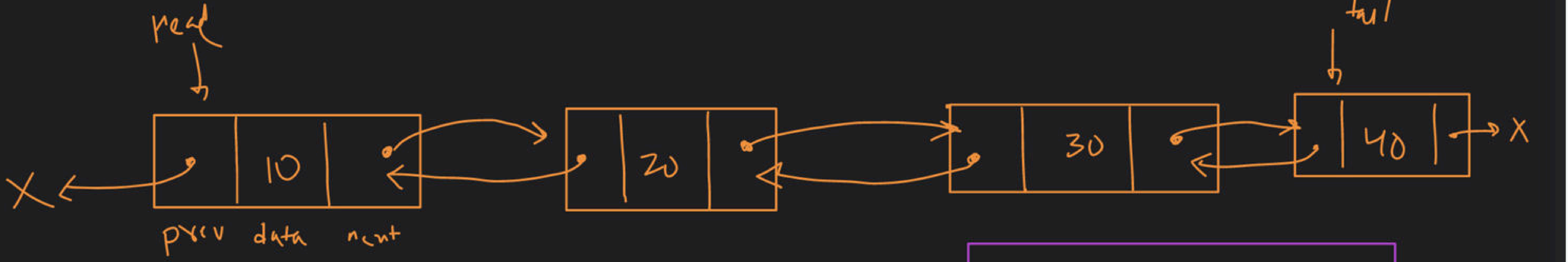
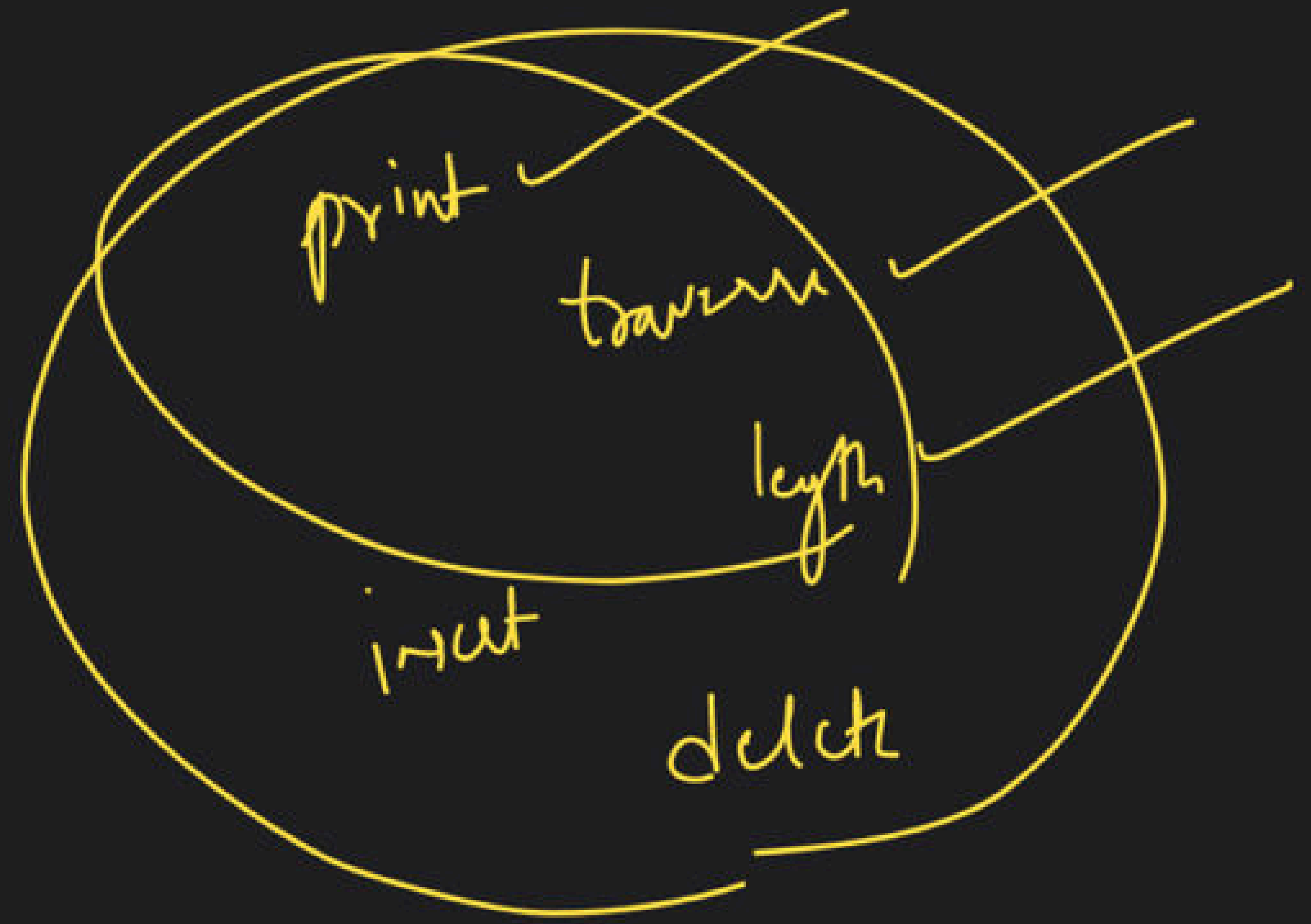
LL Class - 2

