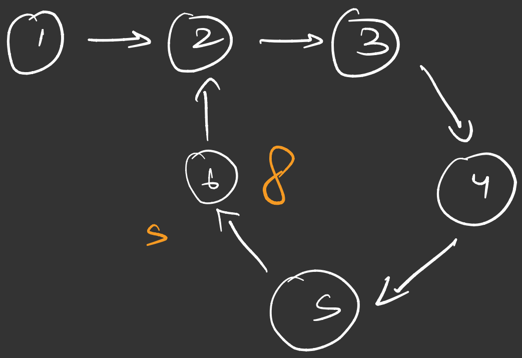
Ques Detect if there is a cycle in LL

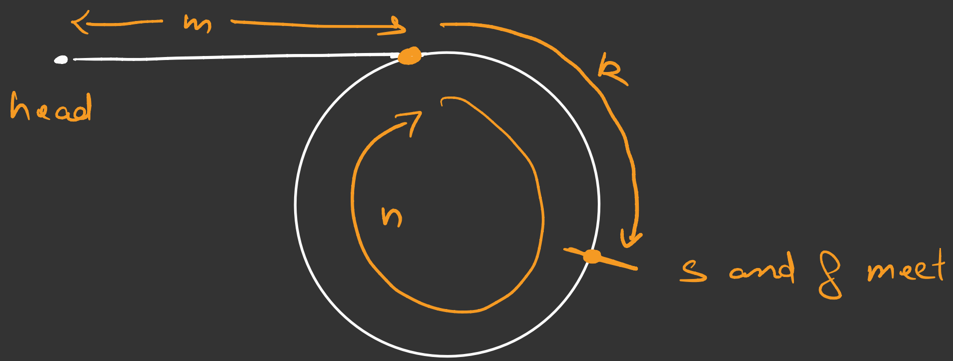


```
has Cycle ( ) {
  slow = head
  fast = head

  while( fast != null &&
              fast.next != null)

  { slow = slow.next;
    fast = fast.next.next;

    if (fast == slow)
        return true;
  }
  return false;
```

Ques Length of cycle

```
int c = 1;

slow = slow.next;

while ( slow != fast) {
   slow = slow.next;
   c++;
}
return count;
```

Ques Find starting point of the cycle



distance travelled by f = 2 × distance travelled by s

$$m + (i \times n) + k = 2 \times (m + (j \times n) + k)$$

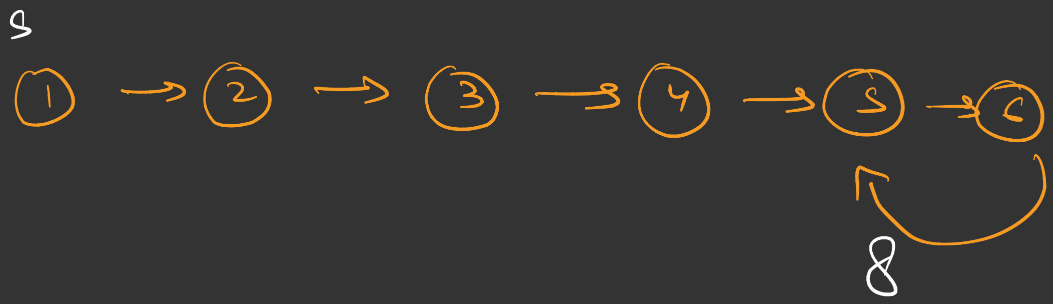
$$\cancel{m} + in + \cancel{k} = \cancel{2}m + 2jn + \cancel{2}k$$

$$\underbrace{m + k} = \underbrace{(i - 2j)}_{c}\boxed{n}$$

$$m + k = \text{factor of } n$$

$$\Rightarrow m = cn - k$$

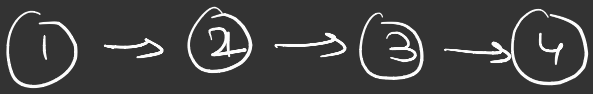
```
s = head
while(s != f) {
    s = s.next;
    f = f.next;
}
```



