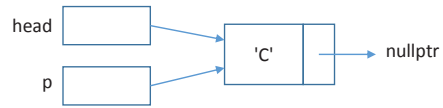


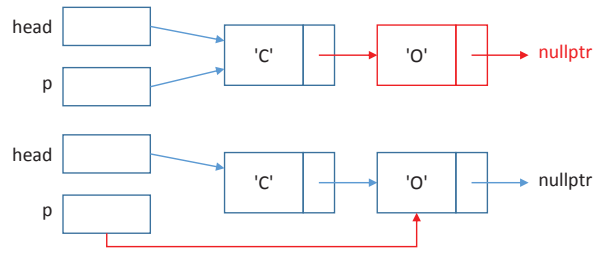
Topic 9: Linked List --- Demonstrating what ll_create does

Suppose we pass "COMP" to ll_create

Step 1



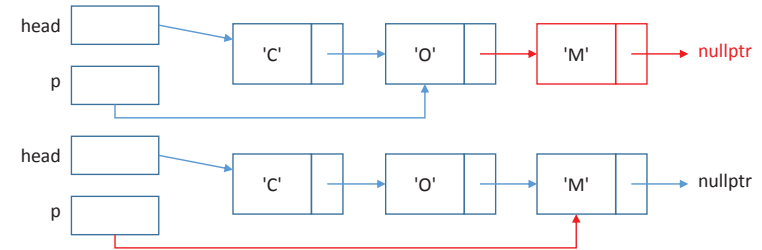
Step 2



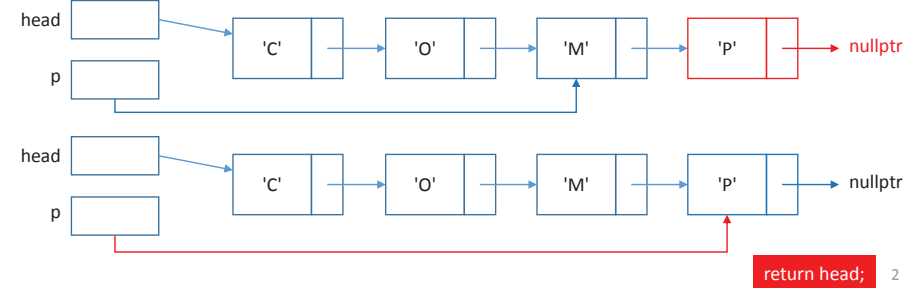
1

Topic 9: Linked List --- Demonstrating what ll_create does

Step 3



Step 4



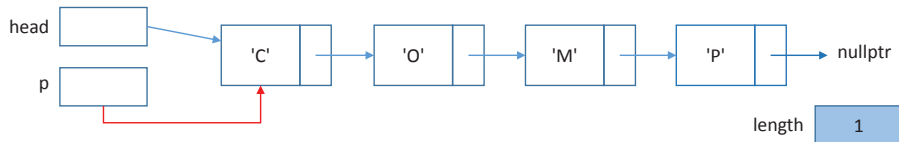
2

Topic 9: Linked List --- Demonstrating what ll_length does

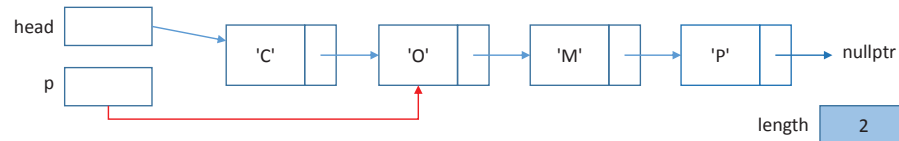
Suppose we pass head pointer to ll_length

