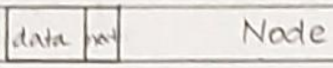
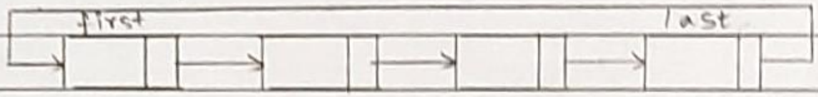
CIRCULAR LINKED LIST

- * Although a singly linked list is a useful and popular data structure, it has some shortcomings
- * Given a ptr A to a node in a linear list, we cannot reach any of the nodes that precede the node to which A is pointing.
- * This disadvantage can be overcome by making a small change; without affecting the data structure
- * The pointer field of the last node can be set to point to the first node rather than NULL.





- * From any node in such a list, it is possible to reach any other node in the list

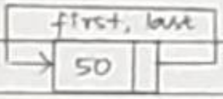
OPERATIONS

1. Insert
2. Display
3. Length
4. Search
5. Delete
6. Reverse
7. Sort

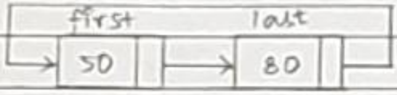
1. INSERT IN CLL

first = last = NULL

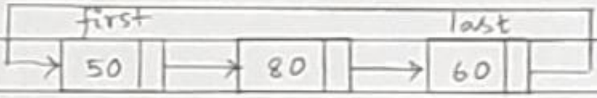
Insert (50)



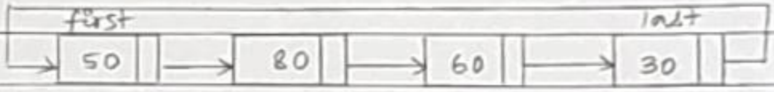
Insert (80)



Insert (60)

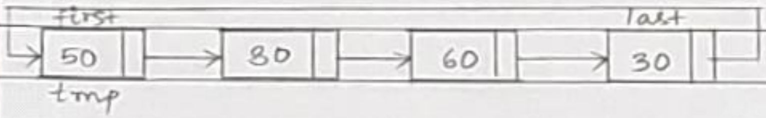


Insert (30)

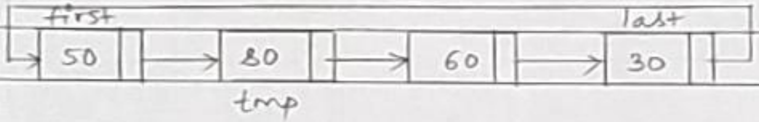


2. DISPLAY THE CLL

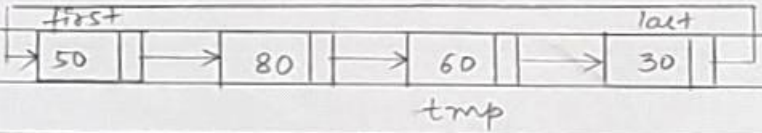
Output Screen



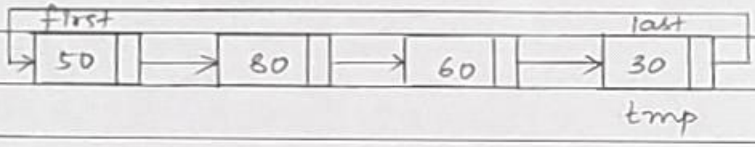
50 →



50 → 80 →



50 → 80 → 60 →



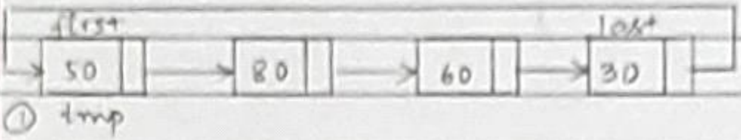
50 → 80 → 60 → 30 →

tmp = first

50 → 80 → 60 → 30 → first

3. LENGTH OF THE CLL

cnt = 0



cnt = 1

② tmp cnt = 2

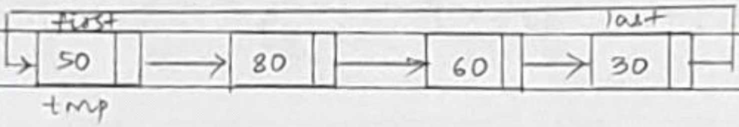
③ tmp cnt = 3

④ tmp cnt = 4

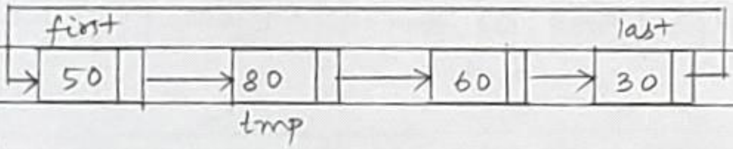
tmp = first

4. SEARCH FOR A NODE

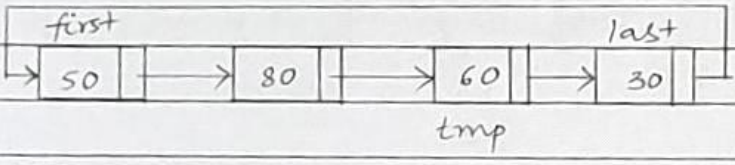
Node not found flag = 0, x = 20



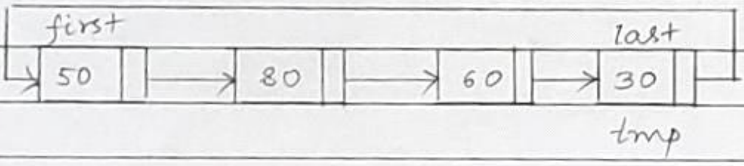
tmp → data != x



tmp → data != x



tmp → data != x

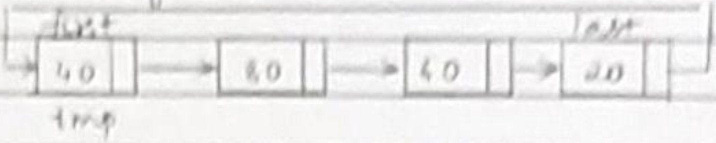


tmp → data != x