Special class

Love Babbar • Apr 9, 2023

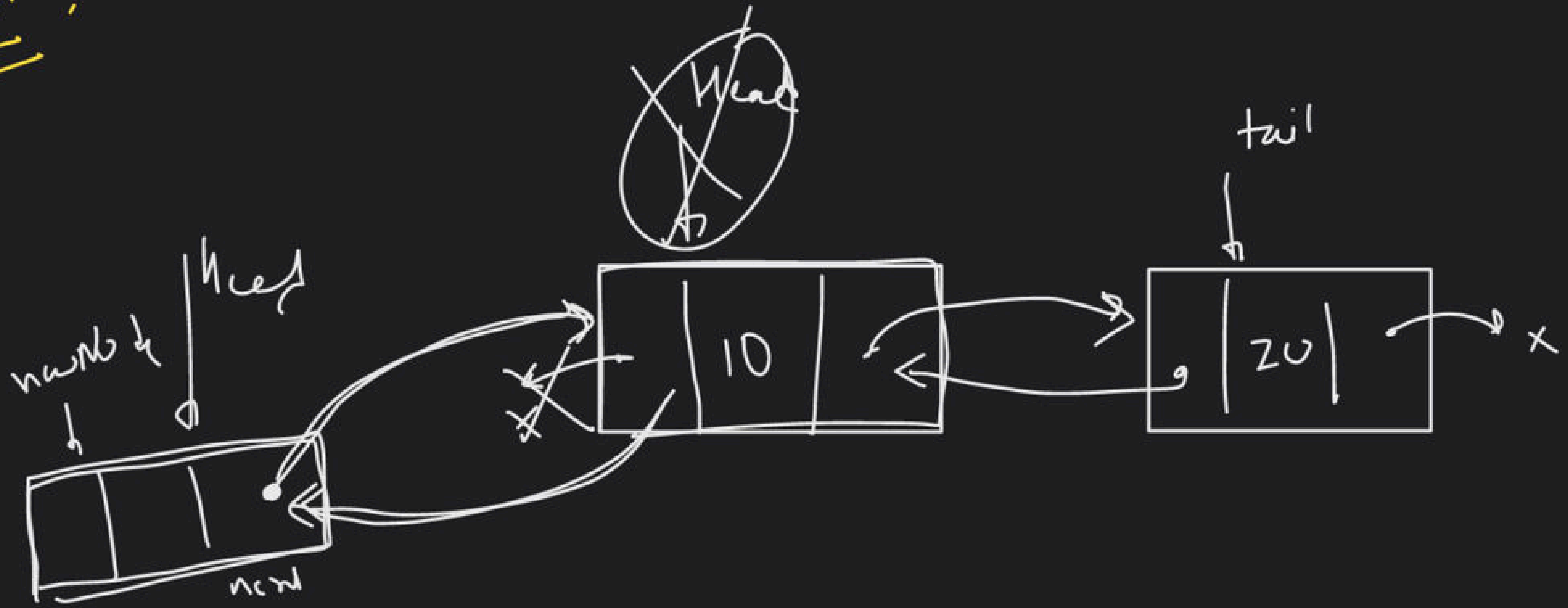
Doubly Linked List





```
Class Node
{
        int data; ✓
        Node* prev; ✓
        Node* next; ✓
}
```