length 0

Step 1



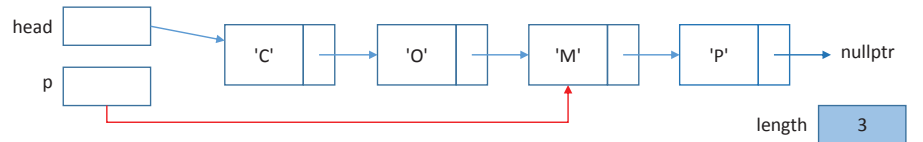
Step 2



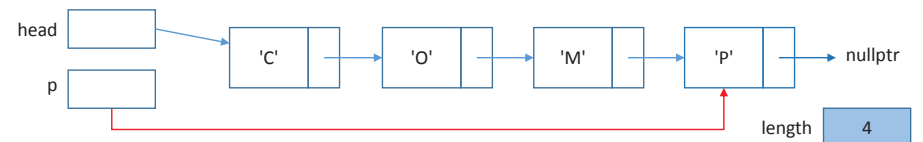
3

Topic 9: Linked List --- Demonstrating what ll_length does

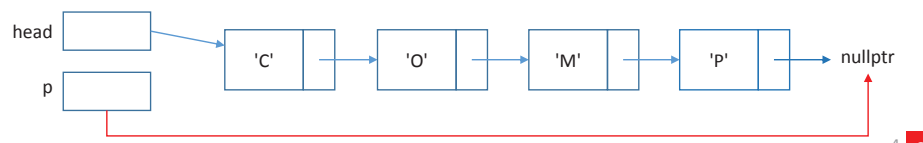
Step 3



Step 4



Step 5



4

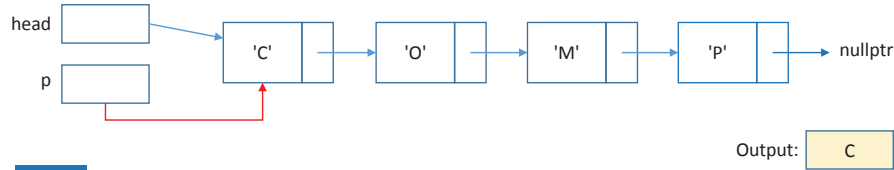
Stop!

Topic 9: Linked List --- Demonstrating what ll_print does

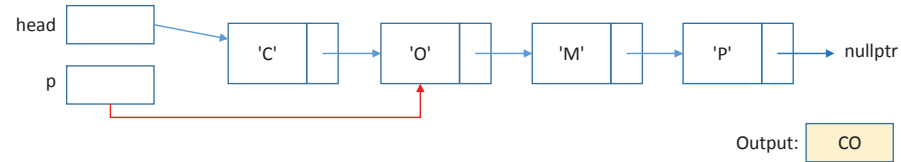
Suppose we pass head pointer to ll_print

Output:

Step 1



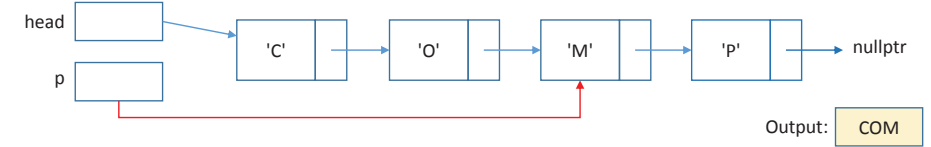
Step 2



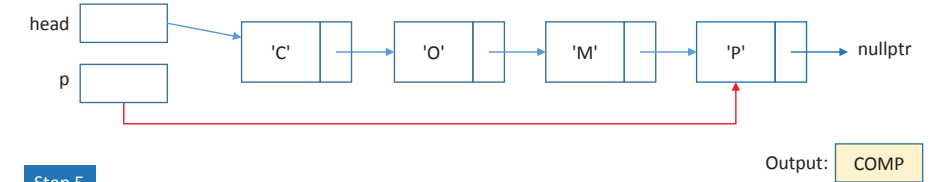
5

Topic 9: Linked List --- Demonstrating what ll_print does

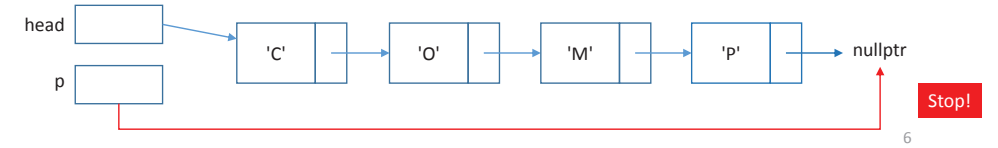
Step 3



Step 4



Step 5



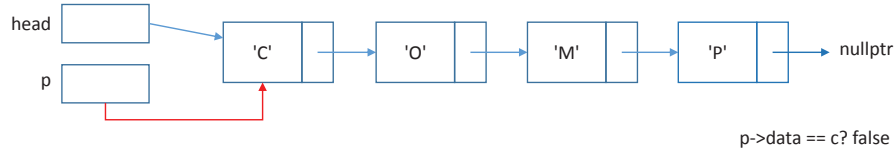
6

Topic 9: Linked List --- Demonstrating what ll_search does

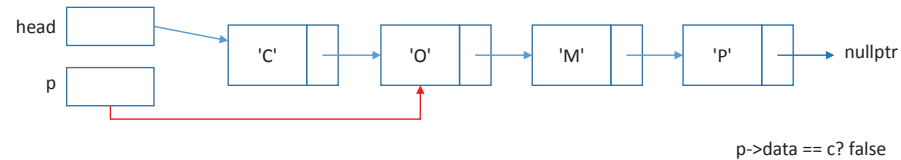
Suppose we pass head pointer and character 'M' to ll_search

c 'M'

Step 1



Step 2

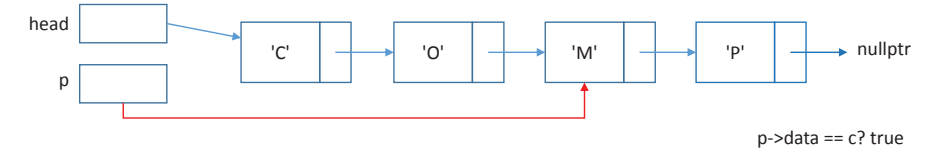


7

Topic 9: Linked List --- Demonstrating what ll_search does

c 'M'

Step 3



Return pointer p!

