

CS & IT ENGINEERING

Data Structure



Linked List
Chapter- 3
Lec- 06



By- Pankaj Sharma sir

TOPICS TO BE
COVERED

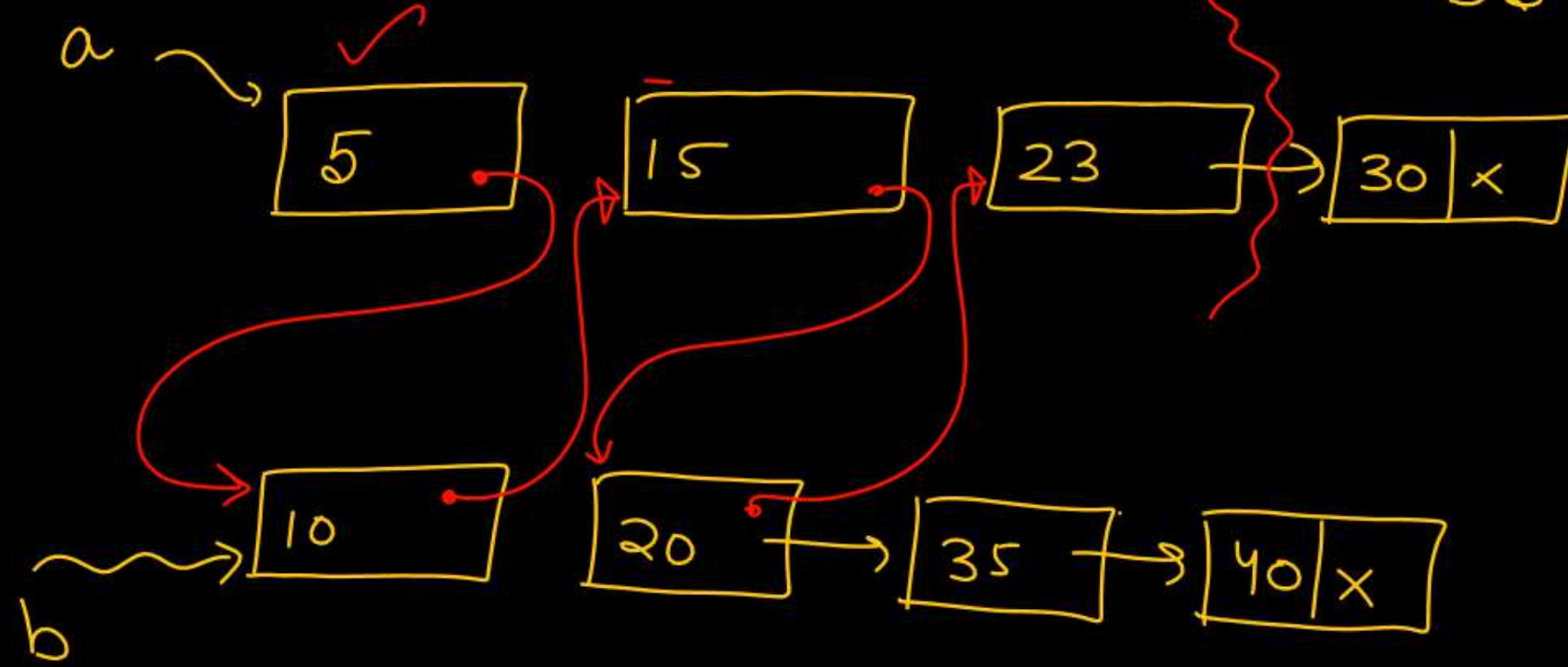
Linked List-VI

1.