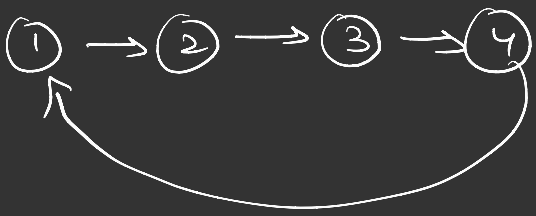
n = 2

Types of Linked List

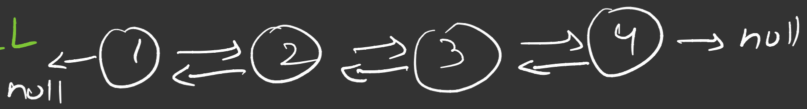
1 Singly LL



2 Circular LL



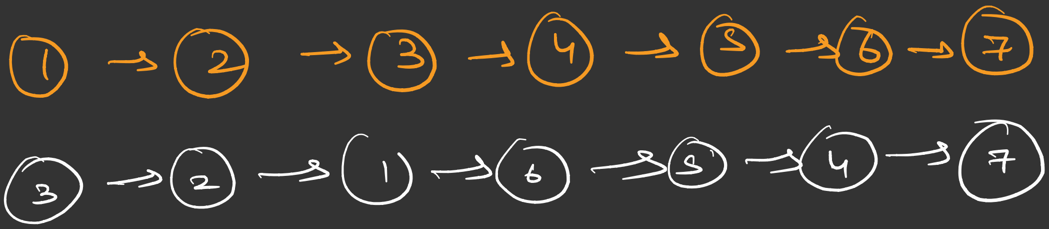
3 Doubly LL



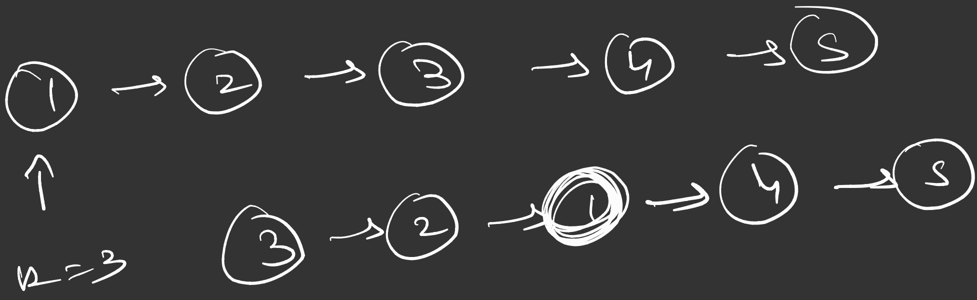
```
class Node {
    int data;
    Node next;
    Node prev;
}
```

```
Node {
    int data;
    Node next;

}
```

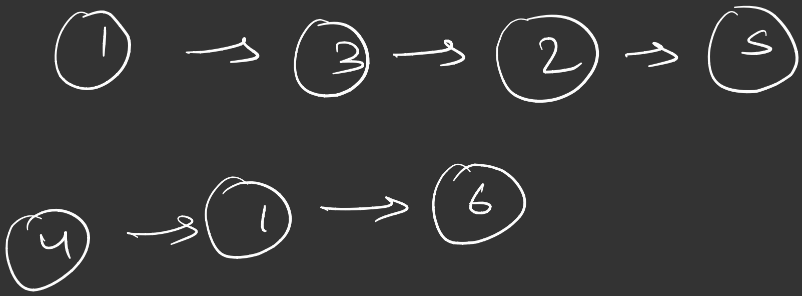
head   k = 3

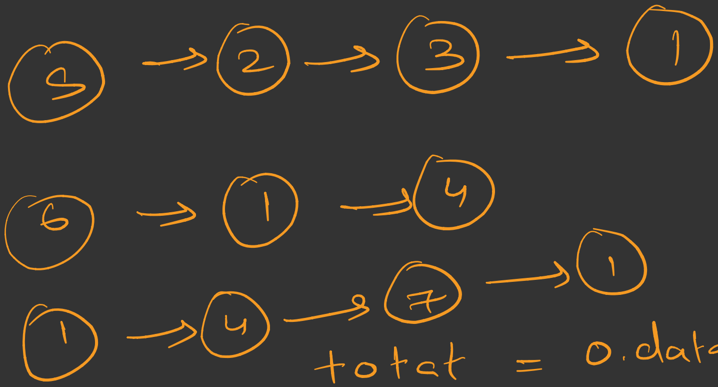


```
reverse knodes (Node n, int k) {

        k




    }
```



Ques Add 2 LL





totat = o.data + t.data + c

c = 0

5 + 6 + 0 = 11

n = new node()

n.data = total % 10;

c = total / 10

1 7 9 4 6 5 4 3 1 2

21