CS & IT ENGINEERING

Data Structure

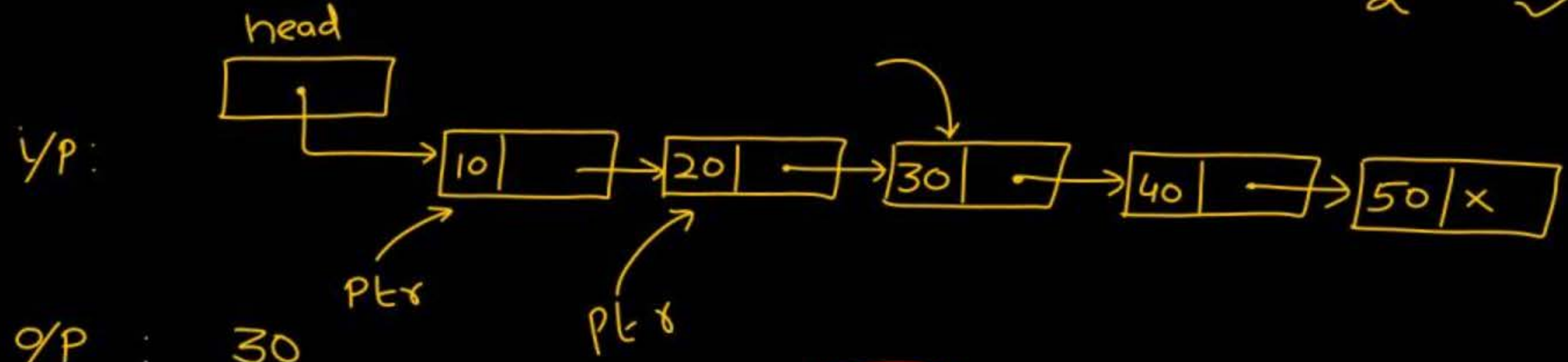
Linked List
Chapter- 3
Lec- 04

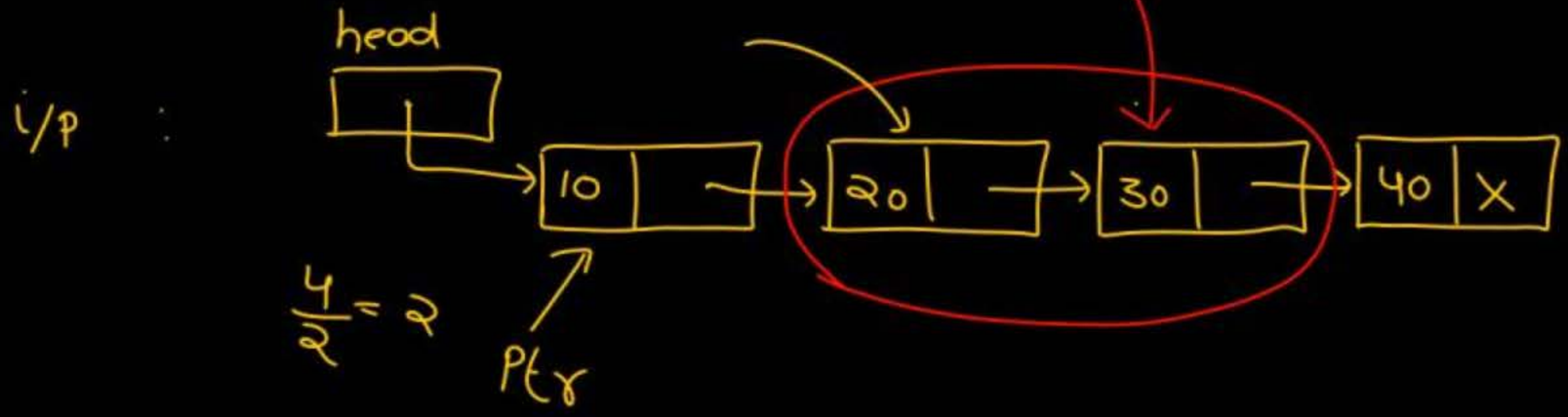
By- Pankaj Sharma sir

Linked List-IV

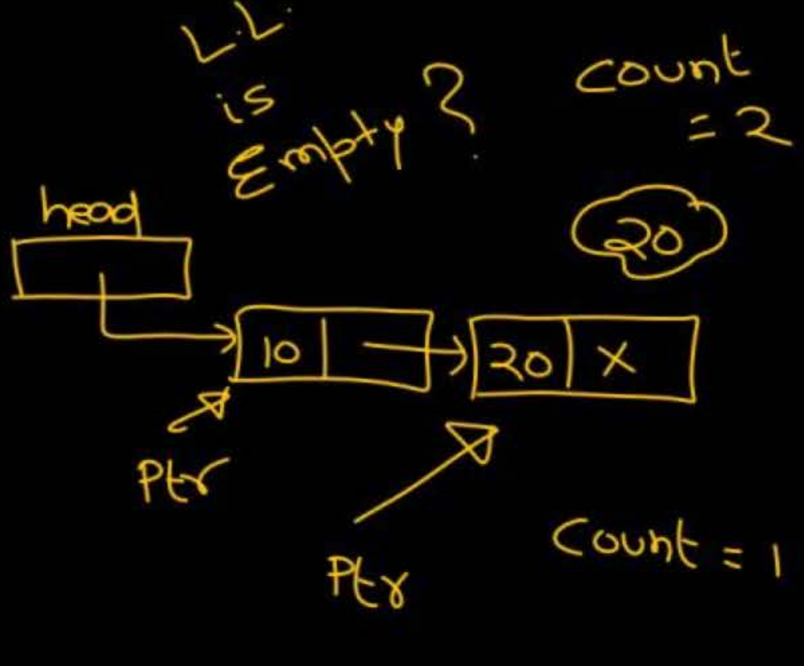
Finding Mid Element in given linked list:

$\frac{5}{2} = 2$





