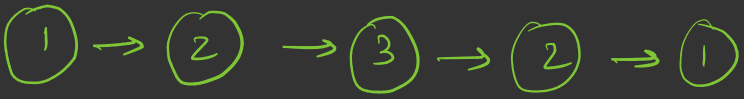
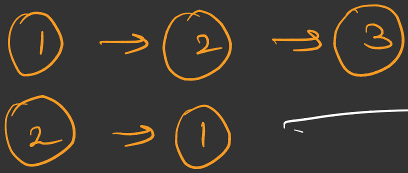
Ques Palindrome in LL

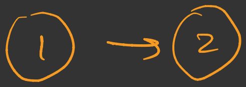


a) iterative solution

1 break from middle

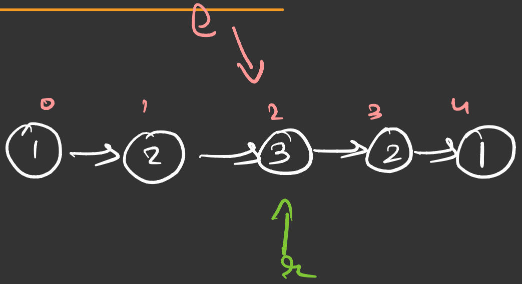


2 Reverse second half



3 compare data

Recursive solution

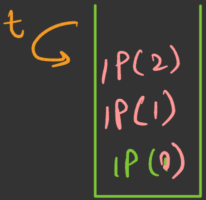


static left

right

isPalindrome(Node right) {

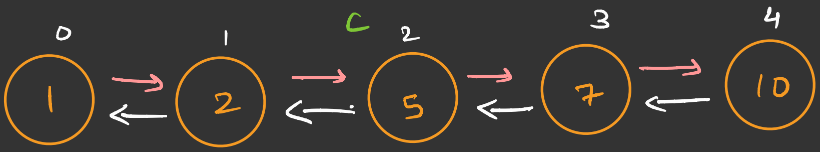
isPalindrome(right.next)



```
isPalindrome(Node right) {
    if (r == null) return true;
bool res = isPalindrome(right.next)

  if (res == false) return false;
  if (l.d == r.d) {
        l = l.next;
        return true;
    }
    return false
```