insert AtHead :-



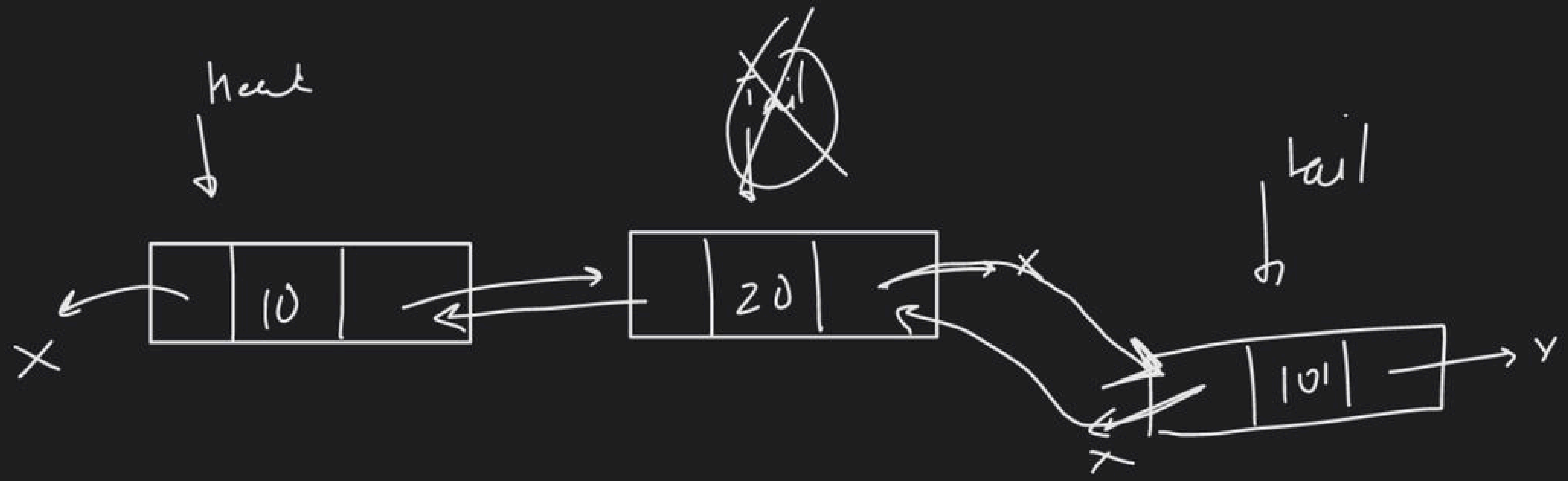
Step → Ⓐ create a NewNode

Ⓑ newNode → next = head

Ⓒ head → prev = newNode

Ⓓ head = newNode

insert ATTail



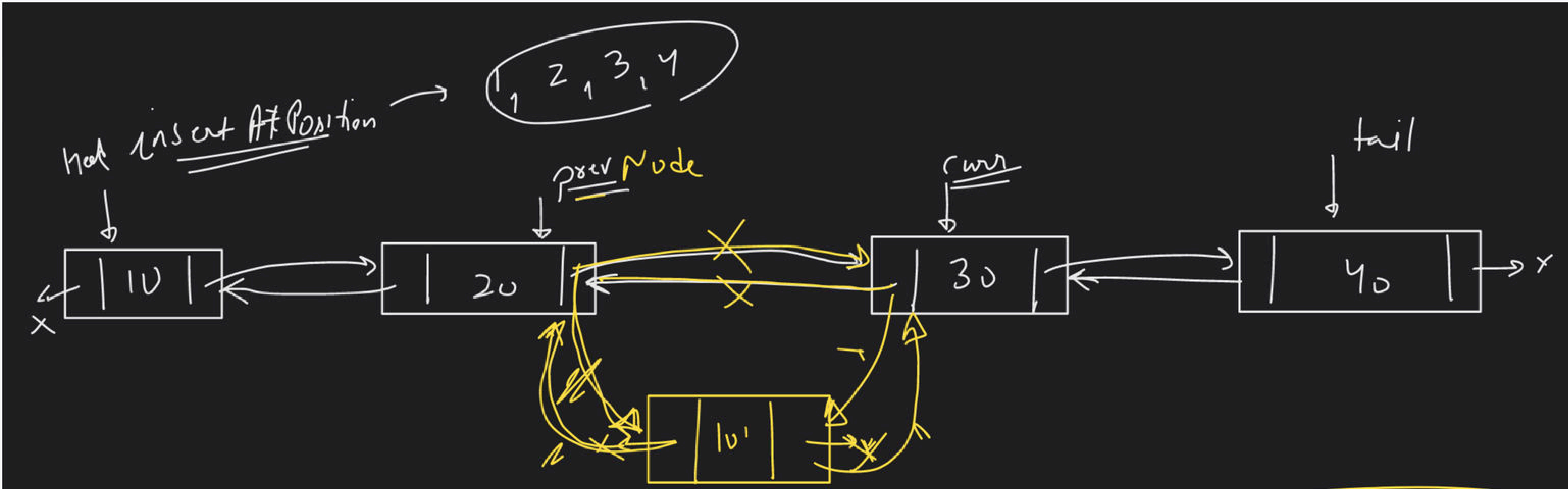
Steps :-

(A) Create a node

(B) tail → next = newNode

(C) newNode → prev = tail

(d) tail = newNode



step

① find prev & curr

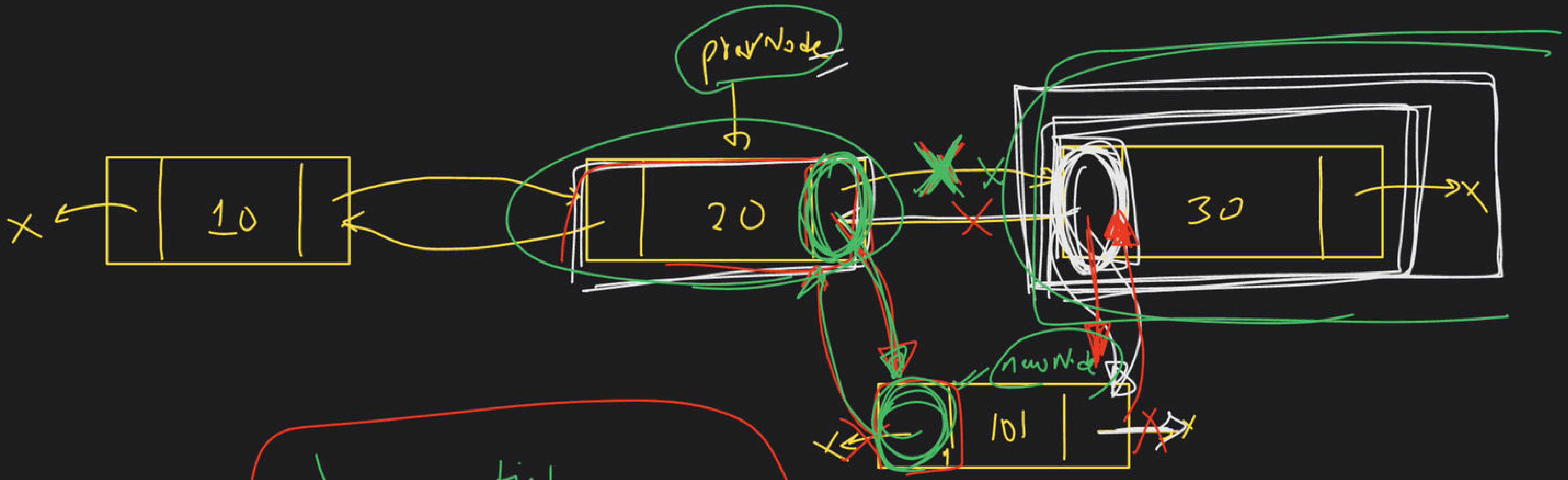
② creat a node

③ prevNode → next = newNode

④ newNode → prev = prevNode

