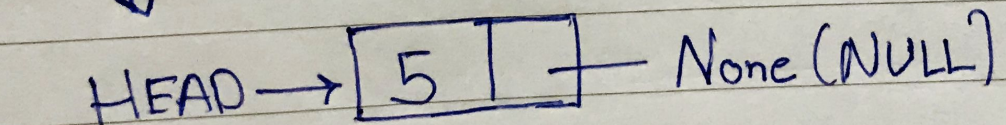
Push Operation on Linked List :-

```
def push(head, data):
    head = node(data, head)
    return head
```

```
def node(self, data=None, Nent=None):
    self.data = data
    self.nent = Nent
```

1. start = None
1. start = push(start, 5)

head = node(5 , None)



return head

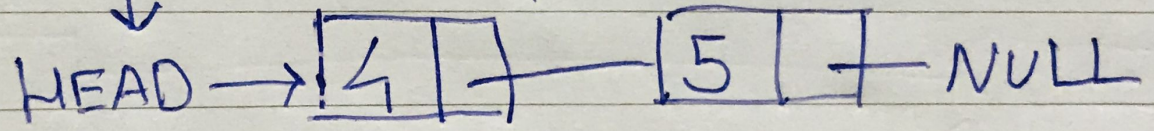
Now start = head , node (5)

return head

Now start = head , node (5)

2. start = push (start , 4)

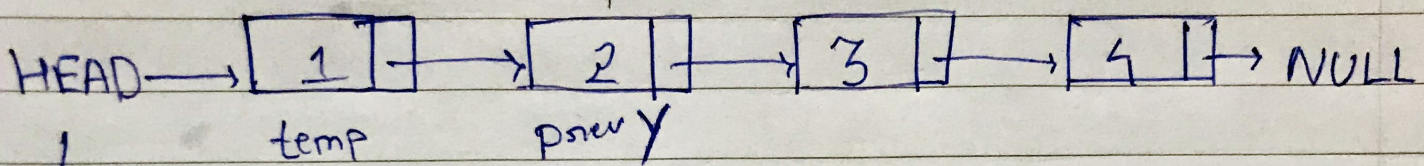
head = node (4 , node(5))



return head

Now, start = node (4)

Simpler approach for Swap :-

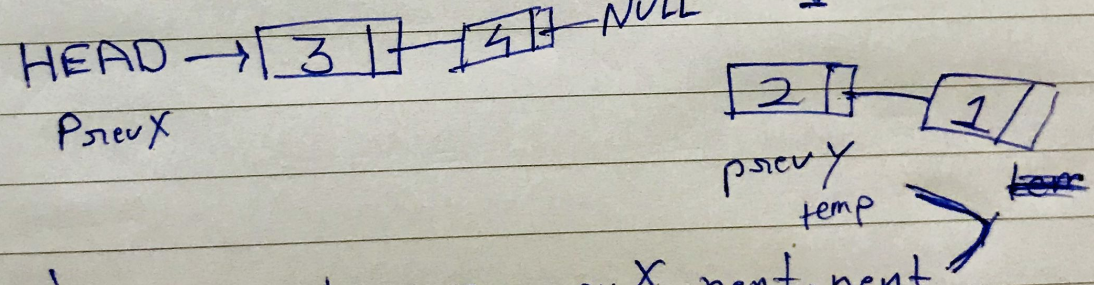


prevX

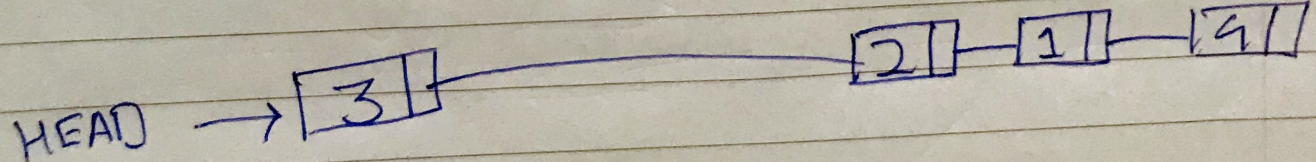
Swap (1, 3)

x y

1) temp = prevX.next
prevX.next = prevY.next
prevY.next = temp



2) temp = prevX.next.next
prevX.next.next = prevY.next.next
prevY.next.next = temp