8

Topic 9: Linked List --- Demonstrating what ll_insert does

Suppose we pass head pointer, character 'A', and 0 to ll_insert

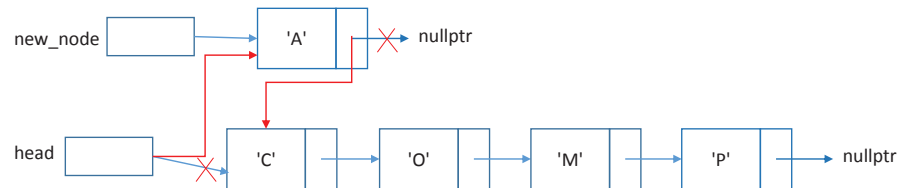
c 'A' n 0

Step 1



Step 2

n == 0? true



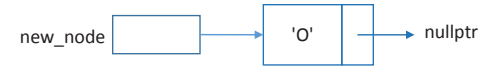
9

Topic 9: Linked List --- Demonstrating what ll_insert does

Suppose we pass head pointer, character 'O', and 2 to ll_insert

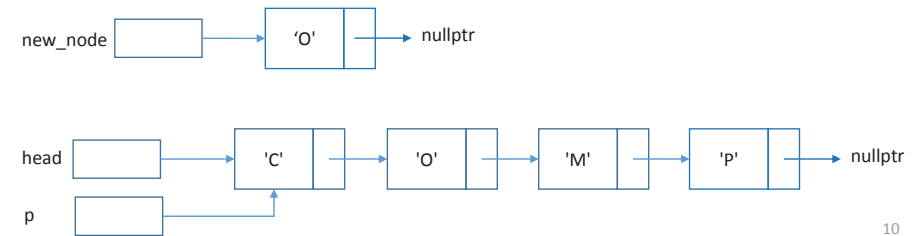
c 'O' n 2

Step 1



Step 2

n == 0 || head == nullptr? false



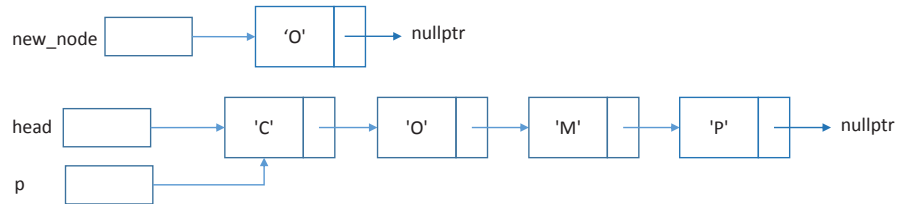
10

Topic 9: Linked List --- Demonstrating what ll_insert does

Step 3

c 'O' n 2 position 0

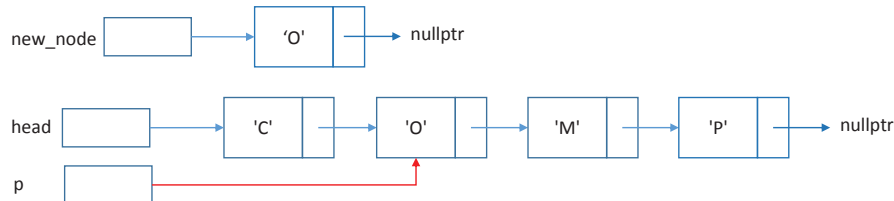
position < n - 1 && p->next != nullptr? true → Execute ;



Step 4

c 'O' n 2 position 1

p = p->next, ++position



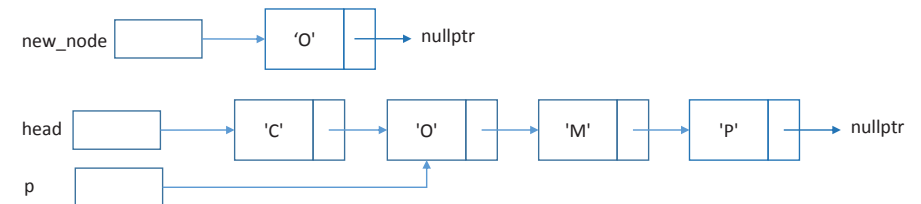
11

Topic 9: Linked List --- Demonstrating what ll_insert does

Step 4

c 'O' n 2 position 1

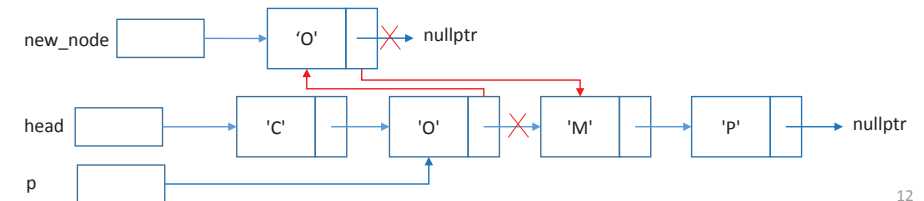
position < n - 1 && p->next != nullptr? false → Leave for loop



Step 5

c 'O' n 2 position 1

new_cnode->next = p->next;
p->next = new_cnode;