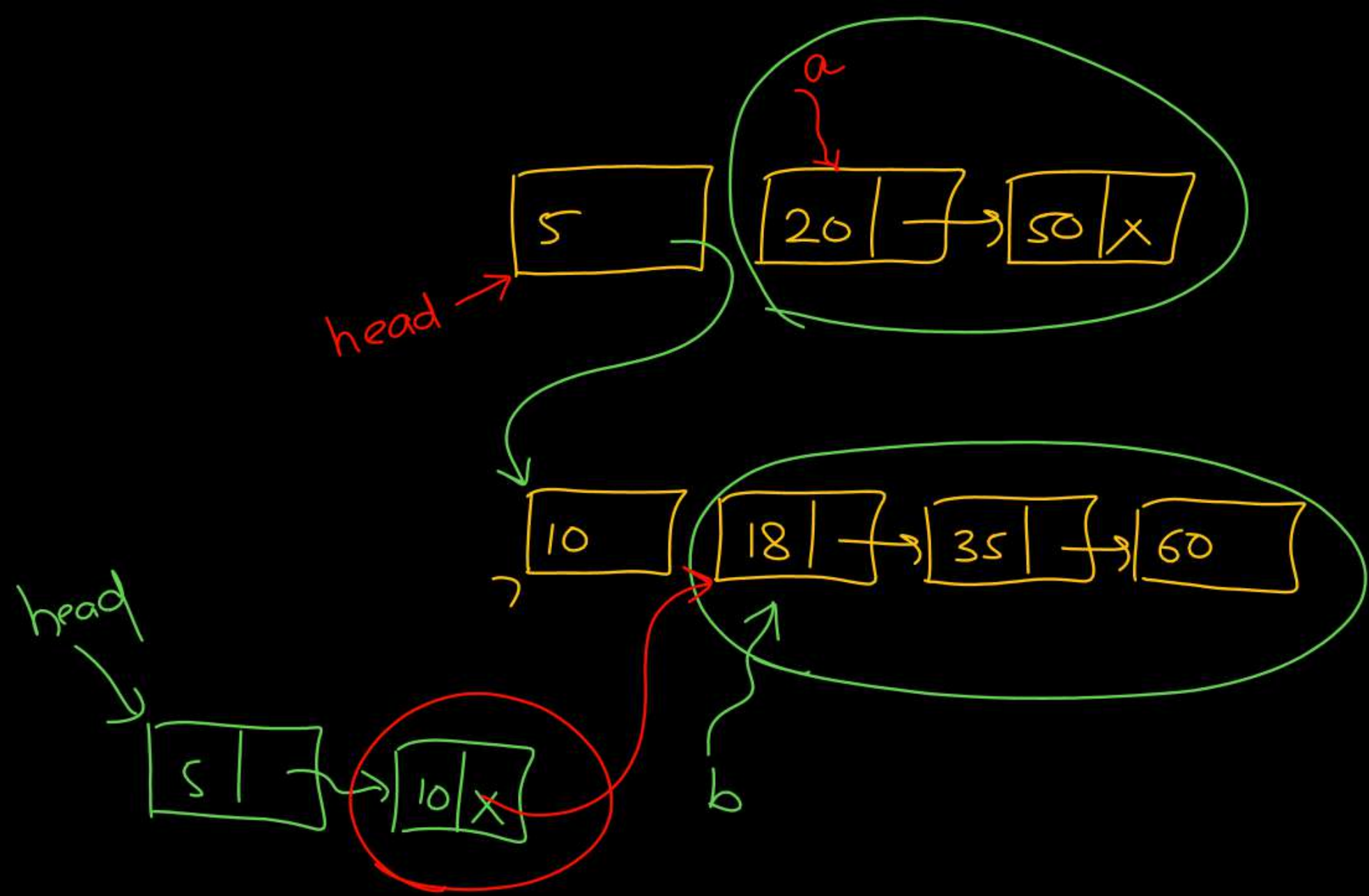
Merge 2 Sorted linked list

8:30 ⇒ PM
DS

l.l ⇒ 1st node
Min data



head : First node in res. l.l.
tail : Last node in res. l.l.