⑤ curr → prev = newNode

⑥ newNode → next = curr



A find prevNode

B create New Node

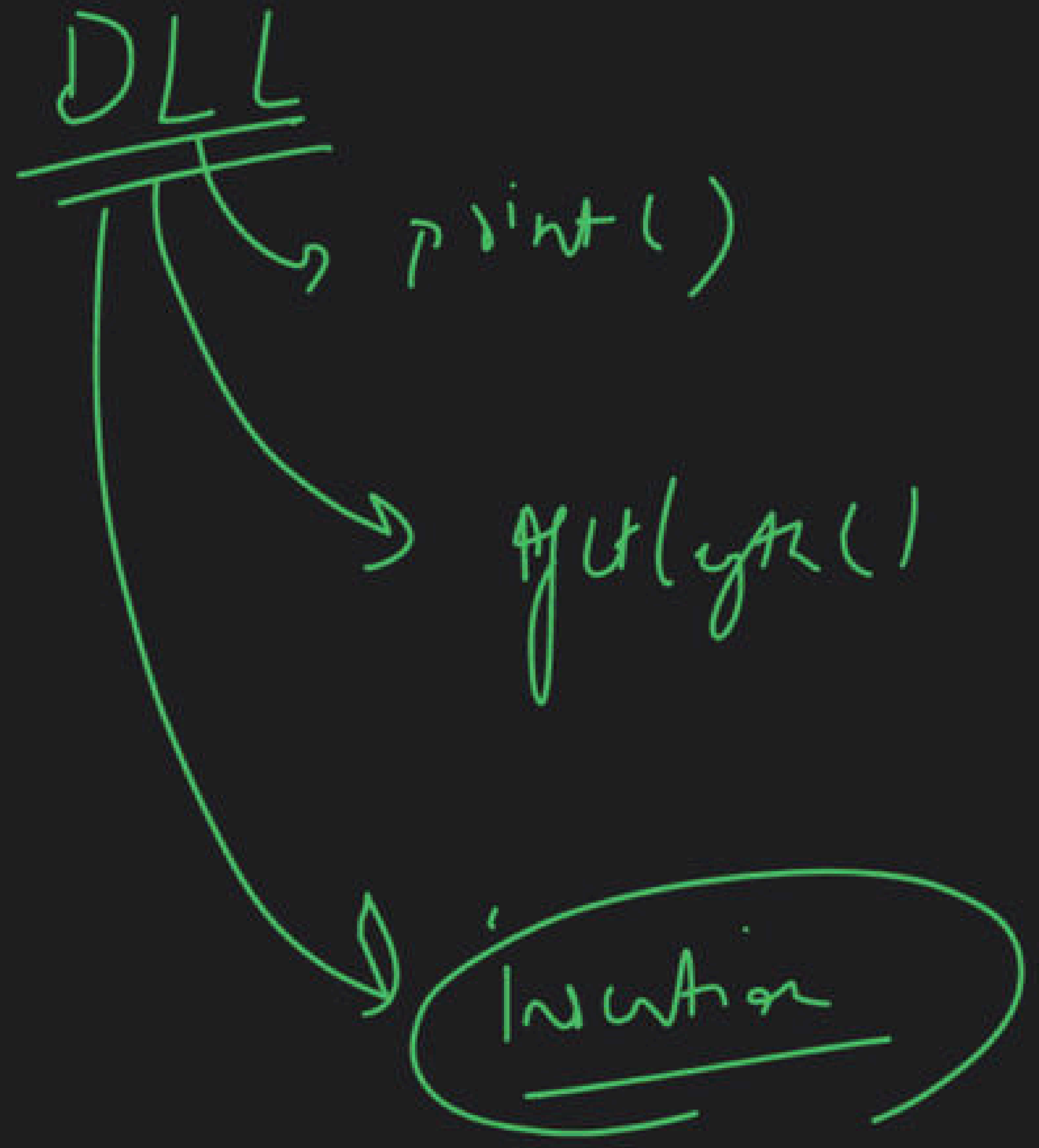
1 prevNode → next → prev = newNode

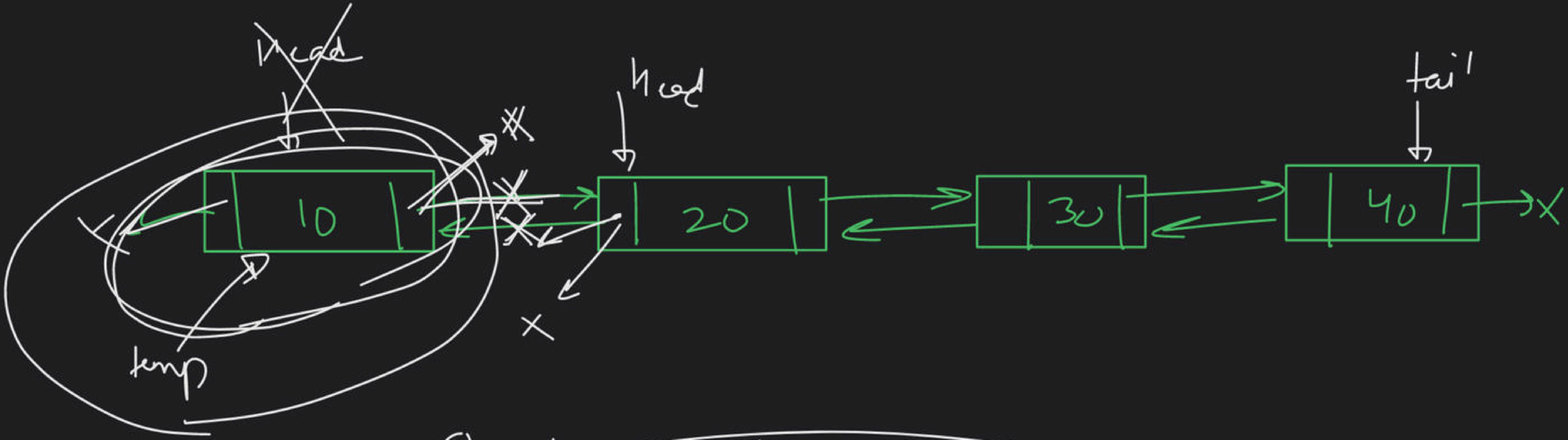
C newNode → next = prevNode → next

C prevNode → next = newNode

1 newNode → prev = prevNode

Deletion





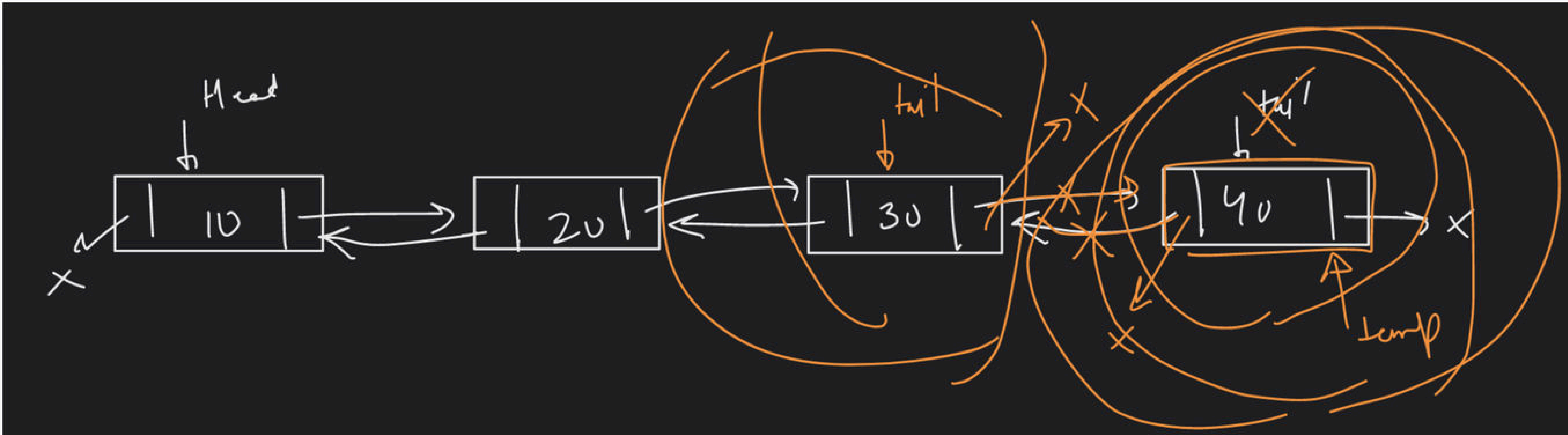
Ⓐ temp = head

Ⓑ Head = head → next

ⓒ Head → prev = NULL

ⓓ temp → next = NULL

ⓔ delete temp



