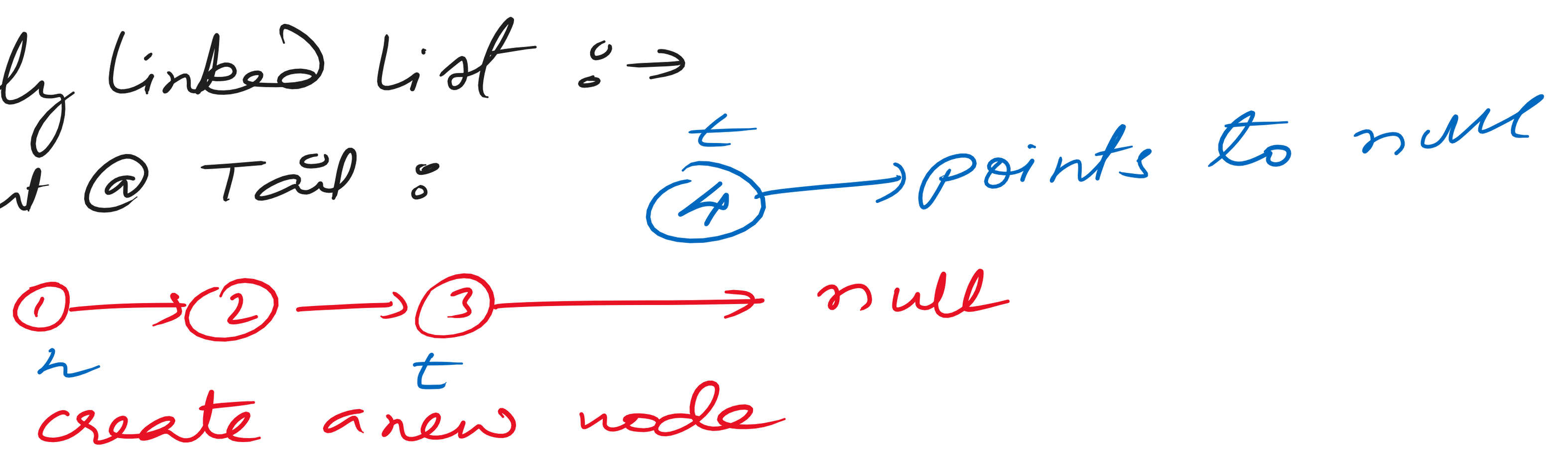
Singly Linked List :→

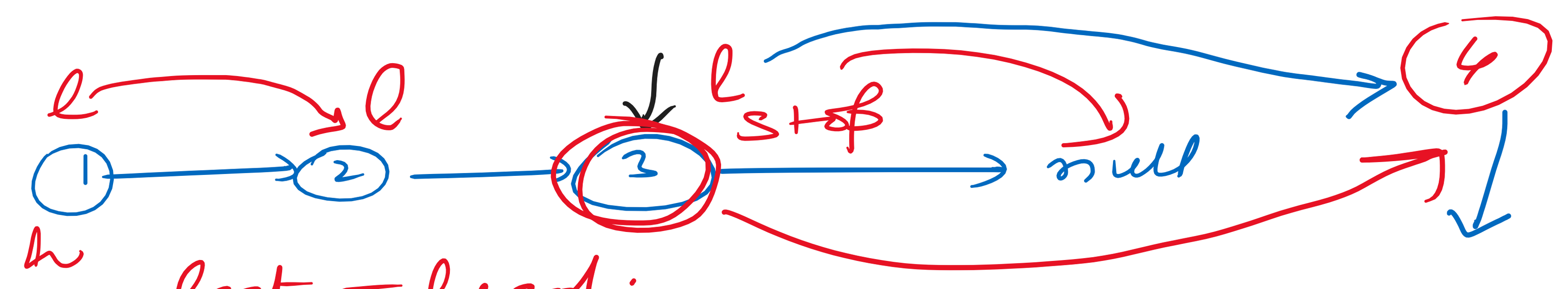
Insert @ Tail :



① create a new node

* Cases : Empty (That node becomes head)

head is null .