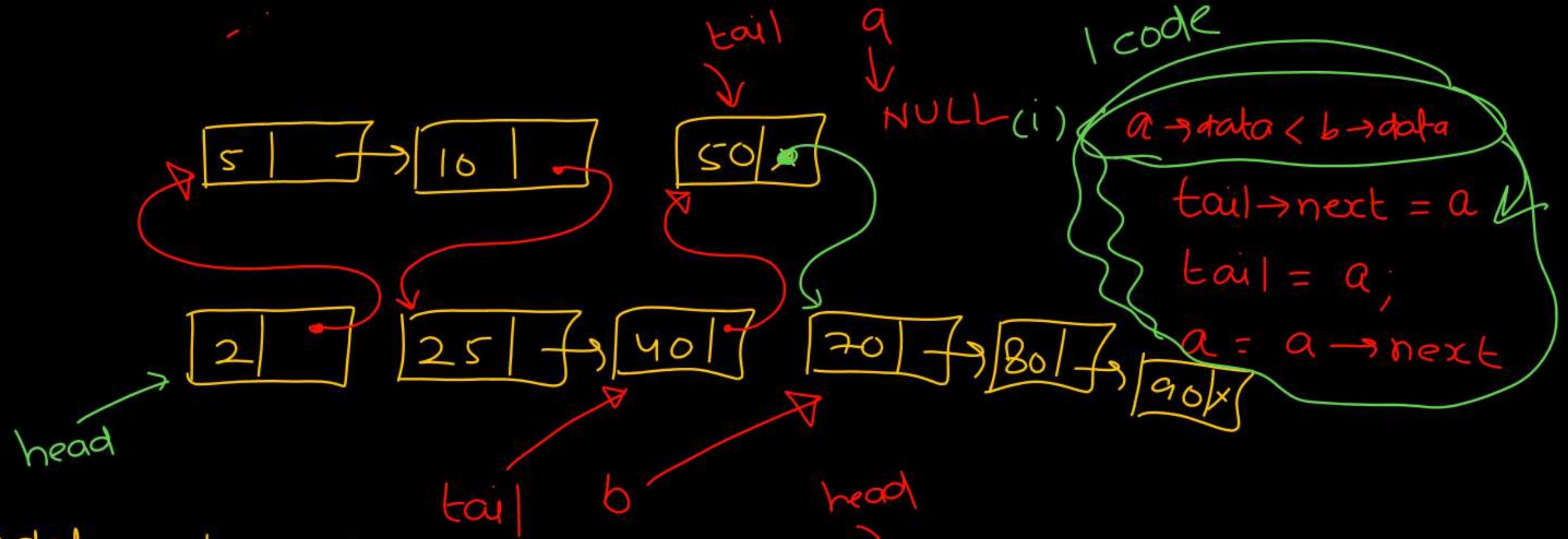
Q



o/p: return a

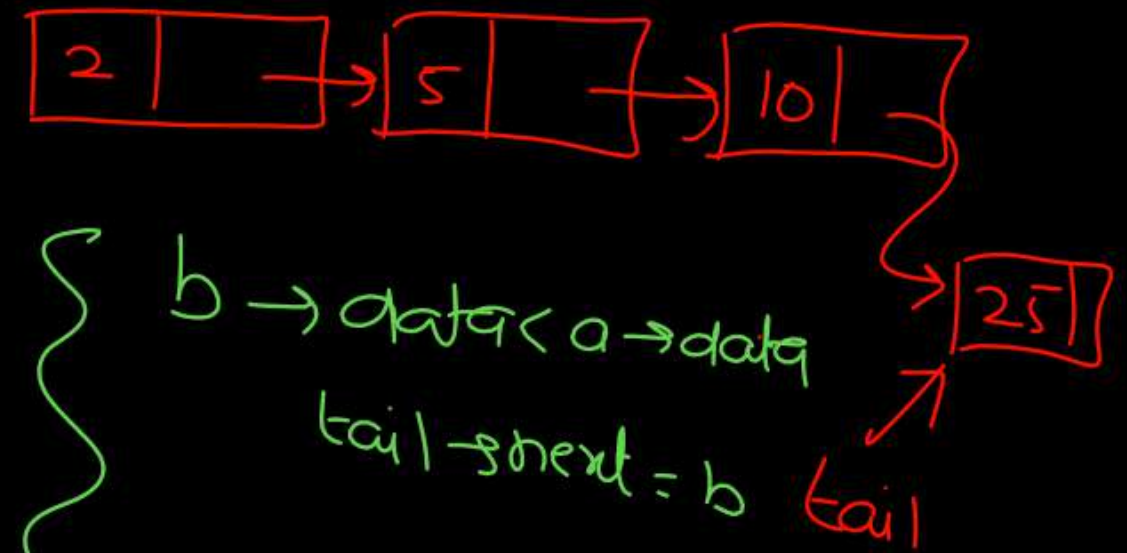


o/p return b



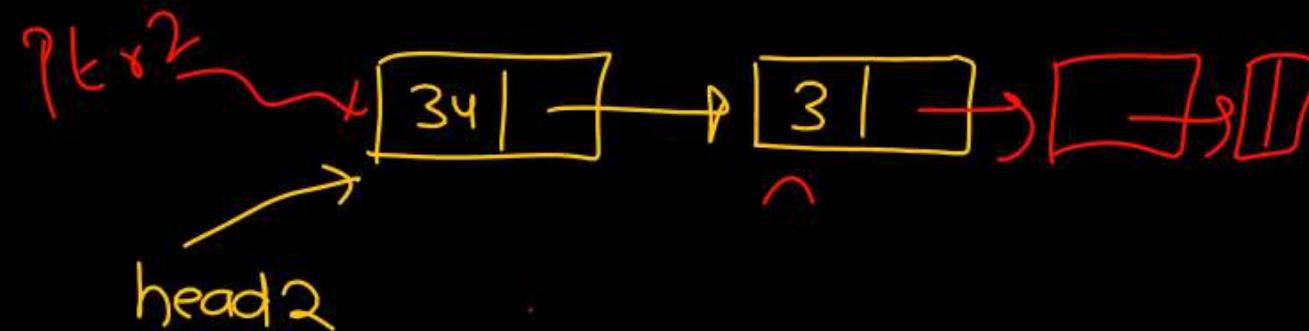
