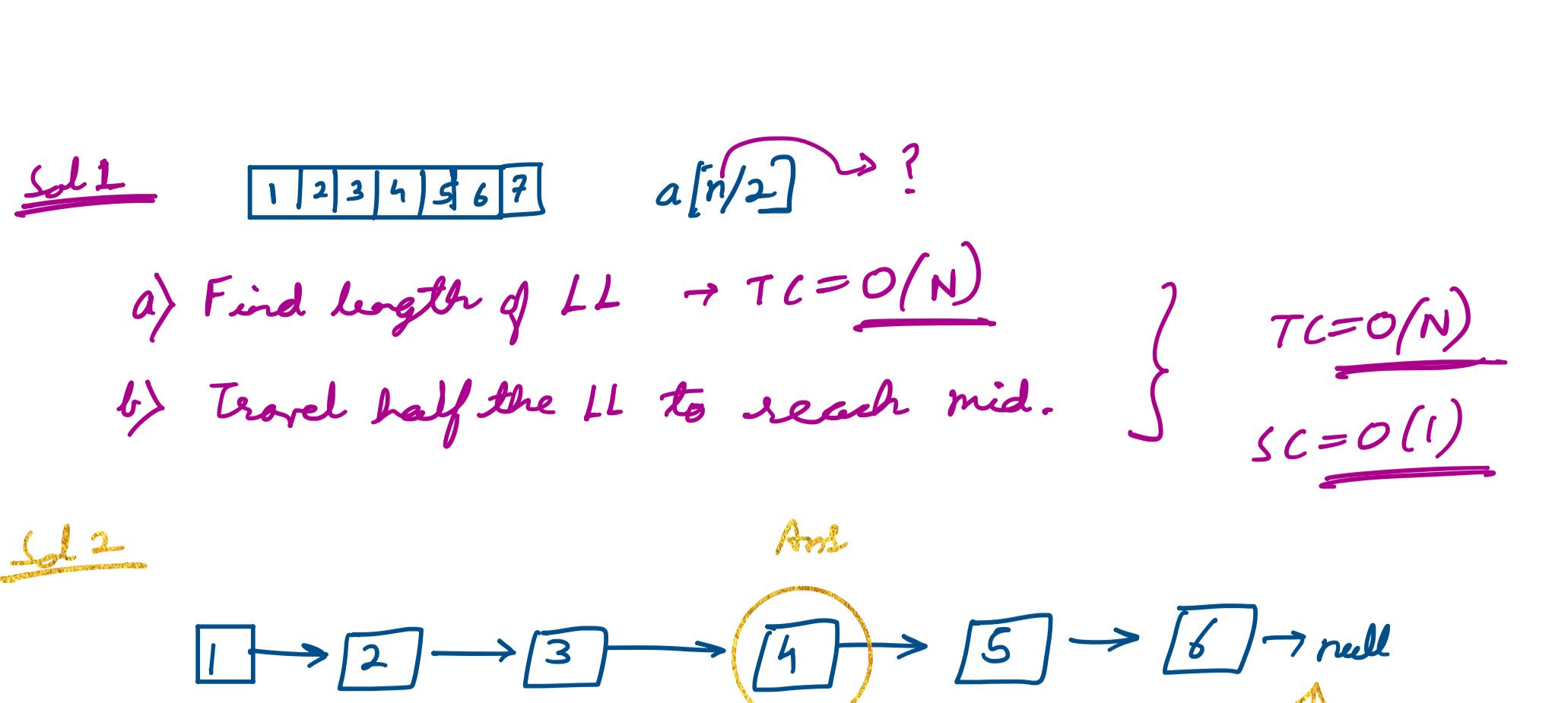


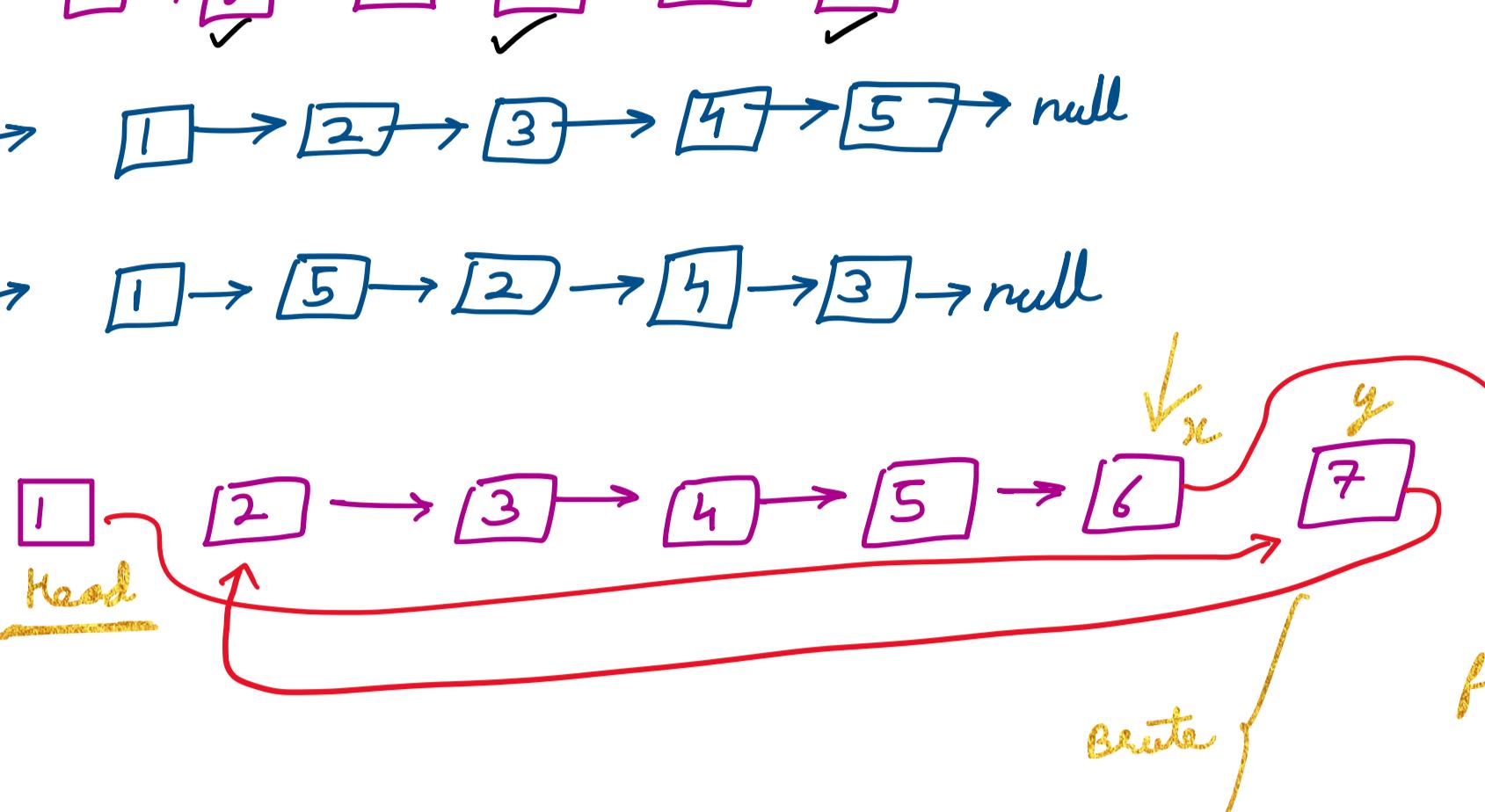
Q) Find middle element of LL.



SLL $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ $a[n/2] \rightarrow ?$

- a) Find length of LL $\rightarrow TC = O(N)$
 b) Travel half the LL to reach mid. $\left\{ \begin{array}{l} SC = O(1) \\ TC = O(N) \end{array} \right.$