Ques insert in sorted LL



insert at kth index

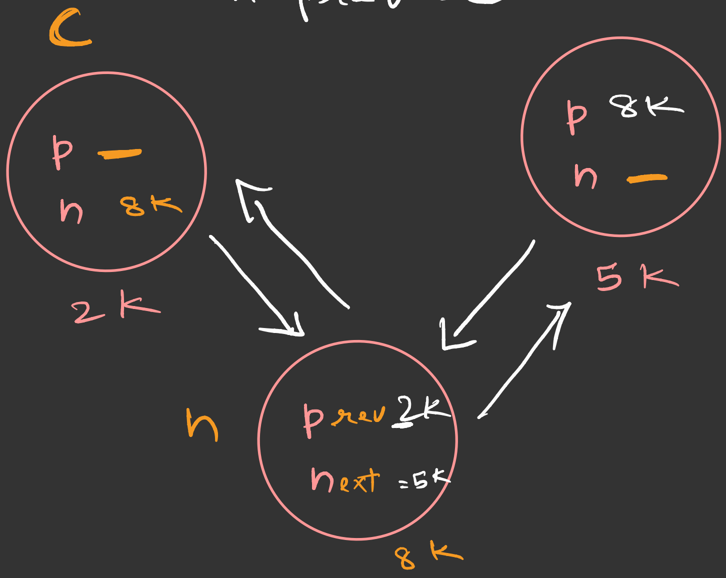
1) Reach k-1 th node

2) n.next = c.next

if(c.next != null) n.next.prev = n

c.next = n

n.prev = c

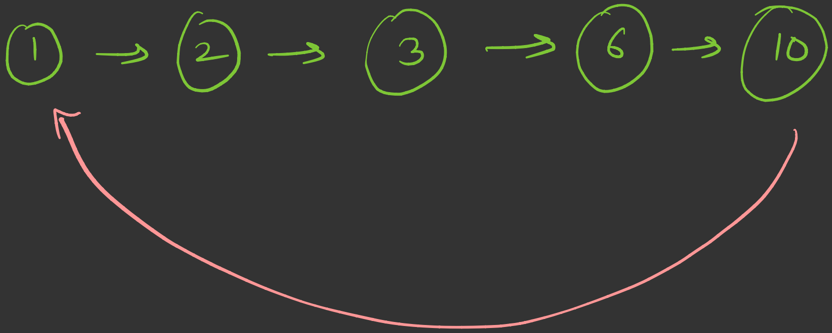


1. Empty LL → return newNode

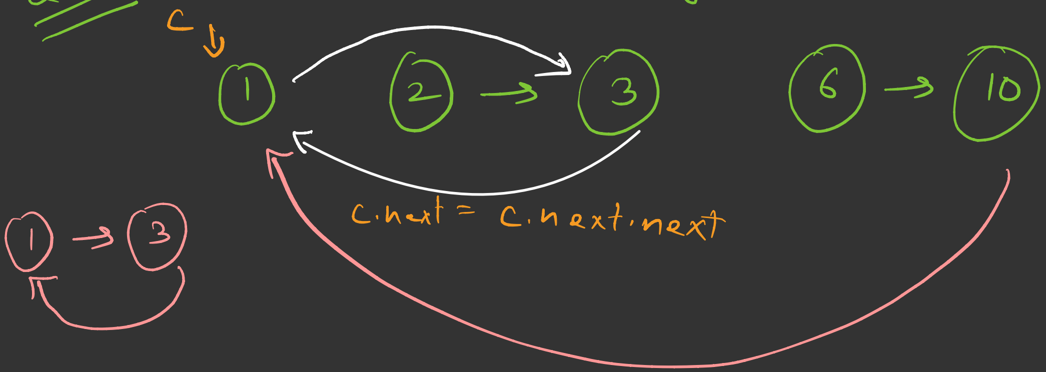
2. Insert at head
   n.prev
   n.next

3. Inert any where

Ques Convert to circular LL

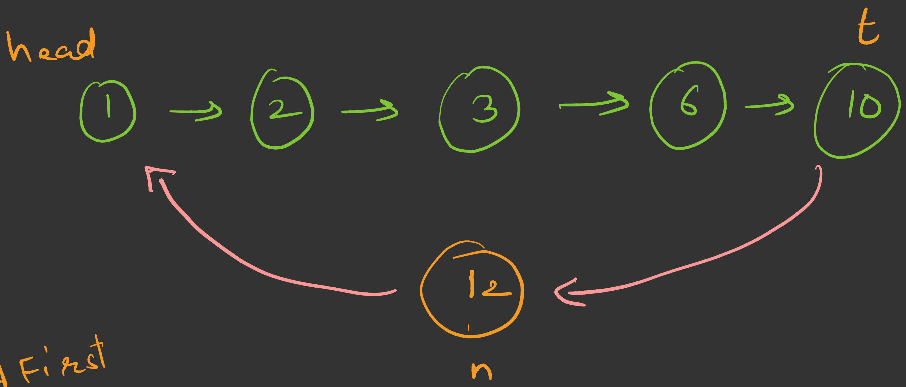


Ques Delete even nodes from the list



c.next = c.next.next

Circular LL



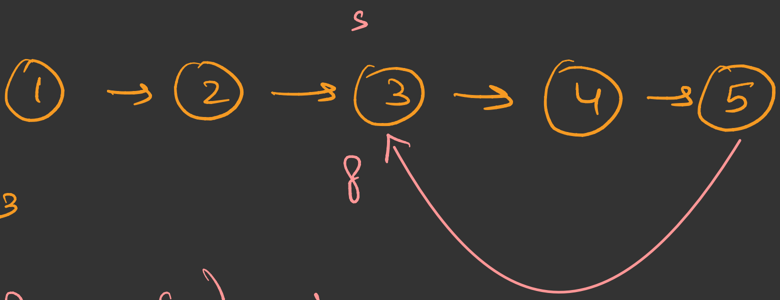
add First
add Last

```
while(t.next != head)     t = t.next

n = new Node(data)

t.next = n;
n.next = head
```

Ques check if there is a cycle LL



c = ~~1~~ ~~2~~ 3

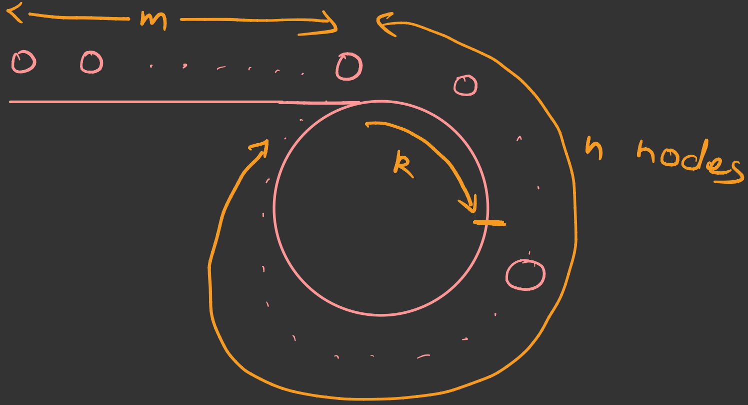
if(f == s) → Cycle in LL

② Count length of cycle

```
c=1
s ⇒ f.next

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

③ Starting point of the cycle



$j < i$

$$\text{dist by fast} = m + in + k$$

$$\text{dist by slow} \Rightarrow m \quad jn + k$$

$$dist_f = 2 \times dist_s$$

$$m + in + k = 2(m + jn + k)$$

$$m + in + k = 2m + 2jn + 2k$$

$$\underbrace{(1 - 2j)}_{x} n = m + k$$

$$xn = m + k$$

$$xn - k \Rightarrow m$$