① No. of nodes ⇒ count

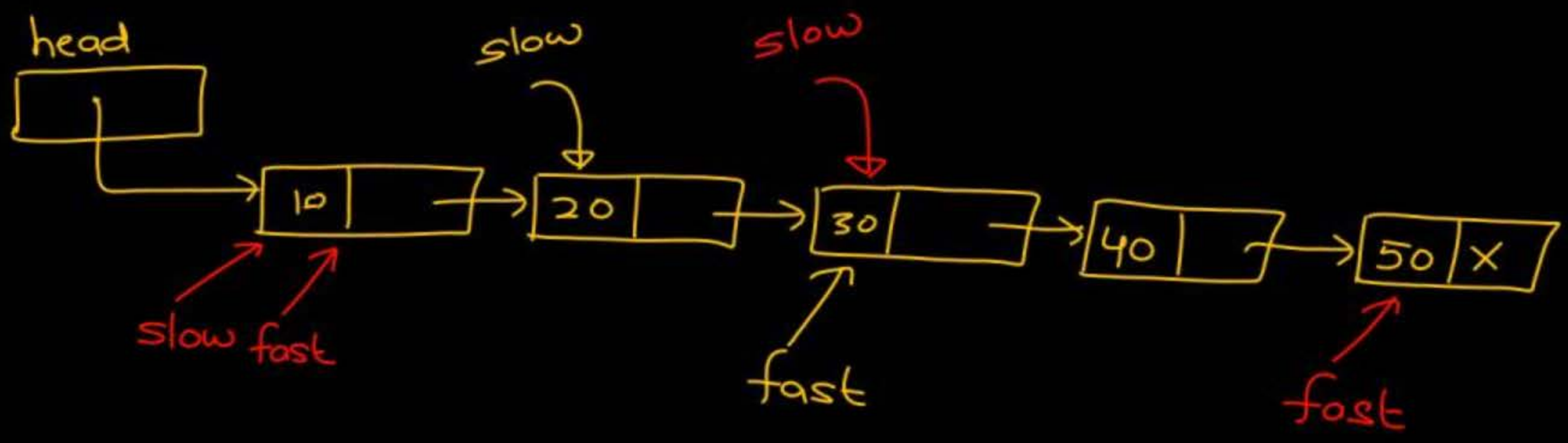



```
count = 0;
struct Node *Ptr;
Ptr = head;
While (Ptr != NULL)
  {
     count ++;
     Ptr = Ptr → Link;
  }
if (count == 0)
Ptr = head;  return;
  count = count/2;
⇒ for(i=1; i<= count; i++)
     Ptr = Ptr → Link ;
printf("%d", Ptr → data);
```