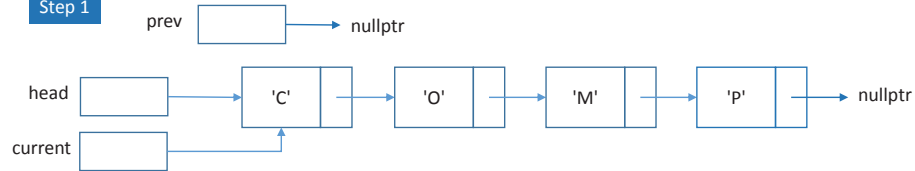
12

Topic 9: Linked List --- Demonstrating what ll_delete does

Suppose we pass head pointer, character 'M' to ll_delete

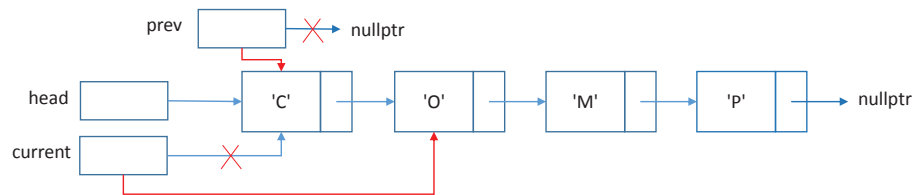
c 'M'

Step 1



Step 2

current != nullptr && current->data != c? true



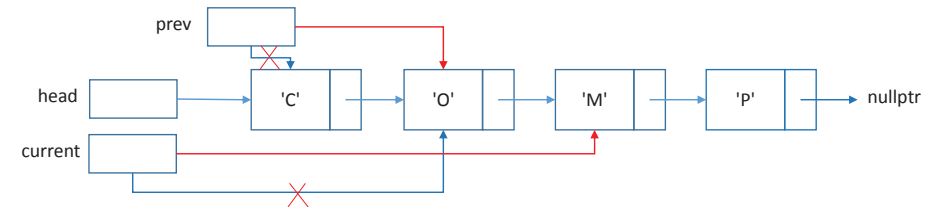
13

Topic 9: Linked List --- Demonstrating what ll_delete does

c 'M'

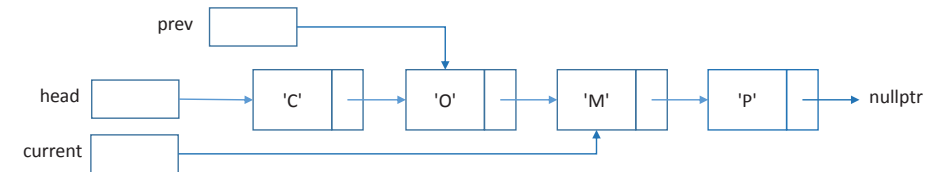
Step 3

current != nullptr && current->data != c? true



Step 4

current != nullptr && current->data != c? false \rightarrow Leave while loop!



14

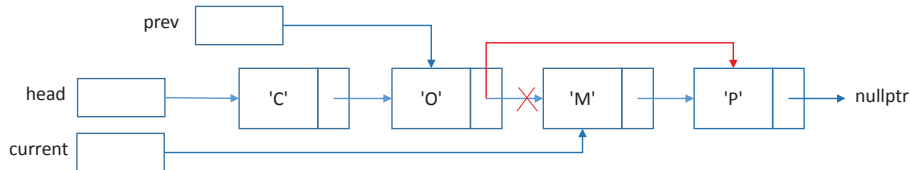
Topic 9: Linked List --- Demonstrating what ll_delete does

c 'M'

Step 5

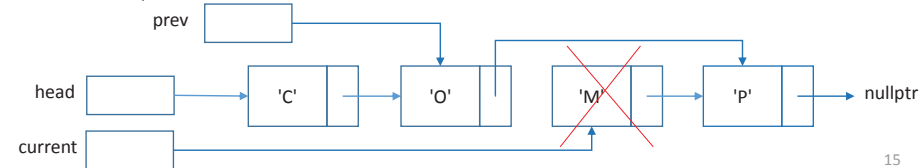
current != nullptr? true

current == head? false \rightarrow prev->next = current->next;



Step 6

delete current;