DLL



$$TC = O(N) \quad \# \text{nodes in LL}$$

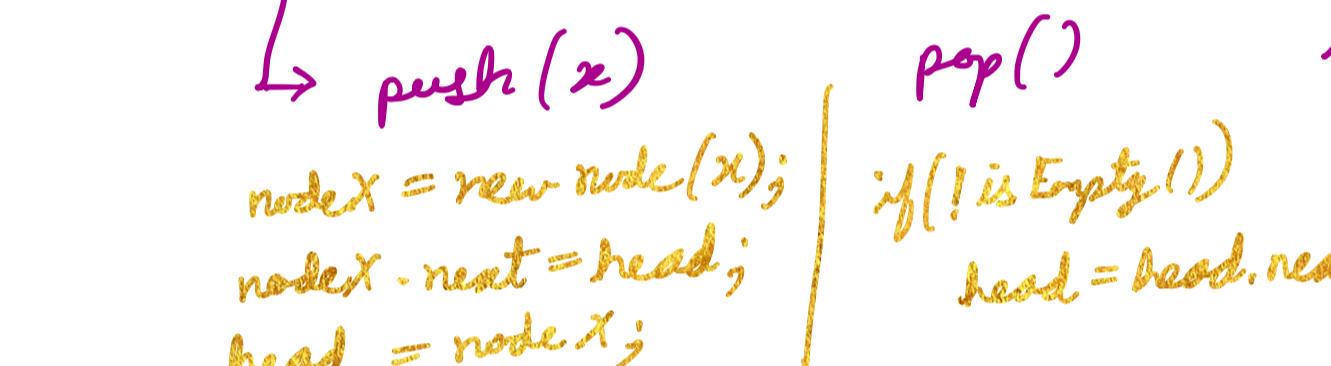
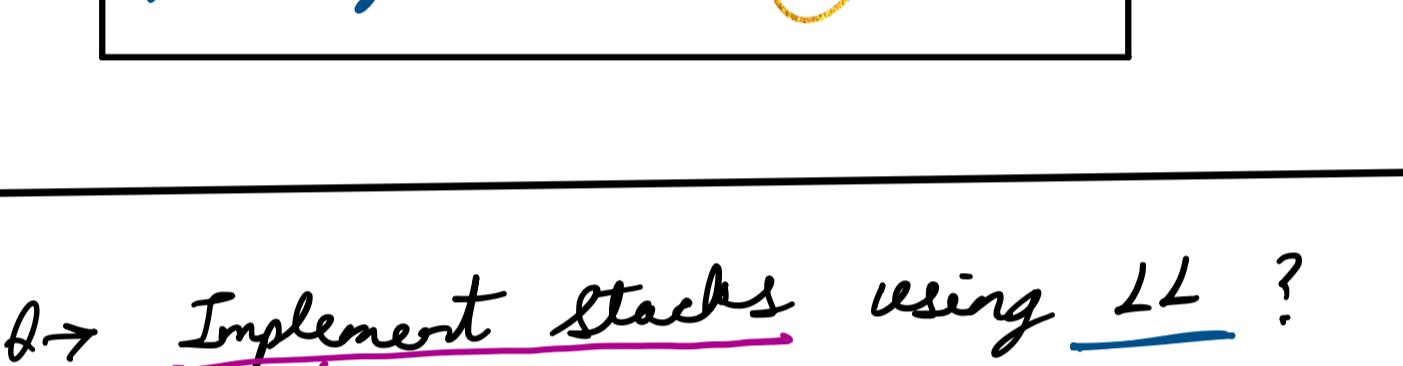
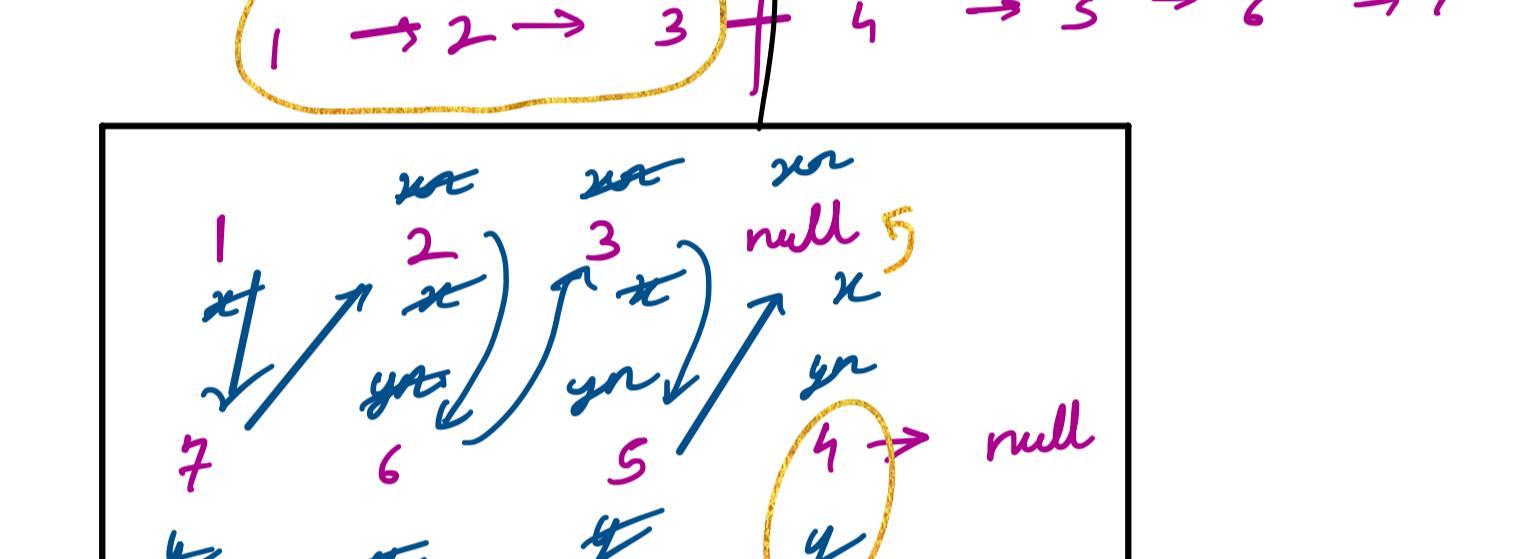
$$SC = O(1)$$

while($f \neq \text{null}$ & $f.\text{next} \neq \text{null}$) {
 s = s.next;
 f = f.next.next;
 return s;