Mistake

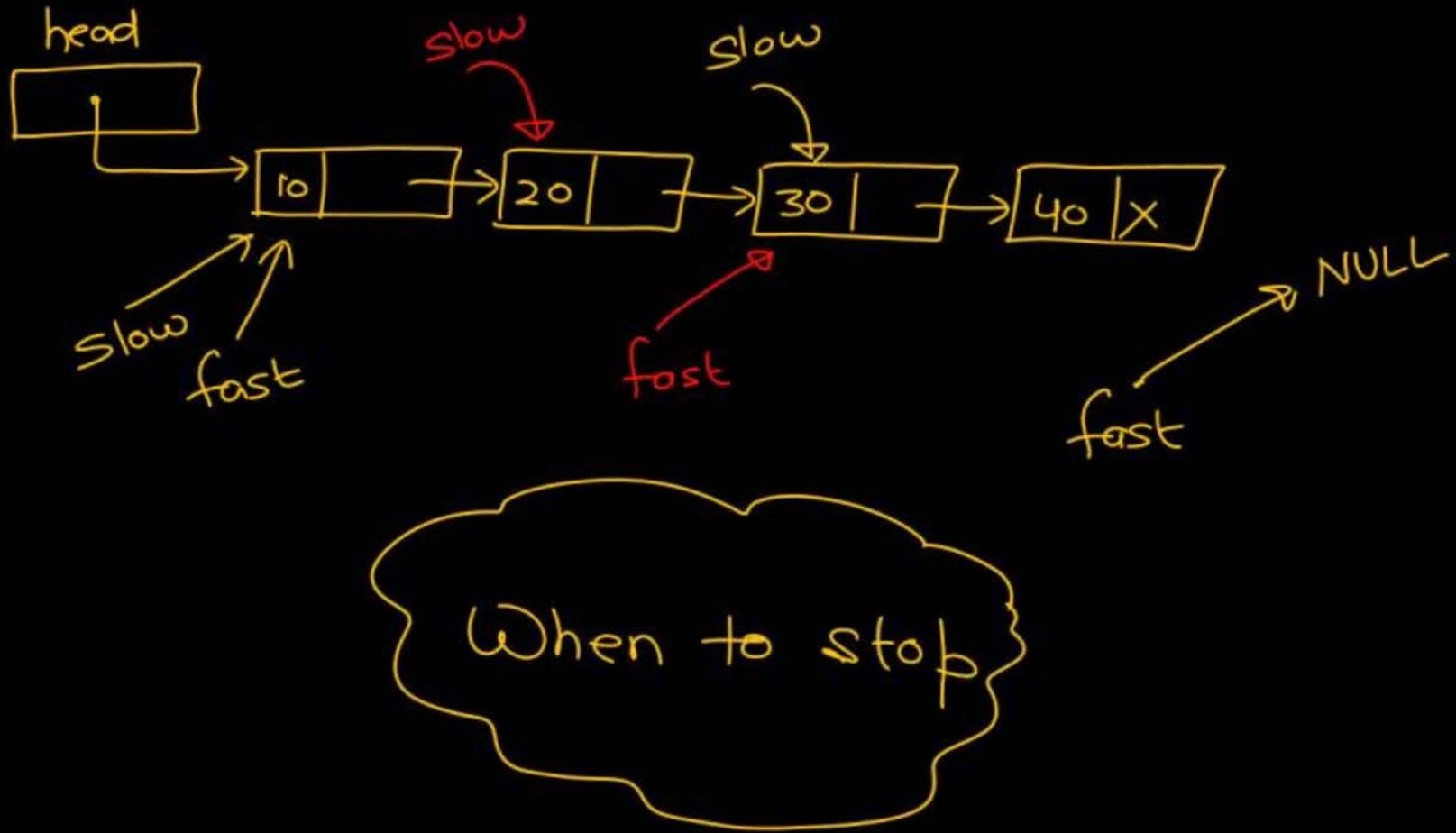
(*Ptr).data

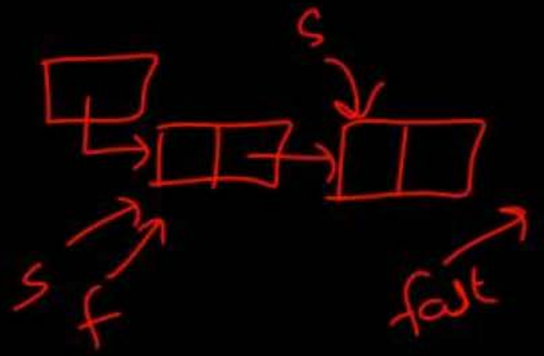
2nd approach

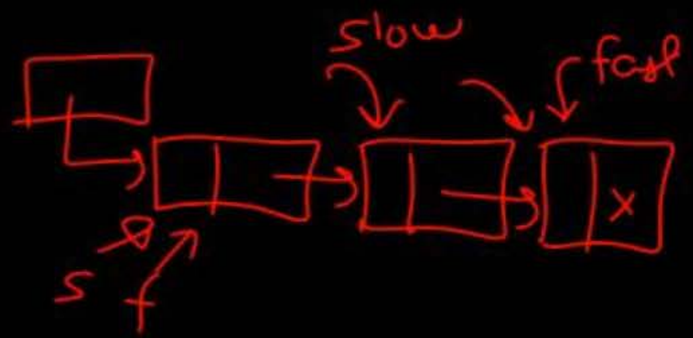


slow = slow → next ;