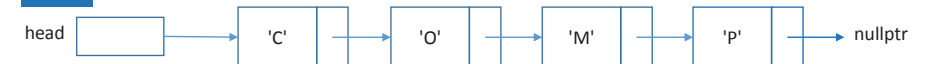


15

Topic 9: Linked List --- Demonstrating what ll_delete_all does

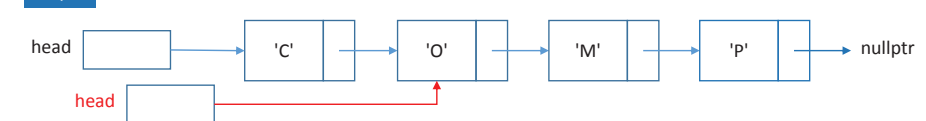
Suppose we pass head pointer to ll_delete

Step 1



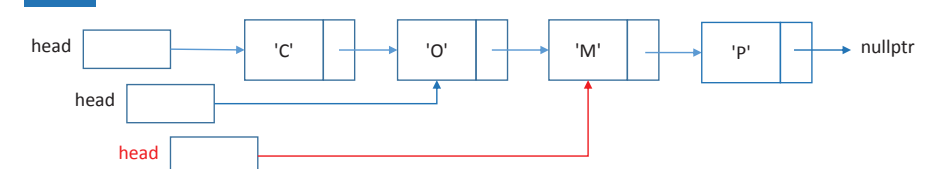
head == nullptr? false \rightarrow ll_delete_all(head->next);

Step 2



head == nullptr? false \rightarrow ll_delete_all(head->next);

Step 3

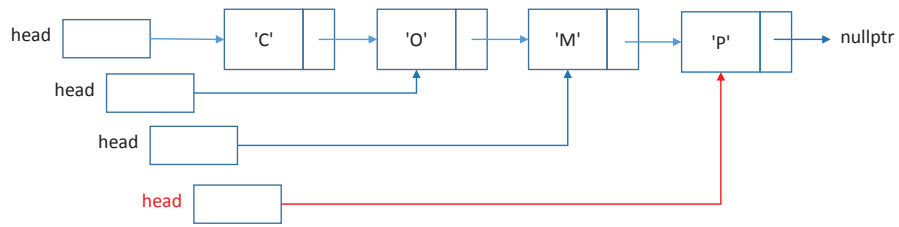


head == nullptr? false \rightarrow ll_delete_all(head->next);

16

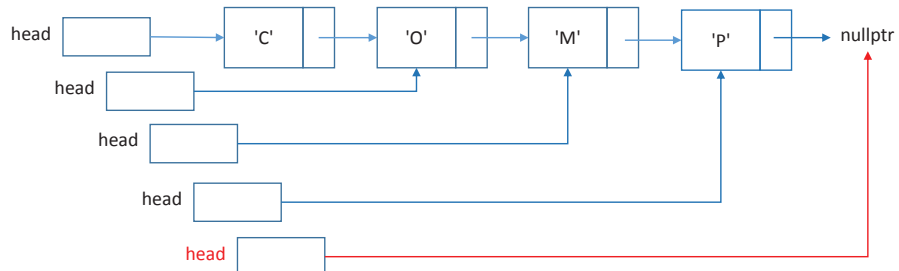
Topic 9: Linked List --- Demonstrating what ll_delete_all does