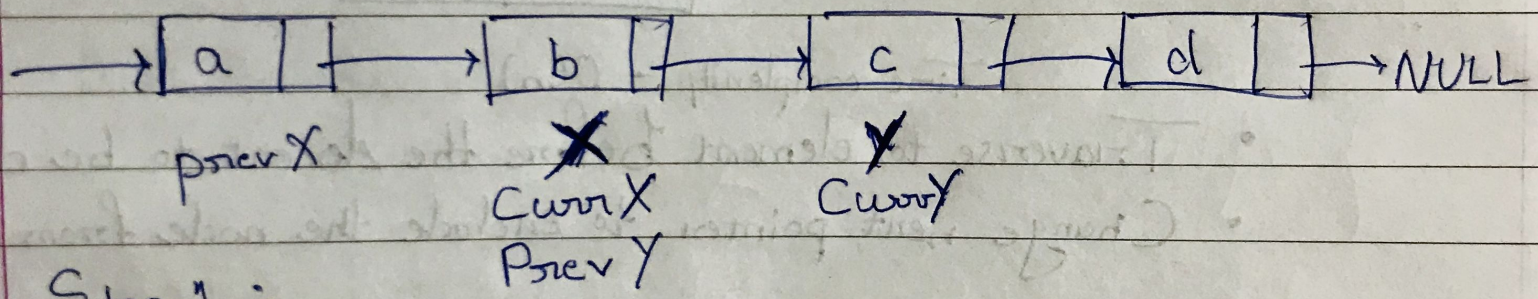


Problems :-

Single Linked List

Swap Nodes of a Linked List :-

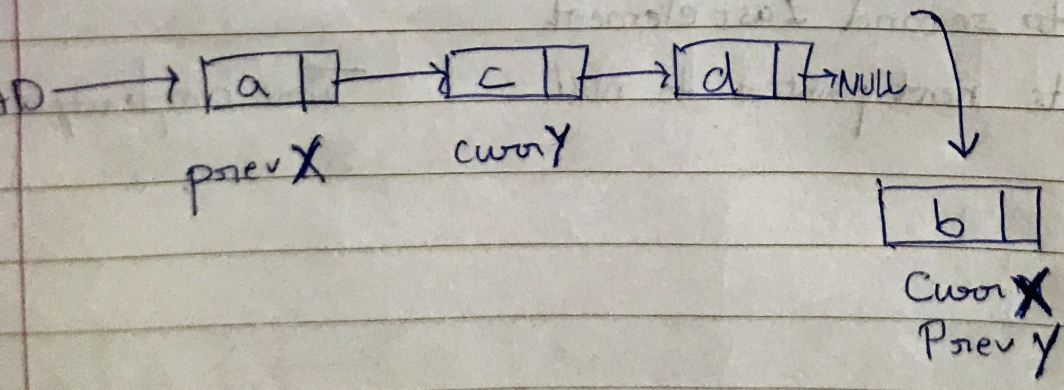
I. If X and Y both Not head



Step 1 :