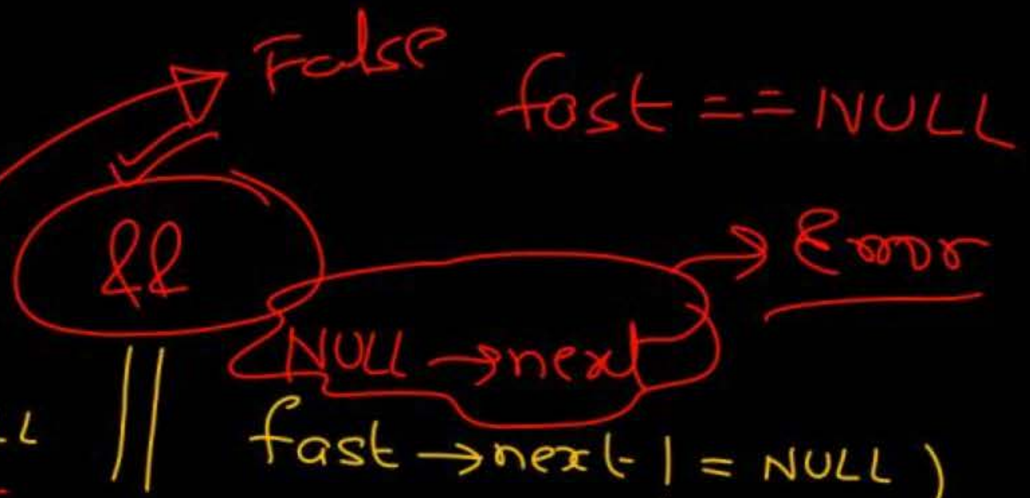
fast = fast → next → next ;

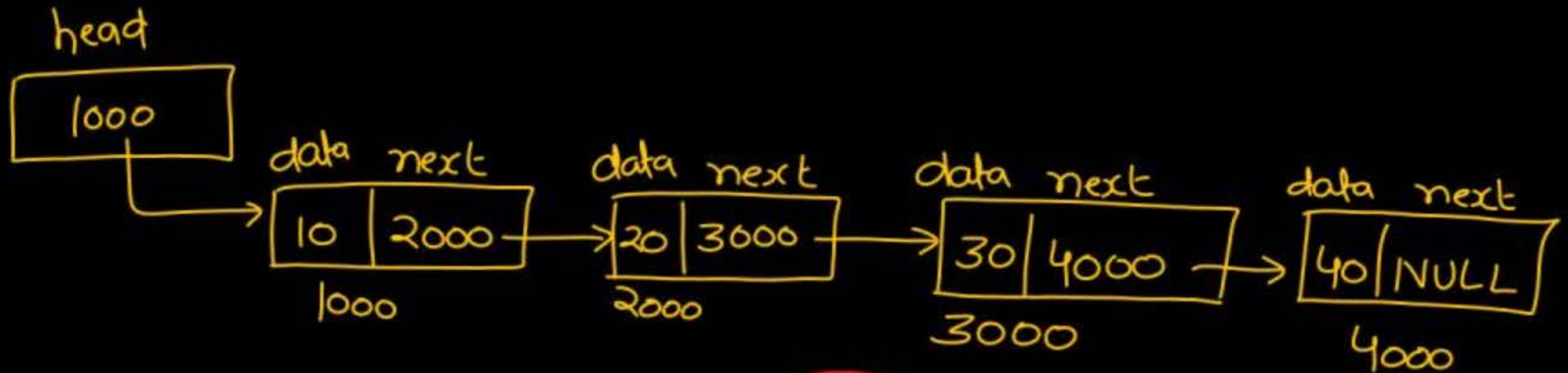

head
slow
slow
10
20
30
40
X
Slow
fast
fast
fast
NULL
When to stop

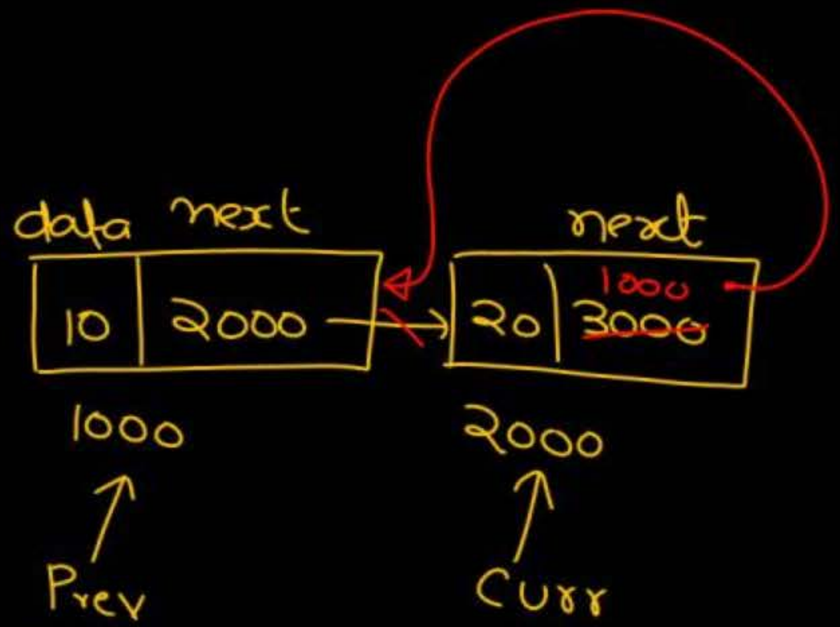





```
struct Node *slow, *fast;
slow = head;
fast = head;
if (head == NULL)
        return;
While( fast != NULL
       True
{
    slow = slow->next;
    fast = fast->next->next;
}
printf("%d", slow->data);
```