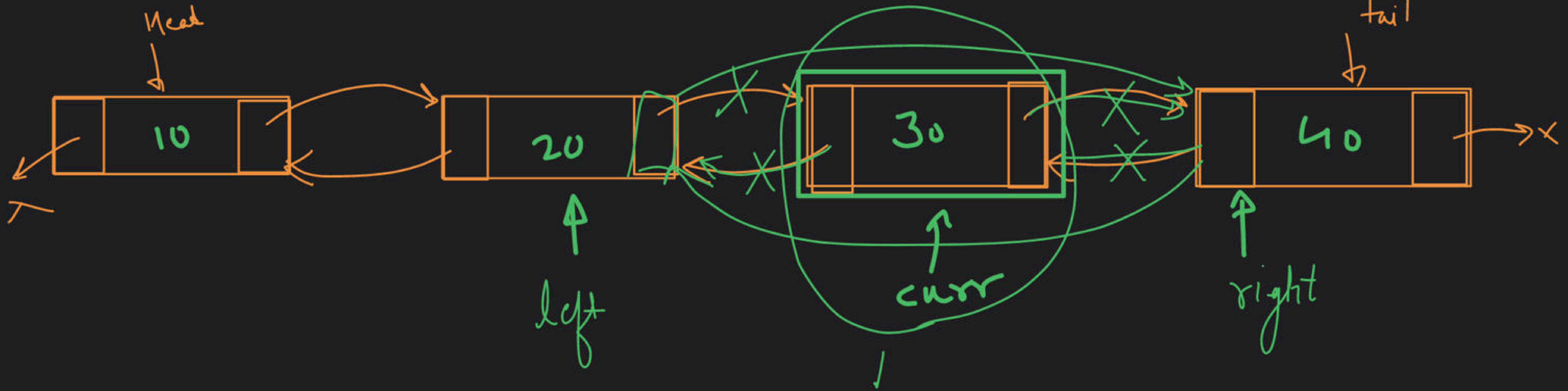
(A) temp = tail

(B) tail = tail → prev

(C) temp → prev = NULL

(D) tail → next = NULL

(E) delete temp



(A) find left, curr, right

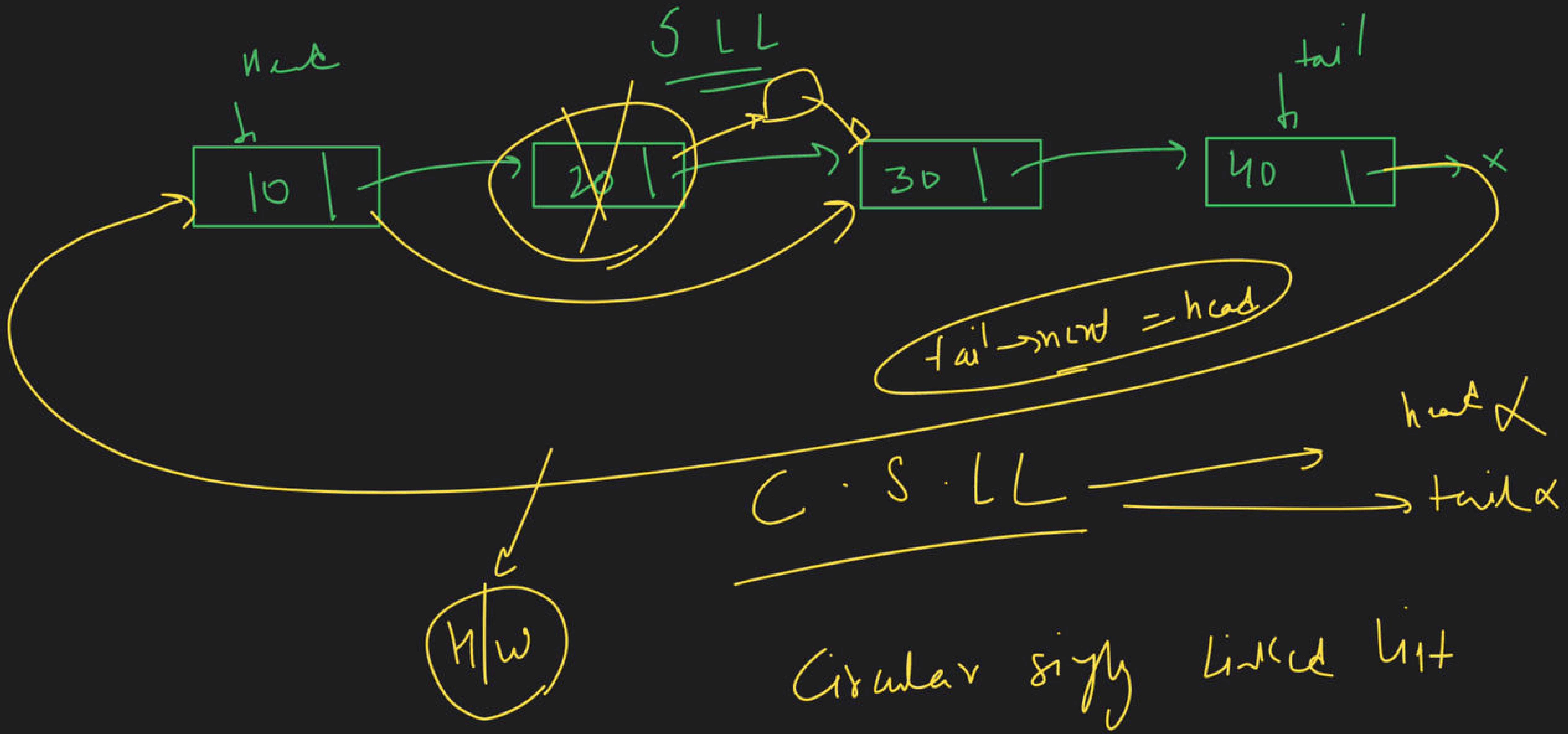
(B) left -> next = right;

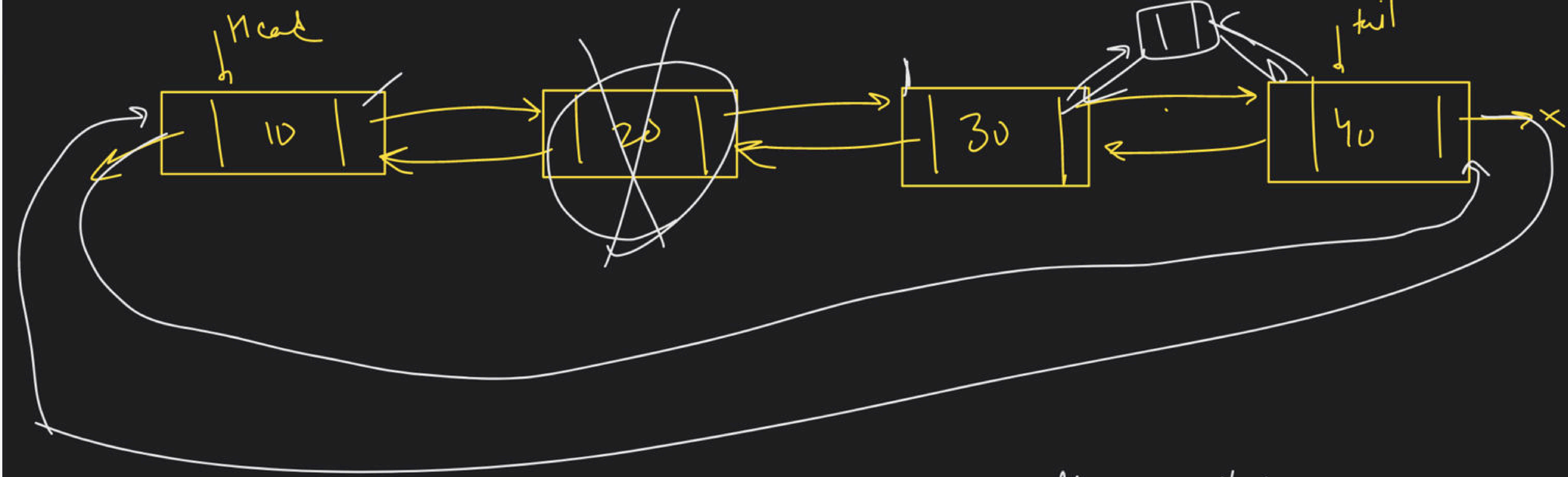
(C) right -> prev = left

(D) curr -> prev = NULL

(E) curr -> next = NULL

(F) delete curr


SLL
head
tail
10
20
30
40
x
tail->next = head
C.S.LL
head ✗
tail ✗
H/W
Circular sigly linked list