prevX.next = currY → for X is not head

prevY.next = currX → for Y is not head

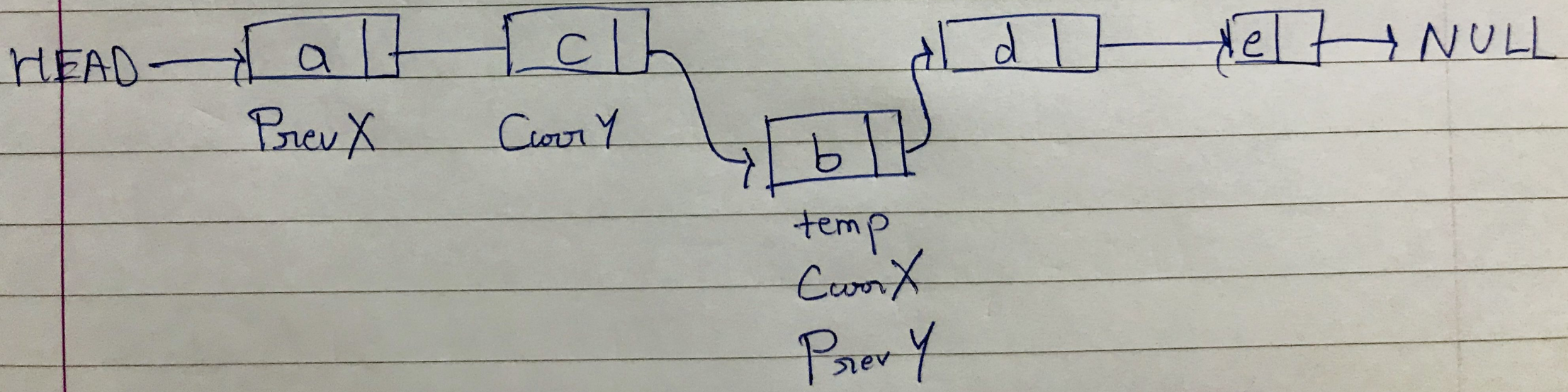


Step 2 :-

temp = CurrX.next

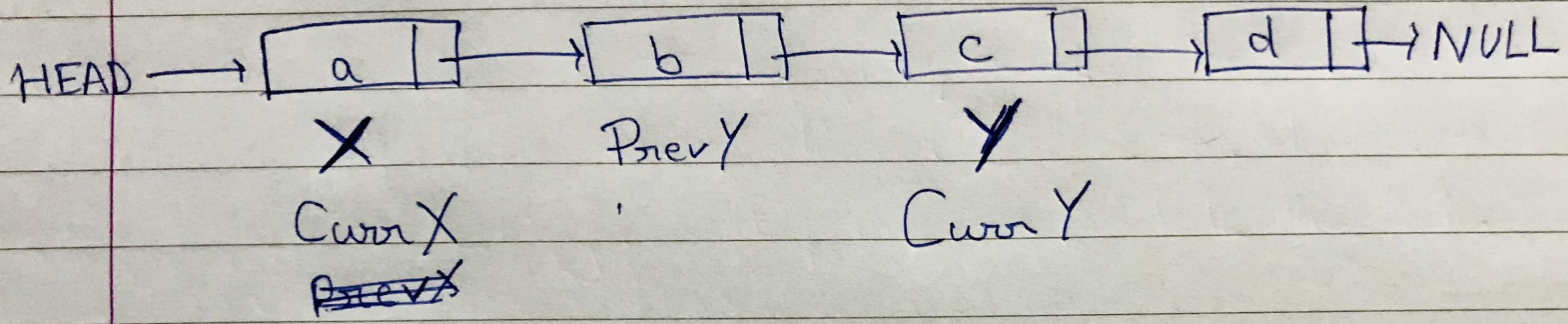
CurrX.next = CurrY.next

CurrY.next = temp



Condition II :-

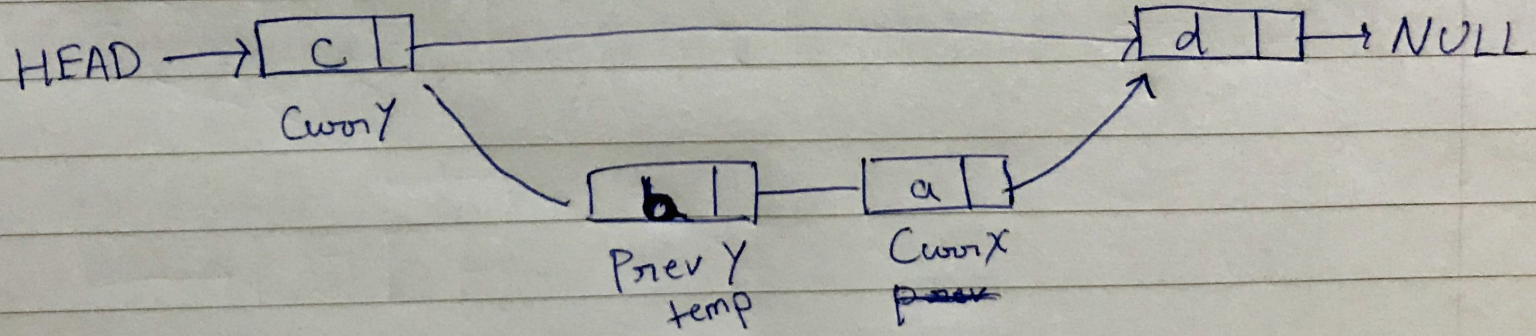
One is head Node (X or Y)



Step 1 :-

self.head = Curr Y → for X is head
Make Y as head

prev Y.next = Curr X → for Y not head



Step-2 :-

temp = Curr X.next
Curr X.next = Curr Y.next
Curr Y.next = temp