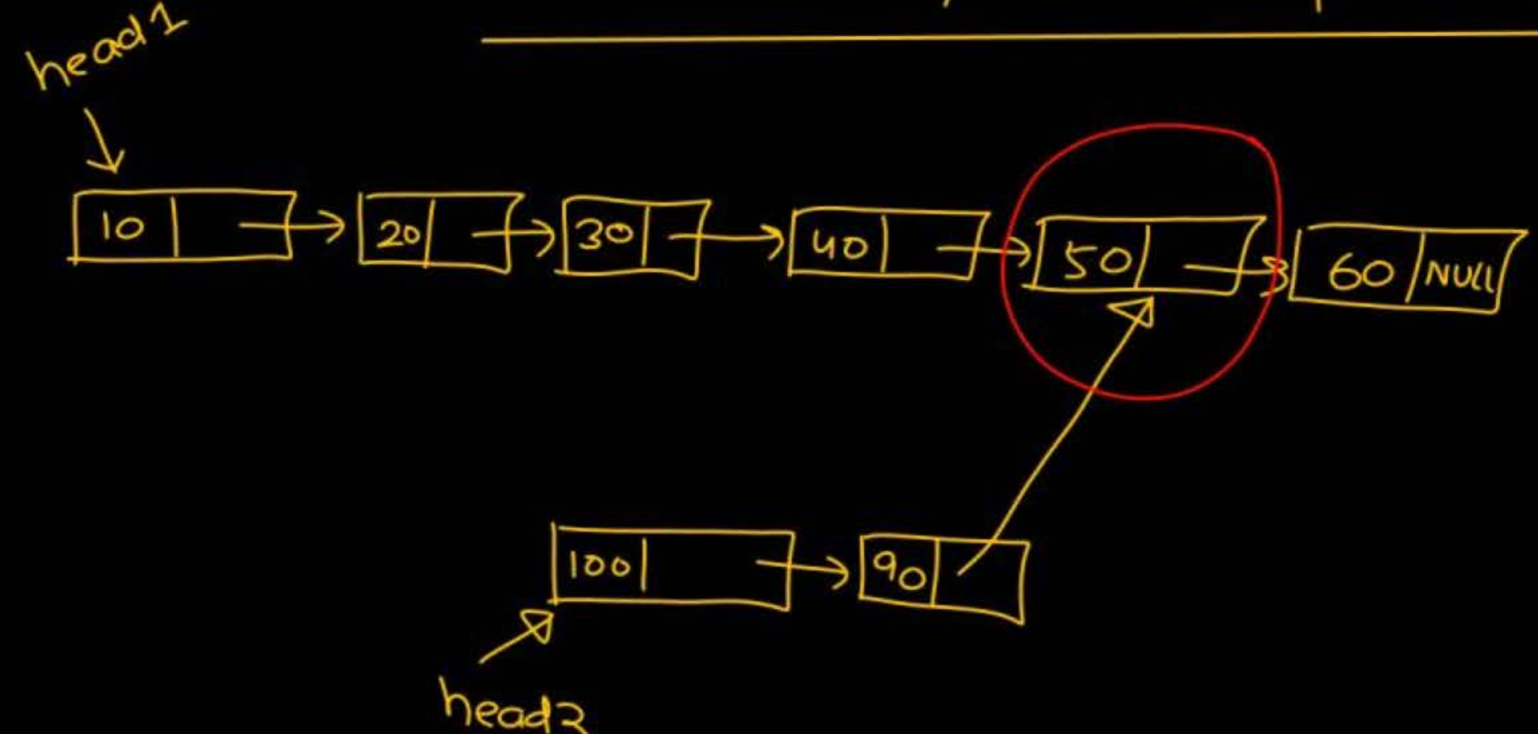




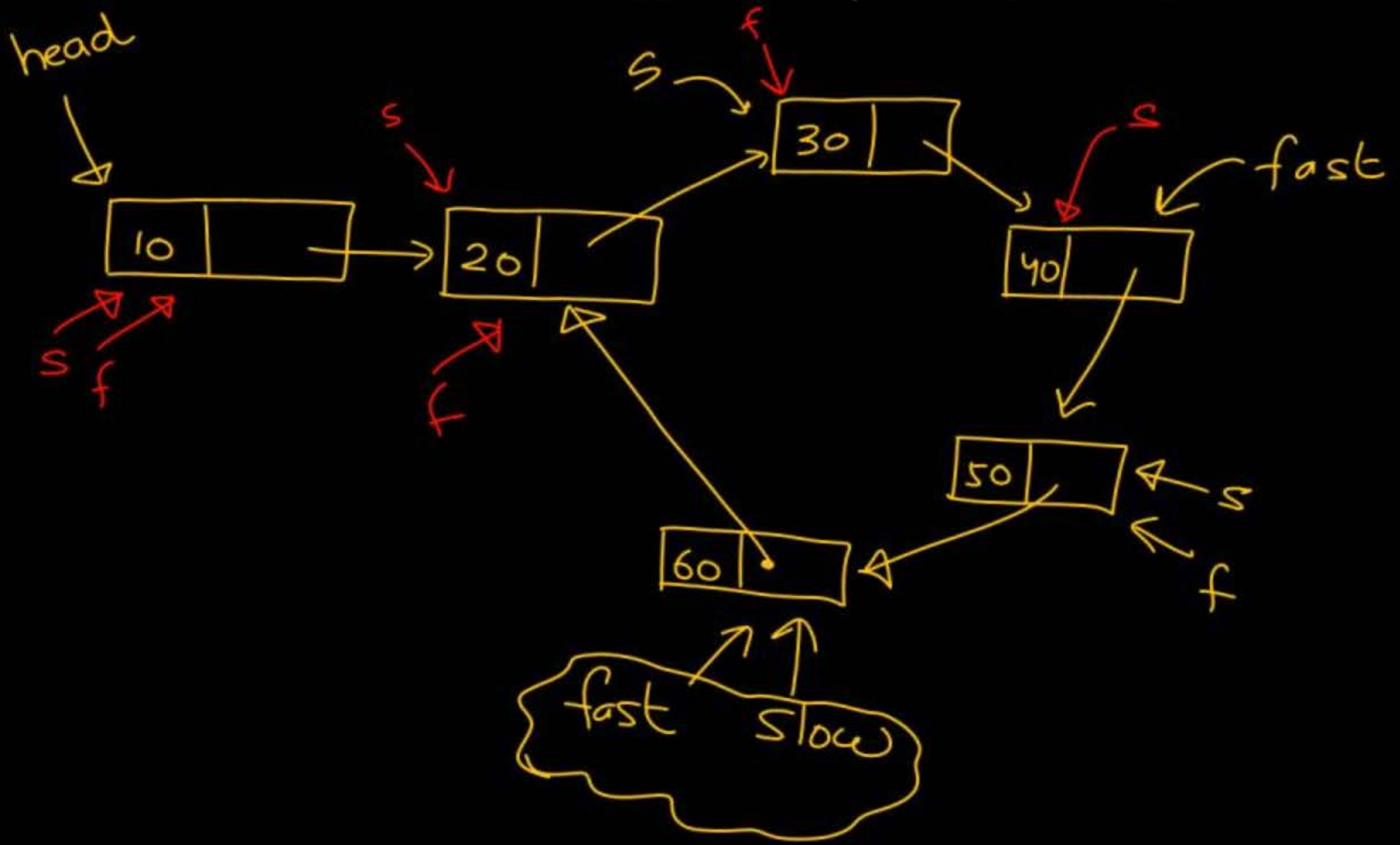


Curr → next = Prev