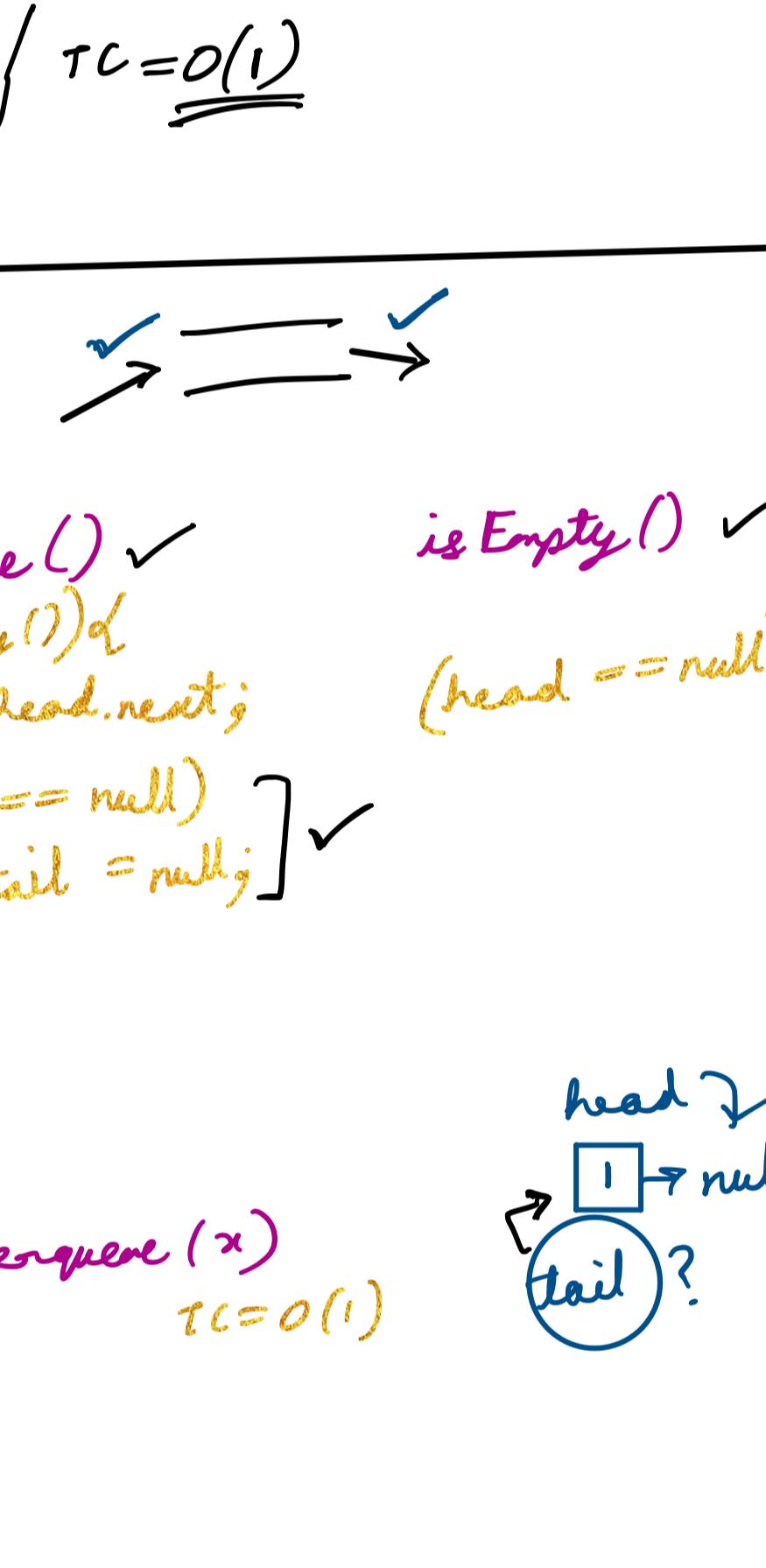
Q) Reorder List



Ans $\rightarrow 1 \rightarrow n \rightarrow 2 \rightarrow n-1 \rightarrow 3 \rightarrow n-2 \rightarrow \dots$