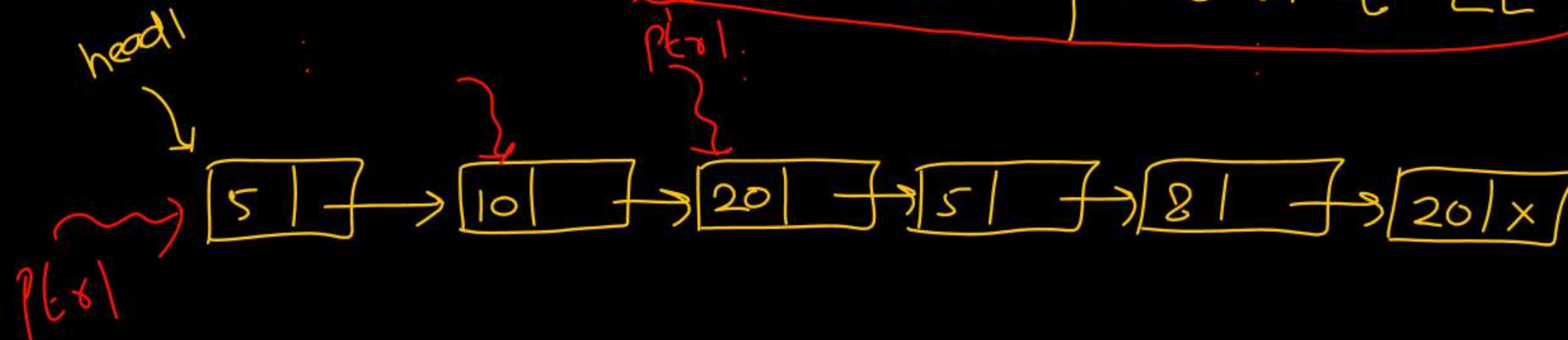
```

if
{
  a->data < b->data
  {
    head = tail = a;
    a = a->next;
  }
else
{
  head = tail = b;
  b = b->next;
}
  