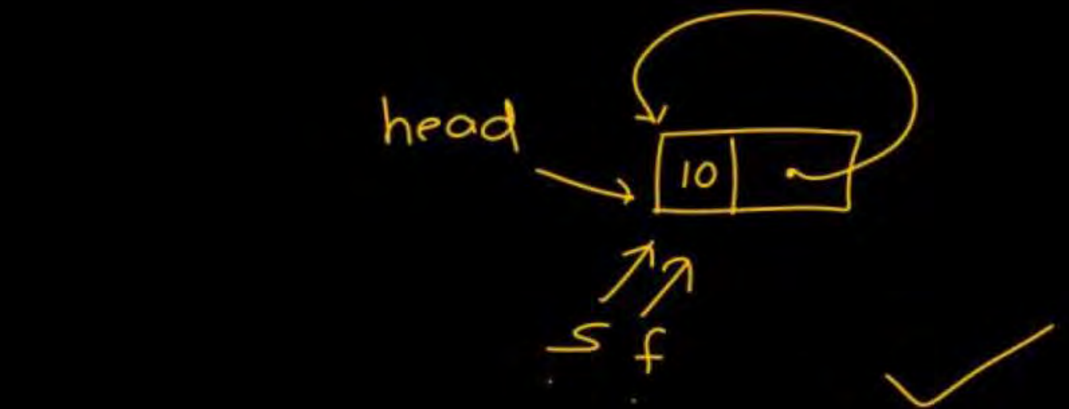
Intersection point in Linked list

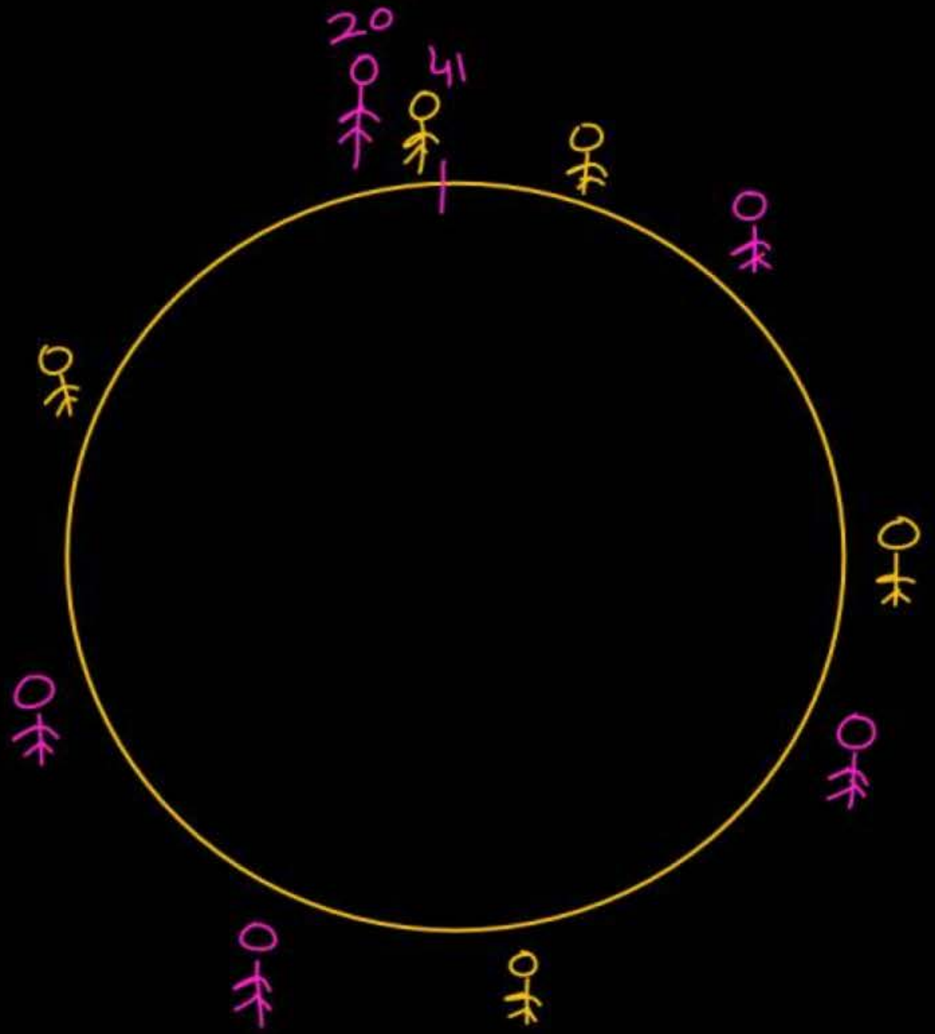


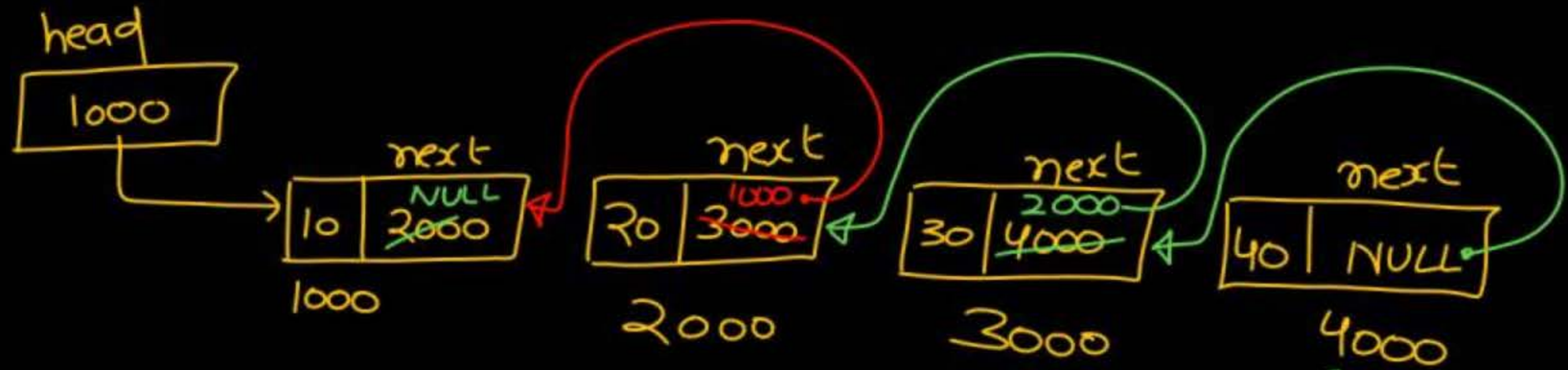
Detect a loop in linked list (O in a linked)




10
20
30
4dX


head
10
s
f


20
41



Prev = NULL ;

Curr = head ;

