



Linked List

↳ Deletion

↳ Insertion



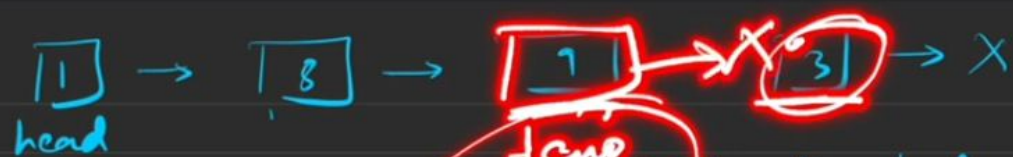
- head ✓
- position
- value
- last







6:59



temp

delete the tail of the LL

VoLTE 97





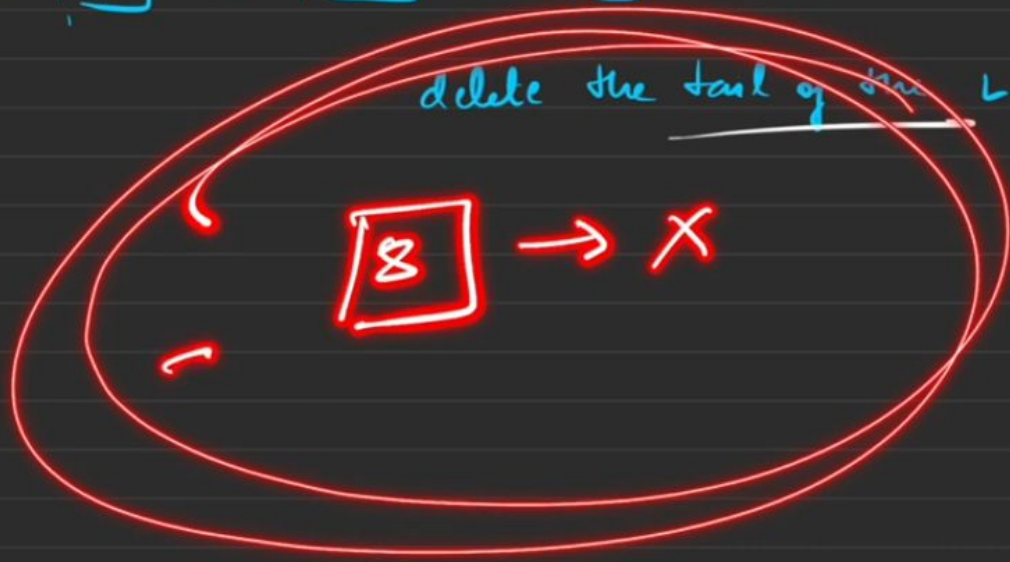
delete the tail of the LL

2 elements





delete the tail of LL




 $\boxed{1} \rightarrow \boxed{8} \rightarrow \boxed{7} \rightarrow \boxed{3} \rightarrow X$
head

delete the tail of LL





delete the tail of the LL

Node * deleteTail (Node * head)
{

LL 1 element \rightarrow delete 1 ele

if (head == null ||
 head->next == null)





Node * deleteTail (Node * head)

{

if (head == null || head->next == null) return null;

LL 1 element \rightarrow delete 1 ele





Node * deleteTail (Node * head)
 {
 temp ↑
 LL 1 element → delete 1 ele

if (head == null ||
 head->next == null) return null;

temp->next->next





head
temp

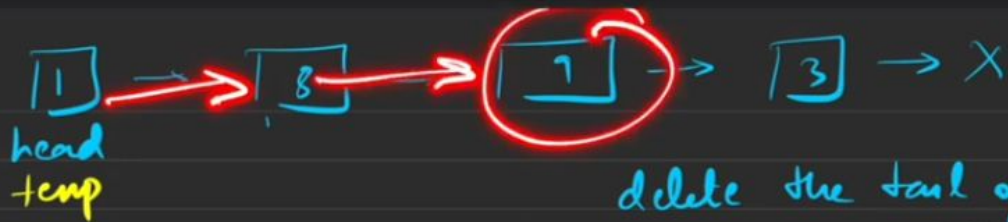
delete the tail of the LL

Node * deleteTail(Node * head)
{

LL 1 element → delete 1 ele

```
if (head == null ||  
    head->next == null) return null;  
Node * temp = head;  
while (temp->next->next != null)  
{  
    temp = temp->next;  
}
```