```



Intersection point in 2 LL

$d=2$



int d = abs(c1 - c2);
if (c1 > c2)
d times
ptr1 → update

else
d times
ptr2 → update

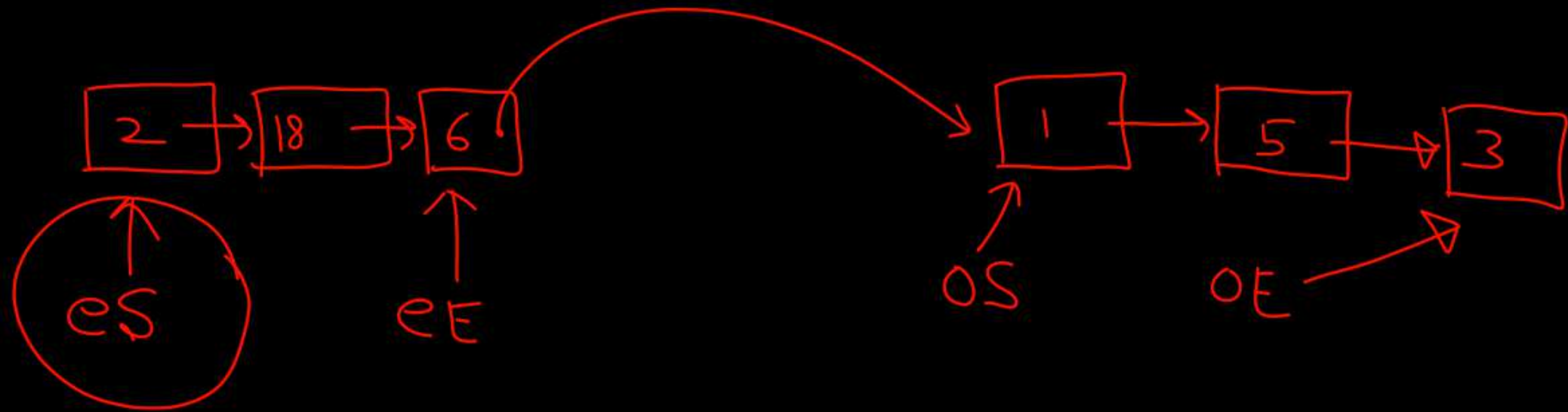
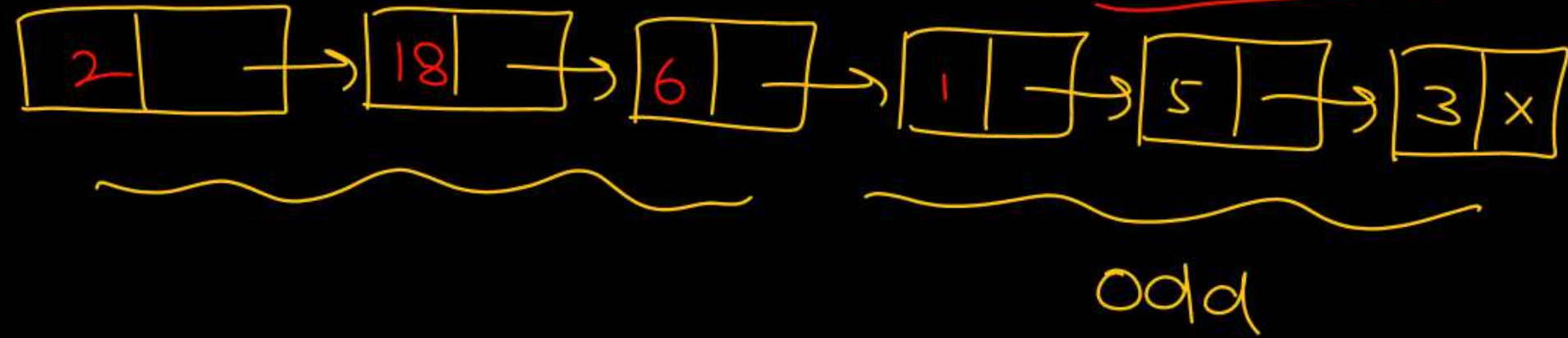
```
c1 = 0;