Prev nex = curr → next

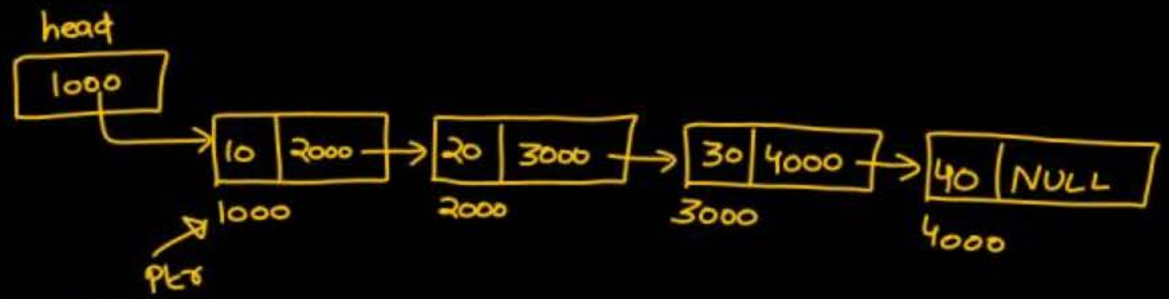
① curr → next = Prev

1) nex = curr → next ;

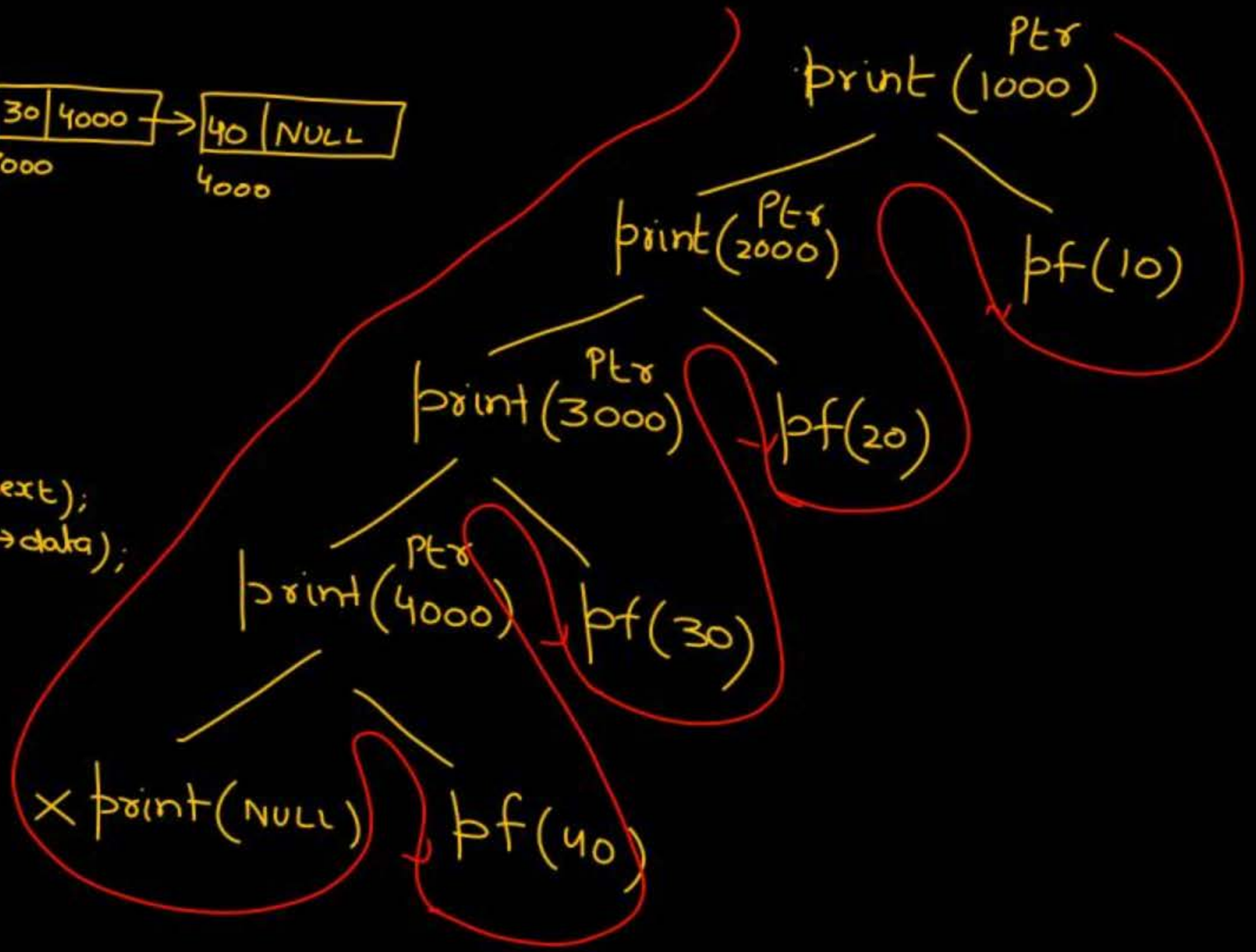
2) curr → next = Prev ;

3) Prev = curr ;

4) curr = nex ;




```
void print ( struct Node *Ptr)
    {
        if (Ptr) {
            1.  print(Ptr->next);
            2.  printf ("%d", Ptr->data);
                }
    }
```




THANK YOU GW
SOLDIERS !