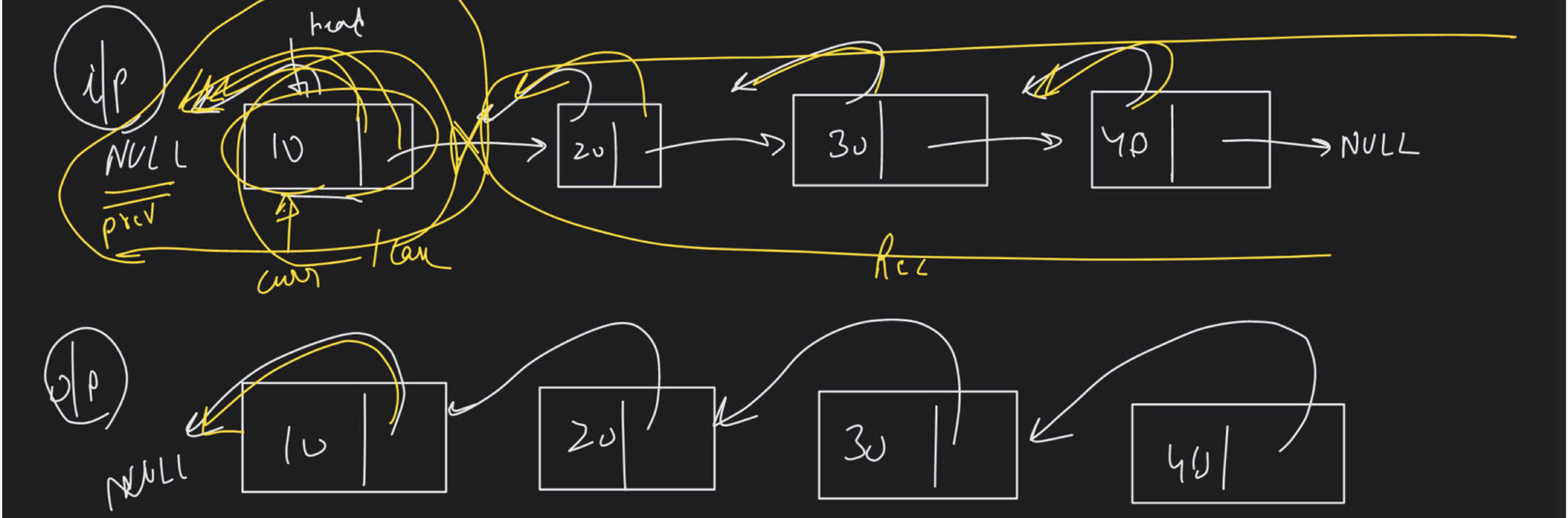
Circular doubly linked list

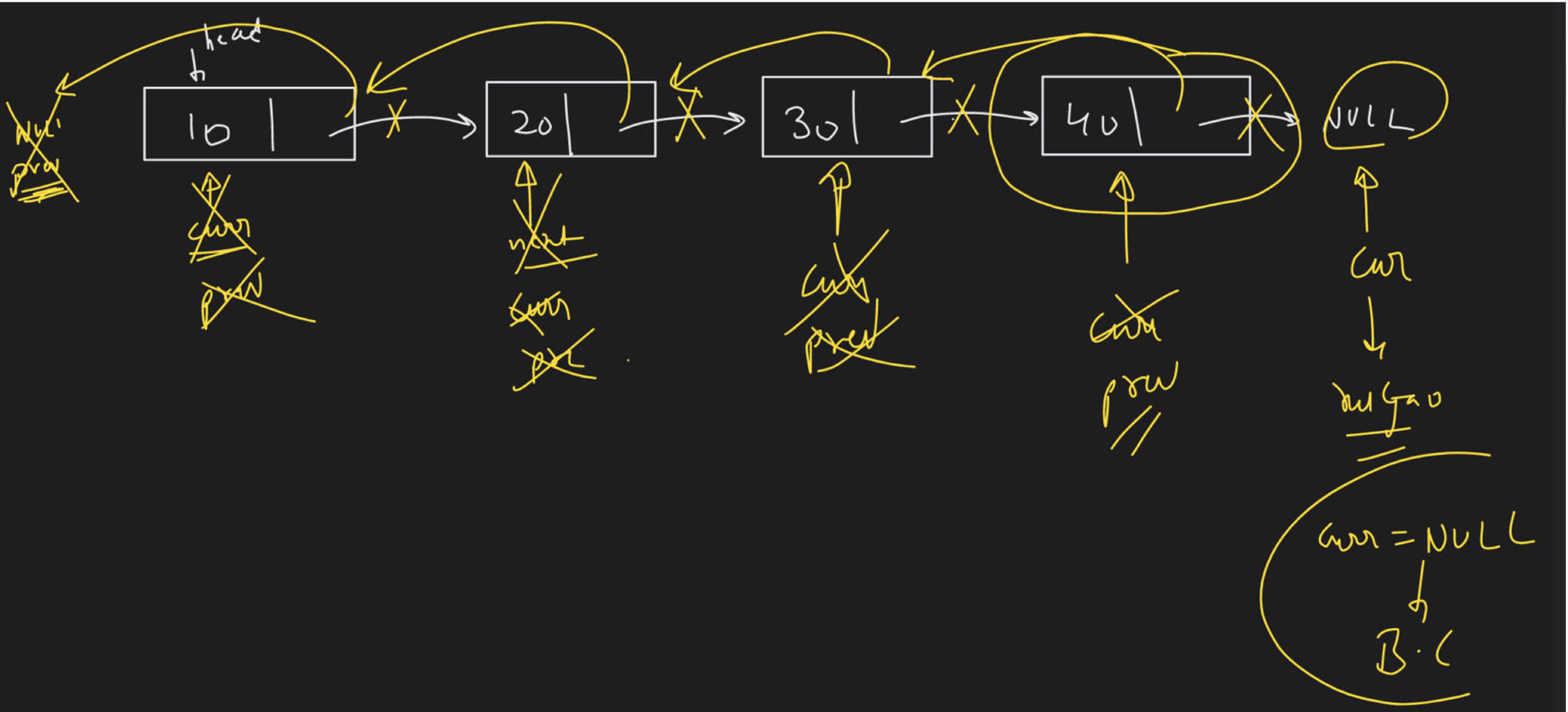
(1) tail → next = head

(2) head → prev = tail

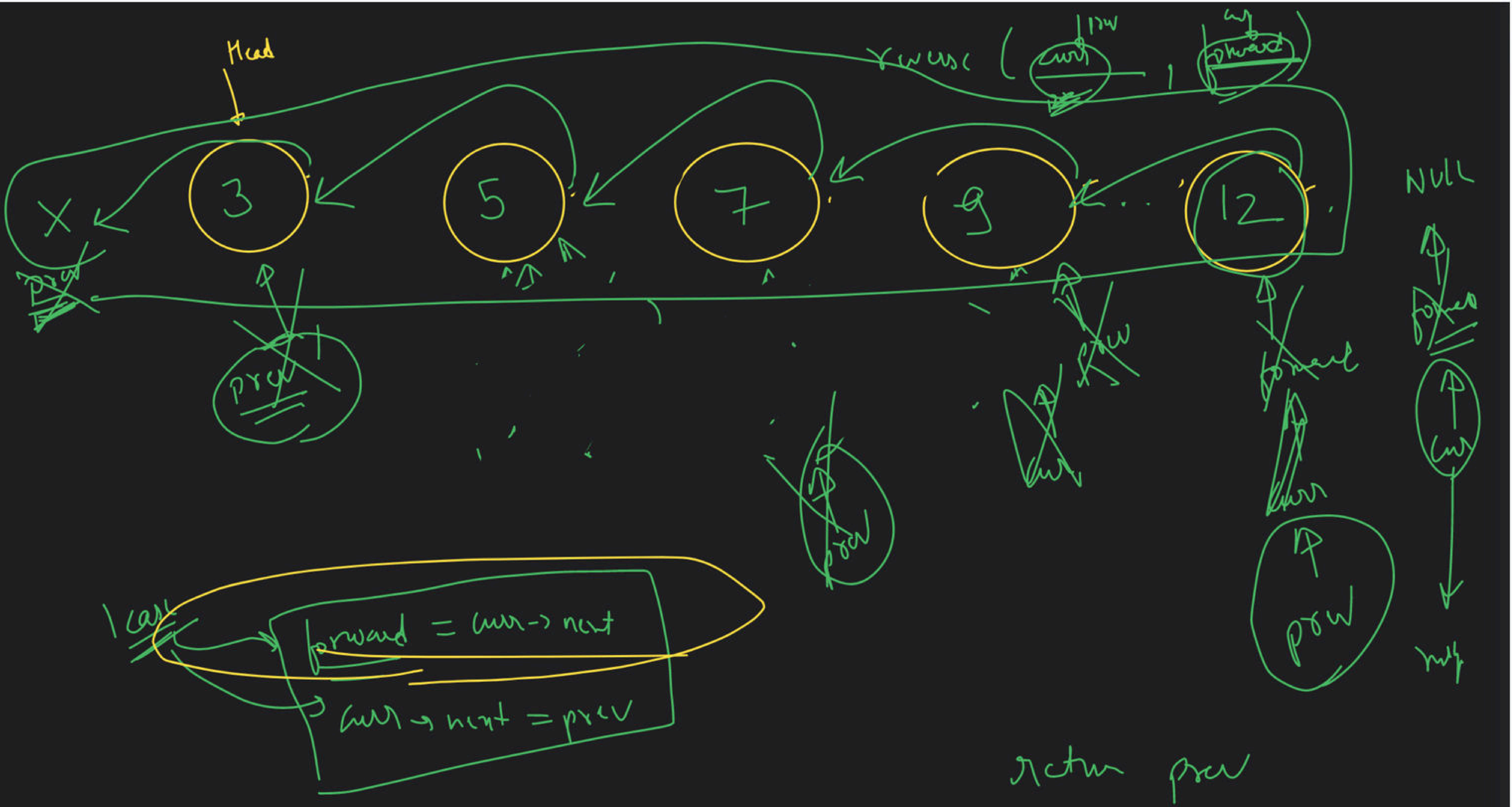