ptr1 = head1;
while (ptr1)
{
    c1++;
    ptr1 = ptr1->next;
}
```

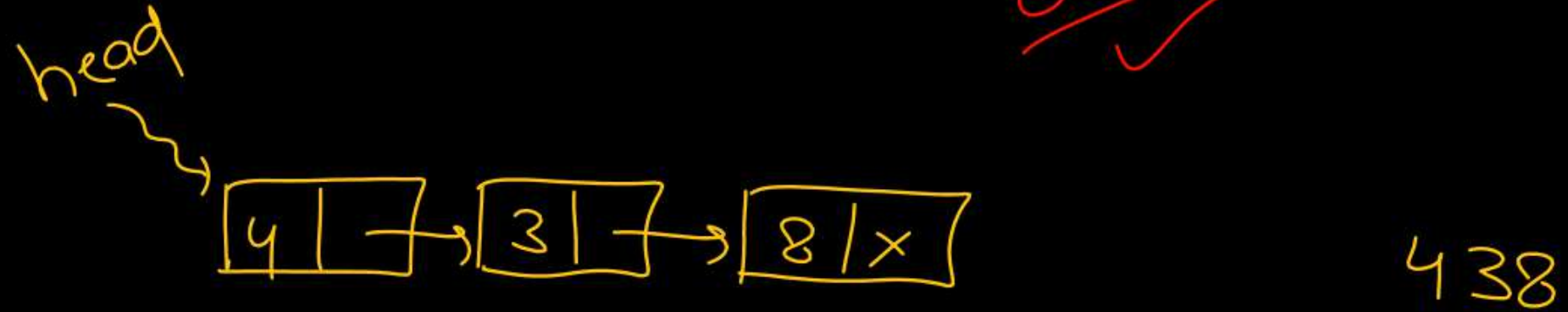
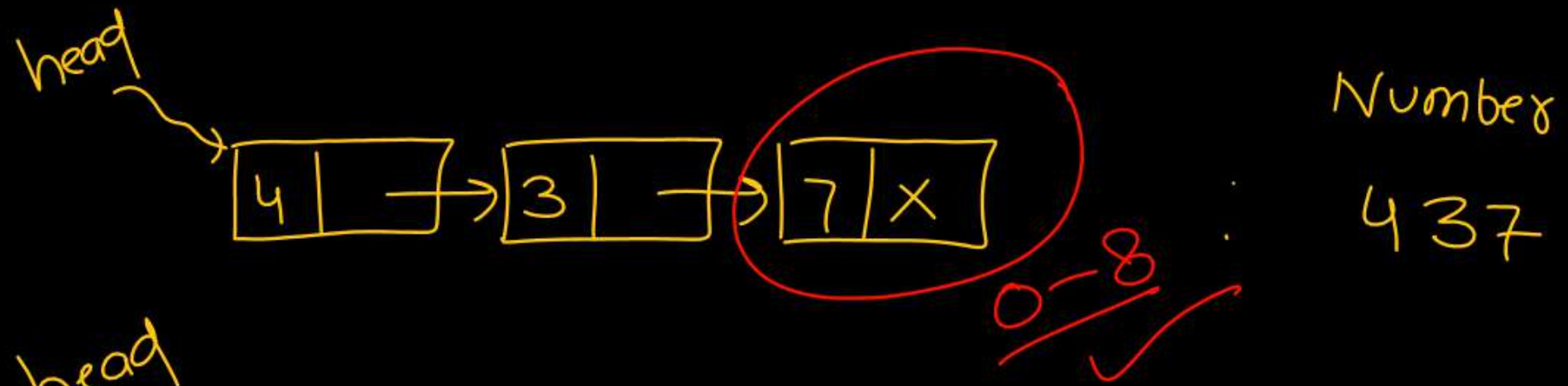
```
c2 = 0;
ptr2 = head2;
while (ptr2)
{
    c2++;
    ptr2 = ptr2->next;
}
```