Step 4



`head == nullptr? false → ll_delete_all(head->next);`

Step 5

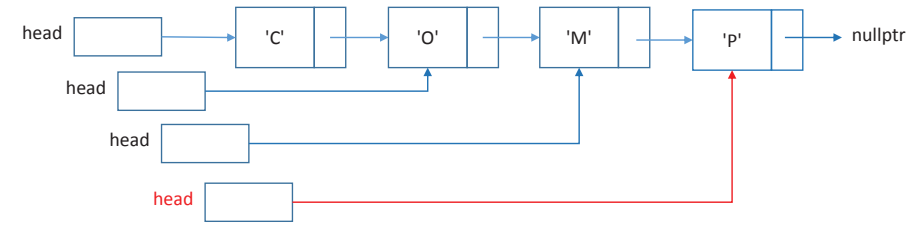


`head == nullptr? true → return;`

17

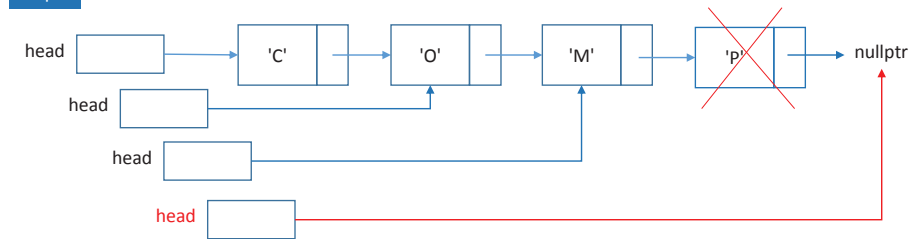
Topic 9: Linked List --- Demonstrating what ll_delete_all does

Step 6



`delete head; head = nullptr;`

Step 7

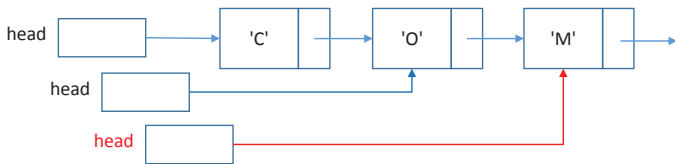


`delete head; head = nullptr; leave ll_delete_all`

18

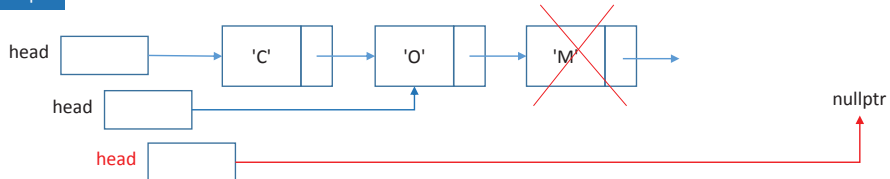
Topic 9: Linked List --- Demonstrating what ll_delete_all does

Step 8



`delete head; head = nullptr; leave ll_delete_all`

Step 9

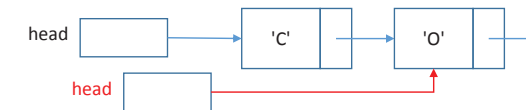


`delete head; head = nullptr; leave ll_delete_all`

19

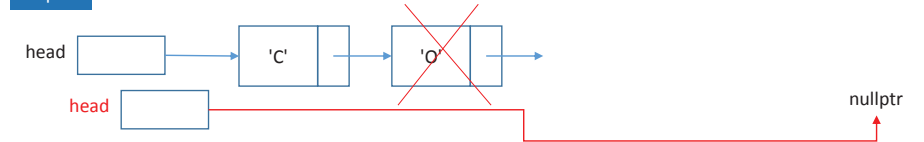
Topic 9: Linked List --- Demonstrating what ll_delete_all does

Step 10



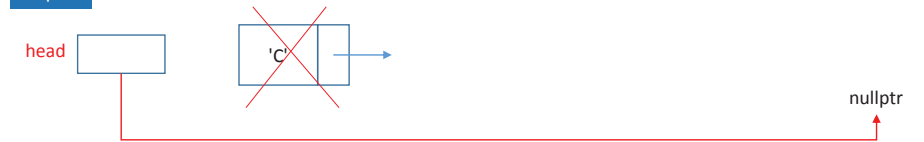
`delete head; head = nullptr; leave ll_delete_all`

Step 11



`delete head; head = nullptr; leave ll_delete_all`

Step 12



`delete head; head = nullptr; leave ll_delete_all`

20

Step 13



head == null? true → return;