Questions →

Reverse a Linked List





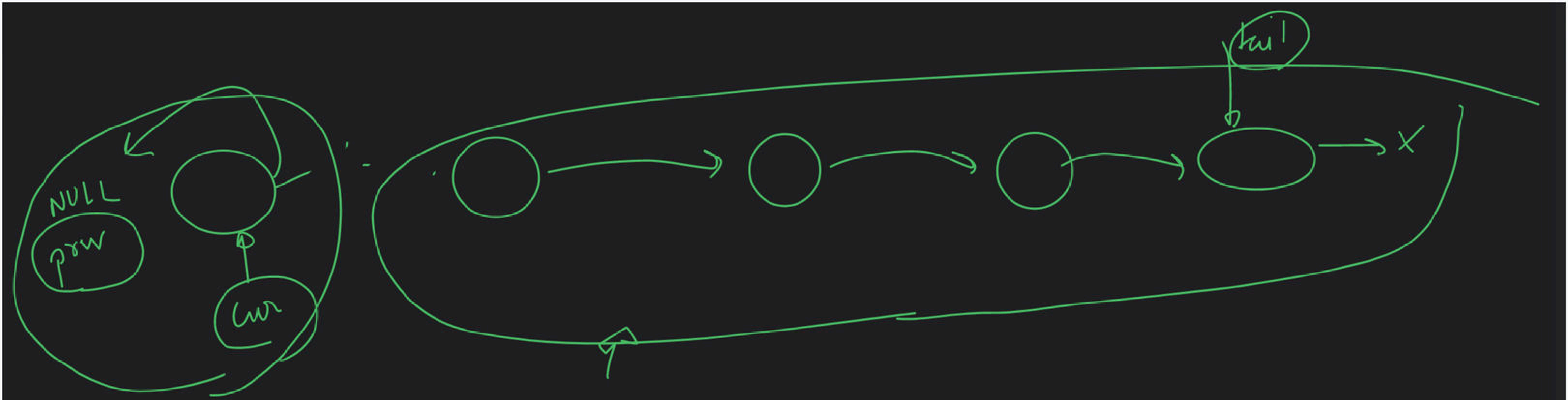
head
NULL
prev
10
20
30
40
NULL
curr
prev
next
curr
prev
curr
prev
curr
prev
curr
next
curr = NULL
B.C