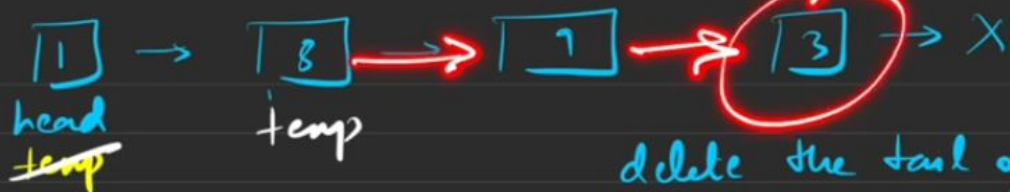
delete the tail of the LL

Node * deleteTail(Node * head)
{

LL 1 element \rightarrow delete 1 ele

```
if (head == null ||  
    head->next == null) return null;  
Node * temp = head;  
while (temp->next->next != null)  
{  
    temp = temp->next;  
}
```





Node * deleteTail(Node * head)
{

LL 1 element \rightarrow delete 1 ele

if (head == null || return null;

head->next == null)

Node * temp = head;

while (temp->next->next != null)

{

temp = temp->next;

}





Node * deleteTail(Node * head)
{

LL 1 element \rightarrow delete 1 ele

if (head == null || return null;

head->next == null)

Node * temp = head;

while (temp->next->next != null)

{

temp = temp->next;

}



10:00

SnapSave.io-L2. Deletion and Insertion in LL_ 8 Problems

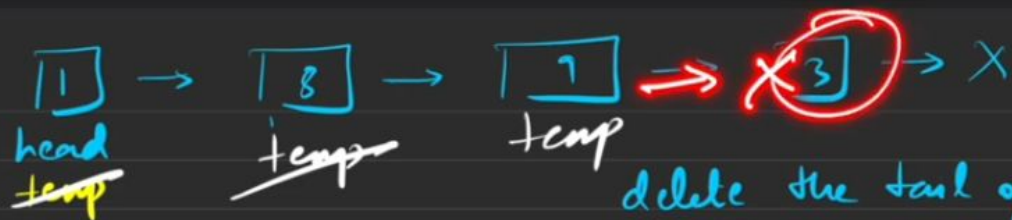
~~head~~
~~temp~~
 1X
 Node *
 deleteTail(Node *head)
 {
 LL 1 element → delete 1 ele

if (head == null ||
 head->next == null) return null;
 Node * temp = head;
 while (temp->next->next != null)
 {

temp = temp->next;
 }

11:31

56:29



delete the tail of the LL

Node * deleteTail(Node * head)
{

LL l = head -> delete l;

```
if (head == null ||  
    head->next == null) return null;  
Node * temp = head;  
while (temp->next->next != null)  
{  
    temp = temp->next;  
}
```



Node * deleteTail (Node * head)
{

// 1 element → delete 1 el

if (head == null || return null;

head->next == null)

Node * temp = head;

while (temp->next->next != null)

{

temp = temp->next;

}

free (temp->next)

temp->next = null ptr;





Node * deleteTail (Node * head)
{

LL 1 element \rightarrow delete 1 ele

if (head == null || head->next == null) return null;

Node * temp = head;

while (temp->next->next != null)

{

temp = temp->next;

}


```
LL.cpp x
LL.cpp > removeTail(Node *)
52 }
53 cout << endl;
54 }
55 Node* removesHead(Node* head) {
56     if(head == NULL) return head;
57     Node* temp = head;
58     head = head->next;
59     delete temp;
60     return head;
61 }
62 Node* removeTail(Node* head) {
63     if(head == NULL || head->next == NULL) return NULL;
64
65     Node* temp = head;
66     while(temp->next->next != NULL) {
67         temp = temp->next;
68     }
69     delete temp->next;
70     temp->next = nullptr;
71
72     return head;
73 }
74
75 int main() {
76     vector<int> arr = {12, 5, 8, 7};
77     Node* head = convertArr2LL(arr);
78     head = removeTail(head);
79     print(head);
80 }
```

input.txt x

```
1 13
```

output.txt x

```
output.txt
1 12 5 8
2
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Executing task: g++ -std=c++17 -c LL.cpp -o LL.o /Users/striver/Desktop/leetcode/LL.cpp 55 /LL < input.txt > output.txt