Sol
 a) Find mid element $- TC = O(N)$
 b) Reverse second half. $- TC = O(N)$
 c) Merge the 2 list. $- TC = O(N)$

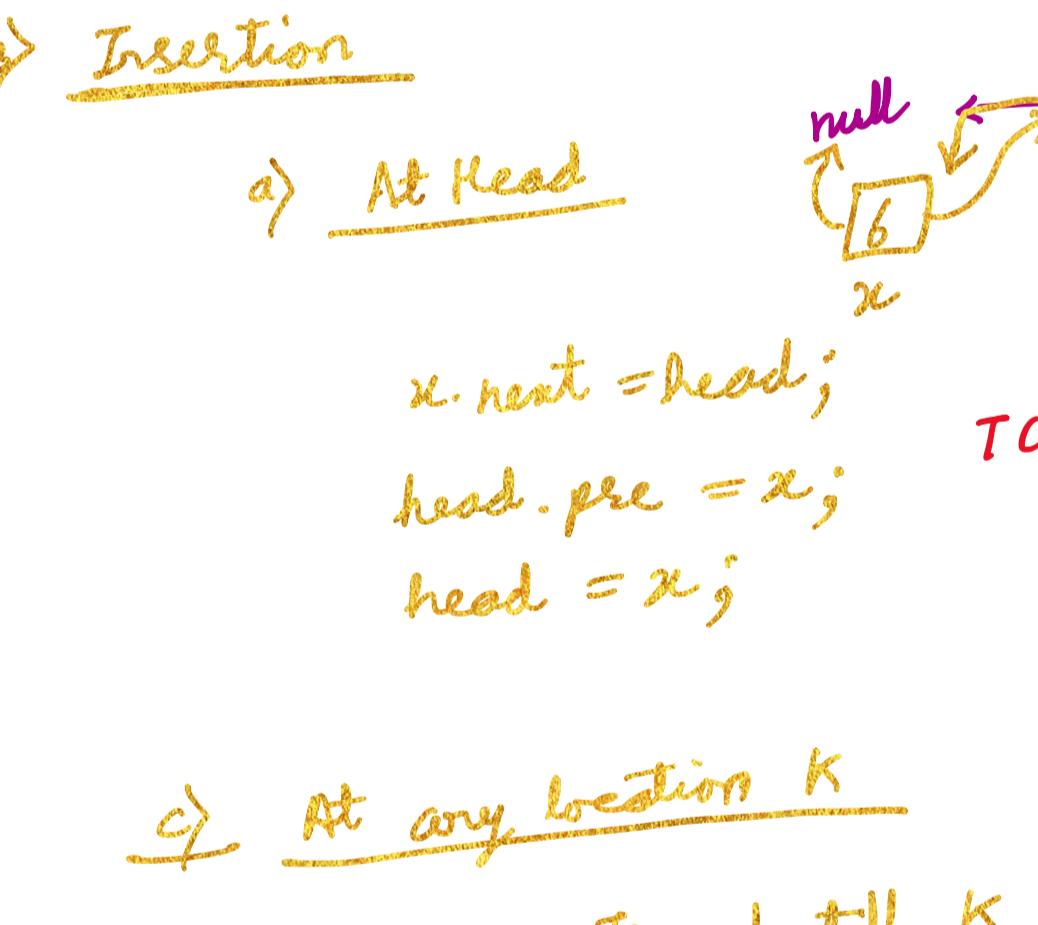


a) $h2 = \text{mid}.next;$
 $\text{mid}.next = \text{null};$

while($y \neq \text{null}$) {
 xn = x.next;
 x.next = y;

b) reverse(h2);

yn = y.next;
 y.next = xn;



$$TC = O(N)$$

$$SC = O(1)$$

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow \text{null}$

Q) Implement stacks using LL?



push(x) \rightarrow Insert at Head

node x = new node(x);
 node x.next = head;
 head = node x;



$$TC = O(1)$$

pop() \rightarrow Remove from Head

if(!isEmpty()) {
 head = head.next;

3 else if(head == null)
 tail = null;

3 else if(tail == null)
 tail = head;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head.next == null)
 tail = head;

3 else if(head == null)
 tail = null;

3 else if(head.next == null)
 tail = head;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail = head.next;

3 else if(head == tail)
 tail = head.next;

3 else if(head.next == tail)
 tail =