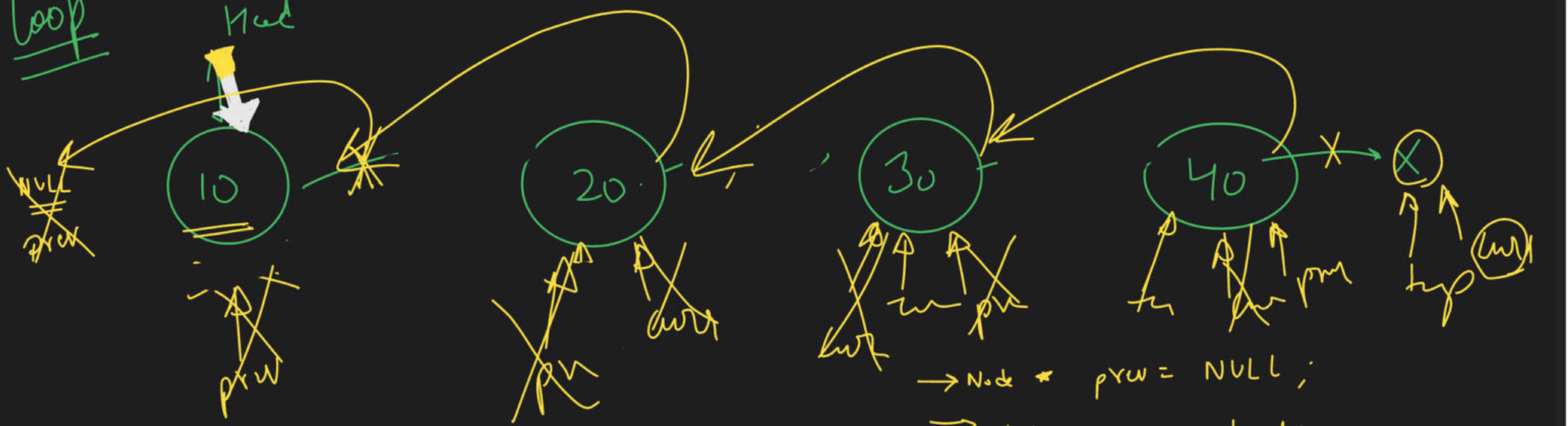
1 case

```
forward = curr -> next
curr -> next = prev
```

return prev

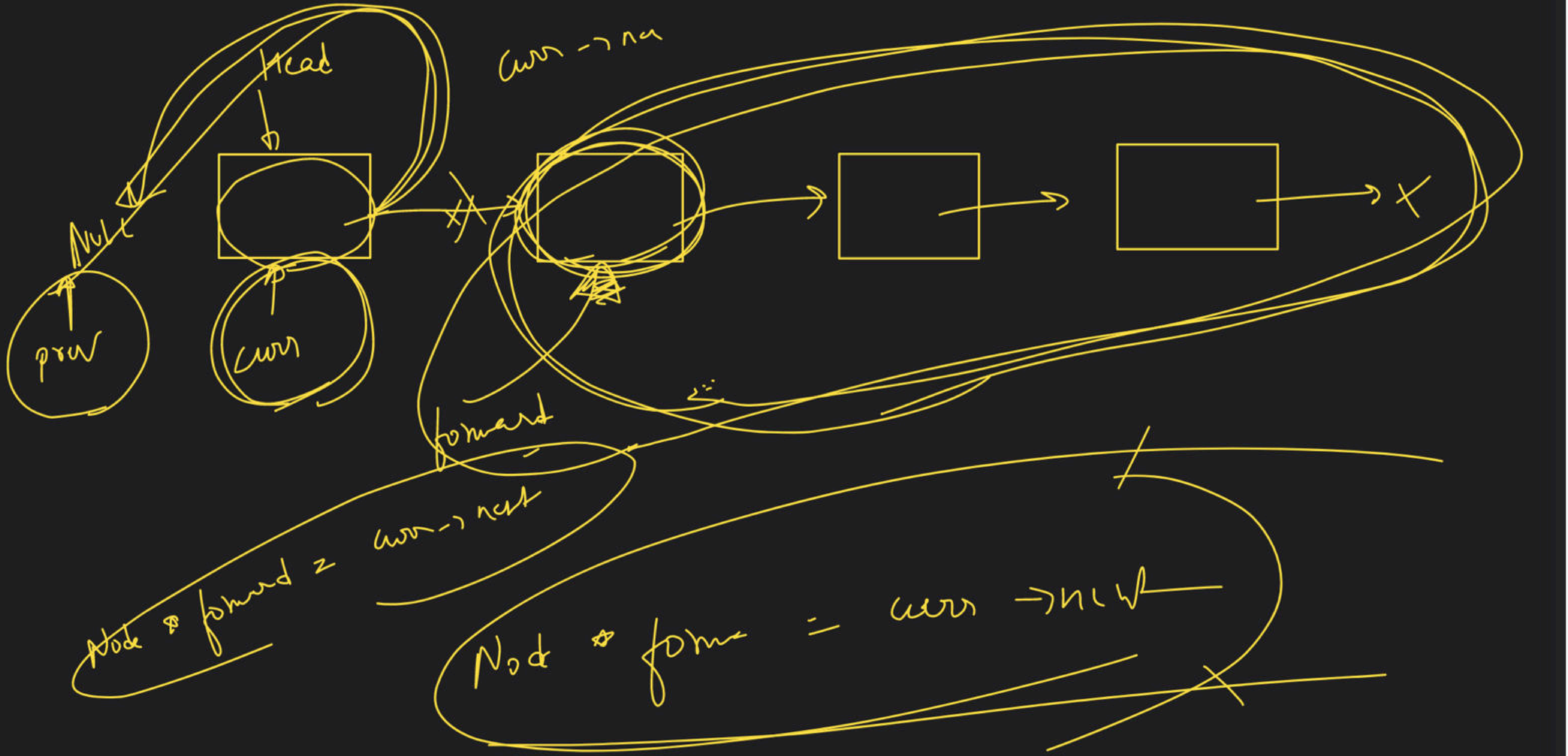

NULL
prev
curr
tail
X

Loop



```
→ Node * prev = NULL;
→ Node* curr = head;

while(curr != NULL)
{
    ✓→ Node* temp = curr->next;
    ✓→ curr -> next = prev
    ✓→ prev = curr;
    ✓→ curr = temp
}
```



Node * forward = curr -> next

Node * forw = curr -> next