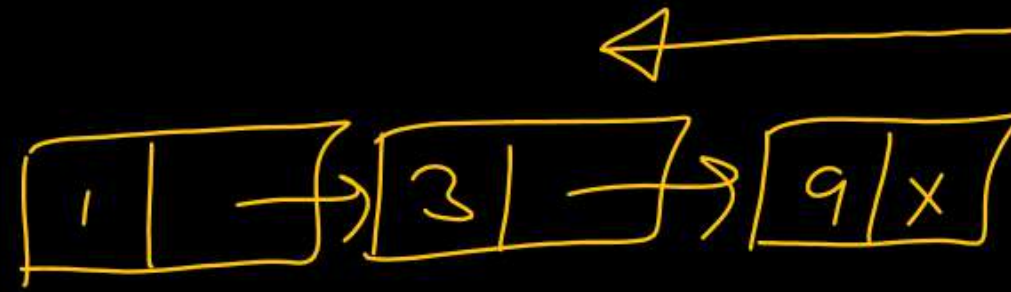
Segregate Even & odd in a LL
gfg Practice

O/p :



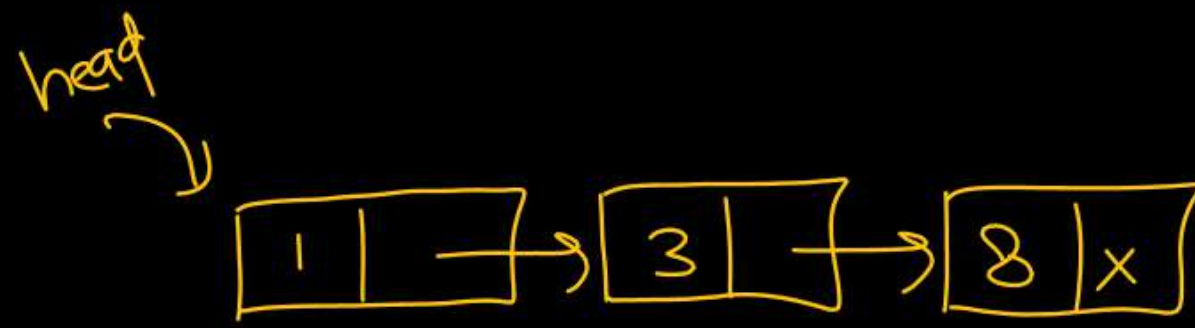


Case 1: last node ⇒ 0-8 ✓ +1 ✓

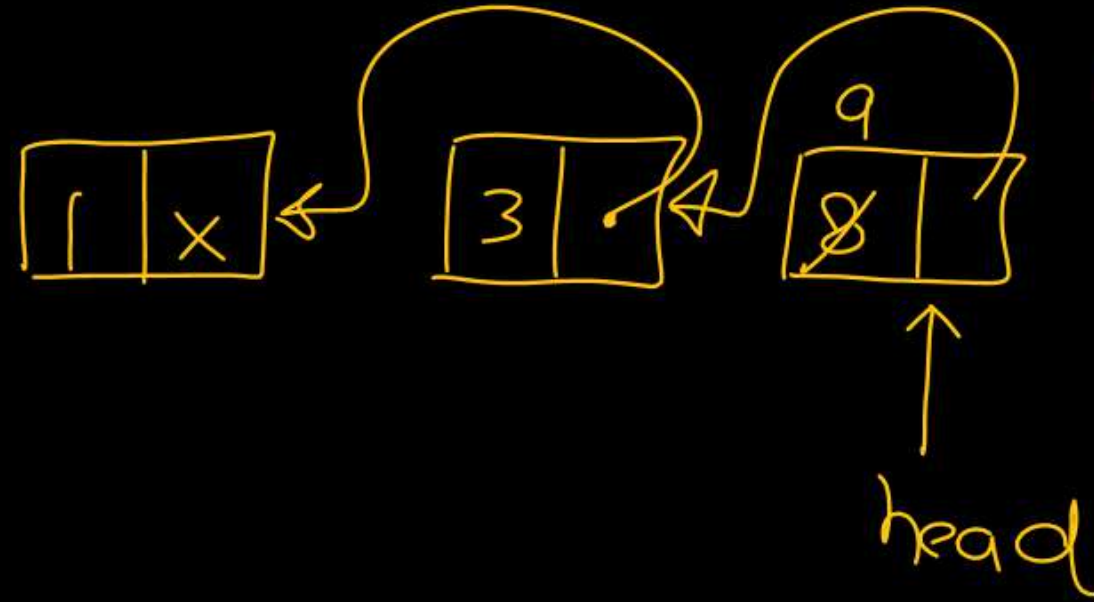


S.L.L

reverse



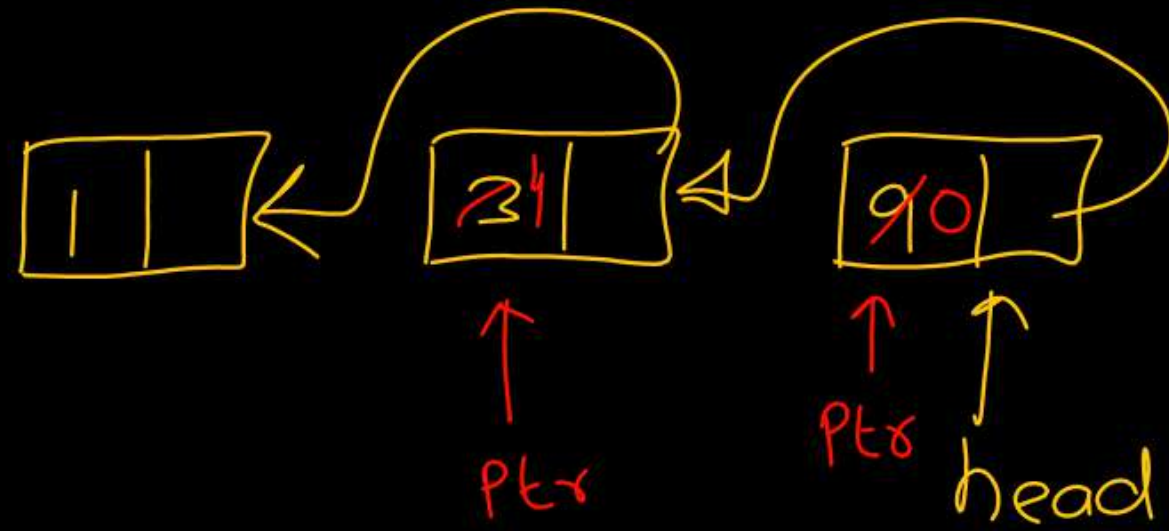
① head = reverse(head)



