    return itr;
}

public static void addLast(int val, Node head){
    Node tail=getTail(head);

    Node nn=new Node(val);
    tail.next=nn;

    tail=nn;
}
```

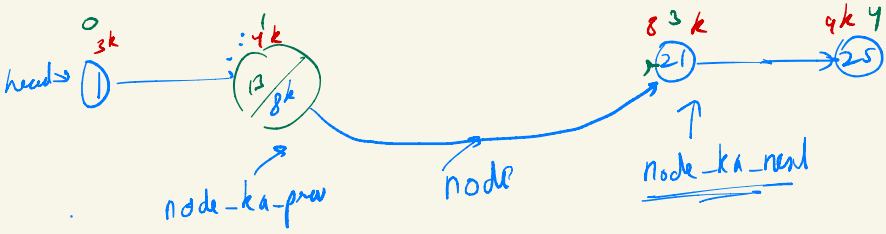


idx = 4, val = 55

i = pos + 2 = 3

```java
public static Node getAt(Node head, int idx){
public static Node addAt(Node head, int idx, int val){
public static Node removeAt(Node head, int idx){
    if(idx==0){
        return removeFirst(head);
    }

    Node node_ka_prev=head;

    for(int i=0; i<(idx-1); i++){
        Node nbr=node_ka_prev.next;
        node_ka_prev=nbr;
    }

    Node node=node_ka_prev.next;

    Node node_ka_next=node.next;

    node_ka_prev.next=node_ka_next;

    return head;
}
```

Ques Add a val at idx



idx = 2

i < 1

Ques Find length of linked list.