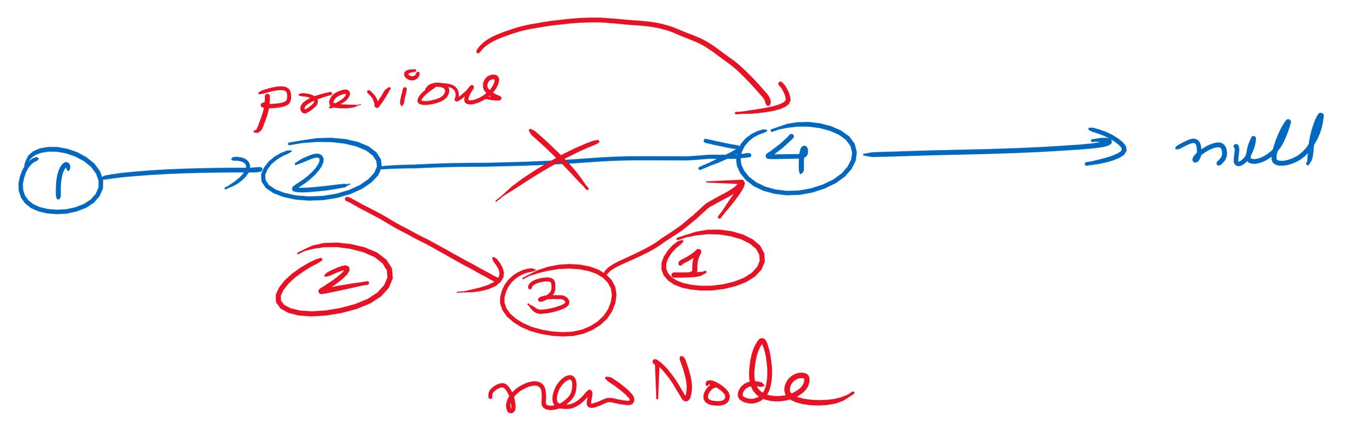


last = head;

while (last. next != null) Stop

last = last. next

last. next = newNode



newNode → next = previous. next

prev . next = new ...

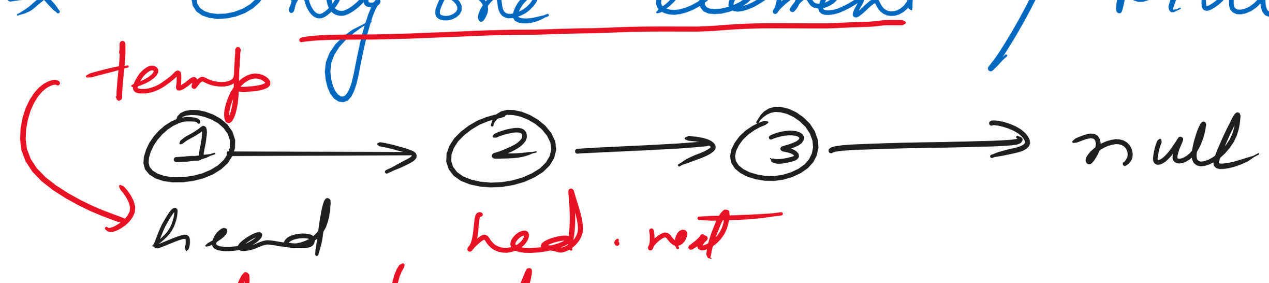
1. Middle node of linked List
2. Reverse a linked List
3. Linked List palindrome
4. Merge 2 sorted L Lists

Delete Operations in SLL :

① Delete Head

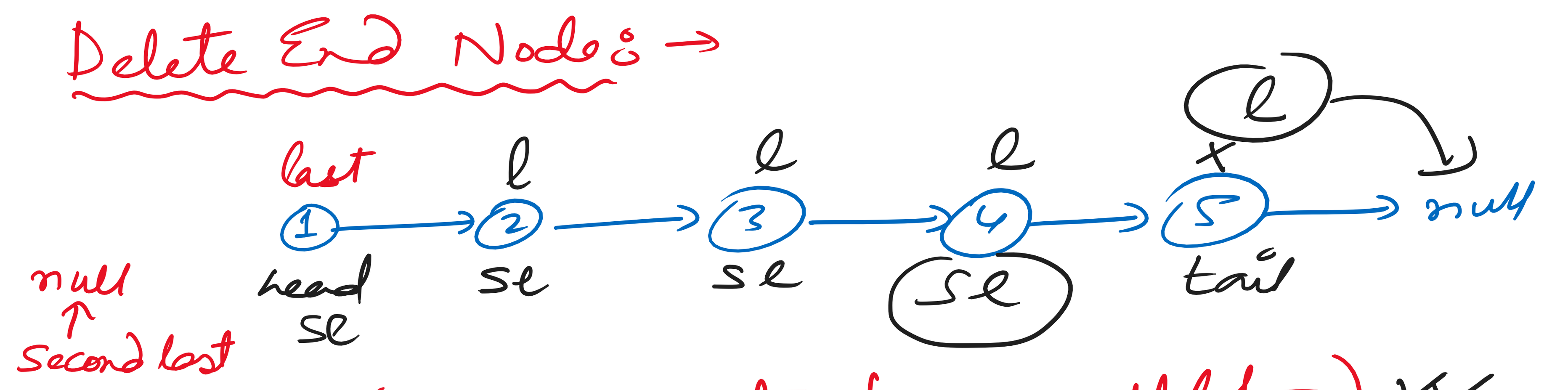
* Empty list → can't delete

* Only one element / Multiple



delete temp.

Delete End Node :→



while (last → next != null)

l = h   {

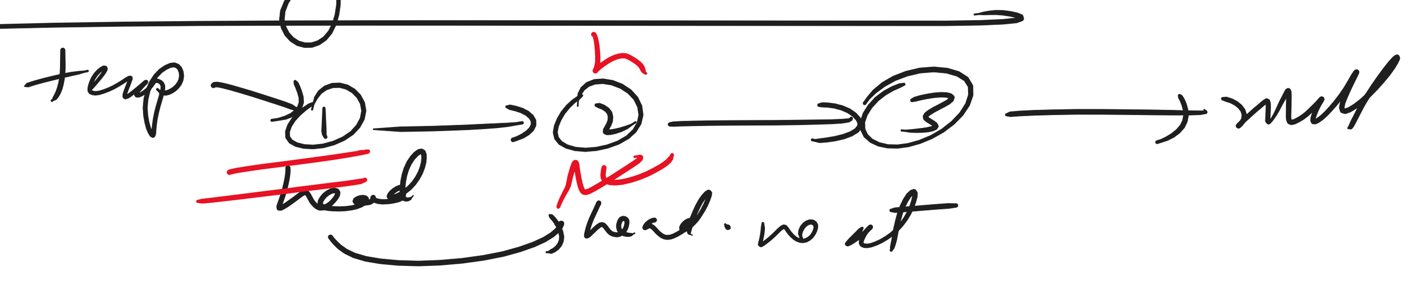
sl = n

second last = last ;

last = last → next;

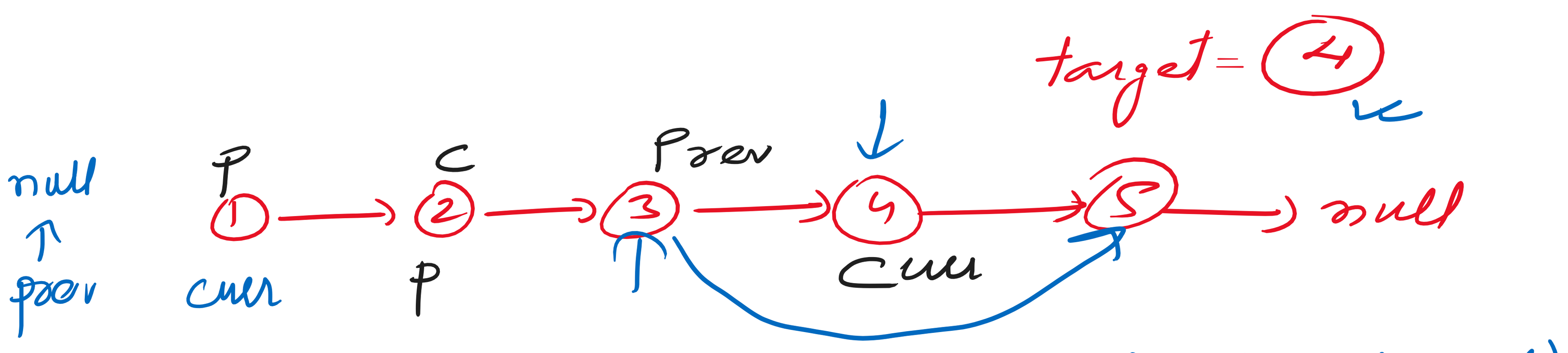
}

Delete Target Node:



delete ①

delete temp



while (curr != null && curr → data != target)

{

prev = curr ;

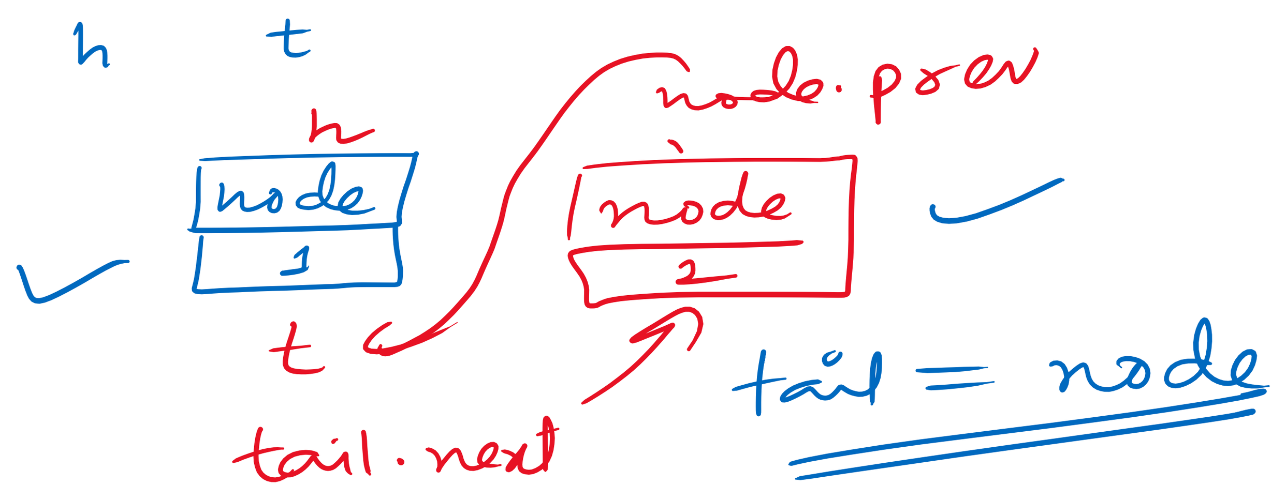
curr = curr → next;

}

P. next = curr . next

delete curr ;

Stop



tail = node