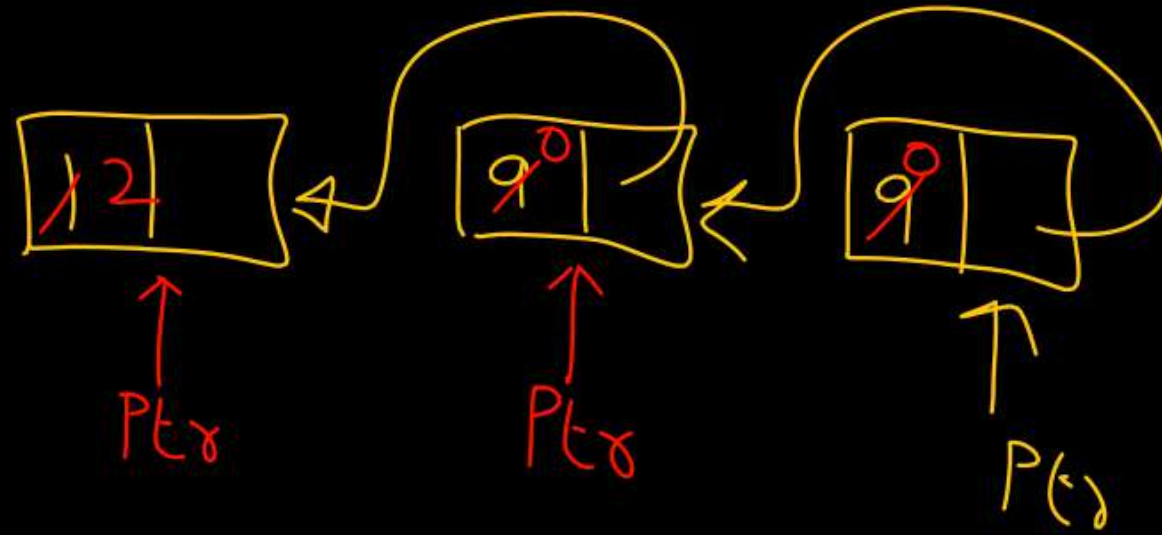
② if ^{now} first node

0-8
↓
inc by 1
and

③ reverse the LL & return head



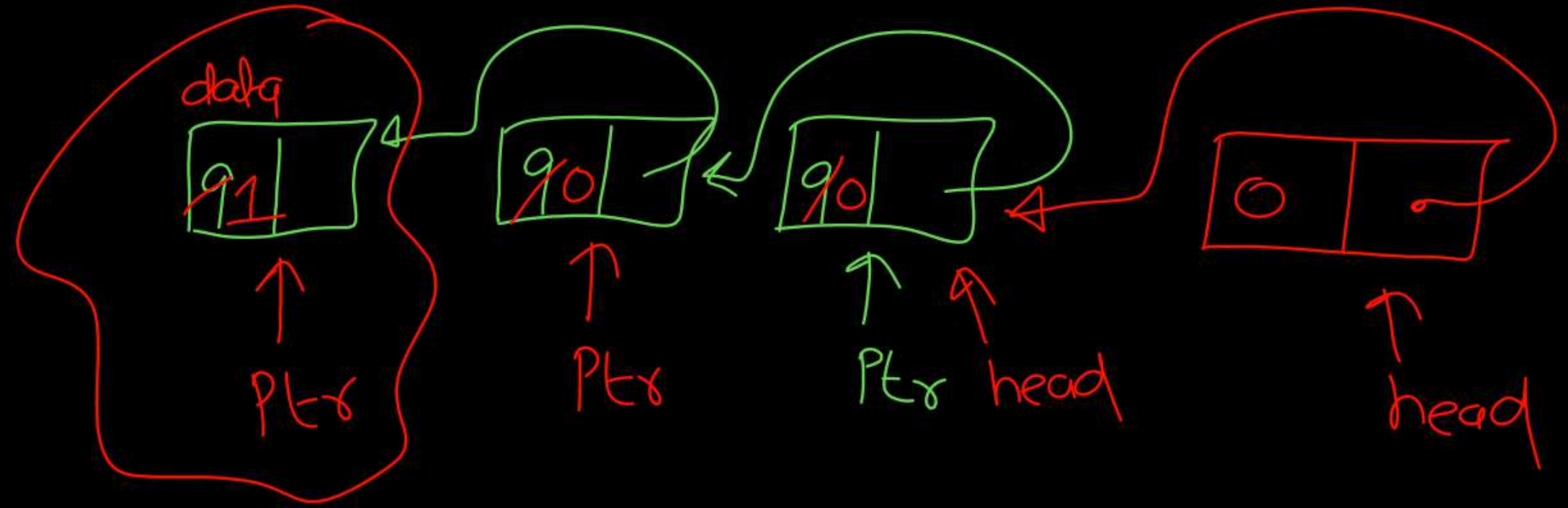
reverse





999

→ 1000



Special
case

$\text{Ptr} \rightarrow \text{data} == 9 \ \&\& \ \text{Ptr} \rightarrow \text{next} == \text{NULL}$

~Add 1 to Linked list

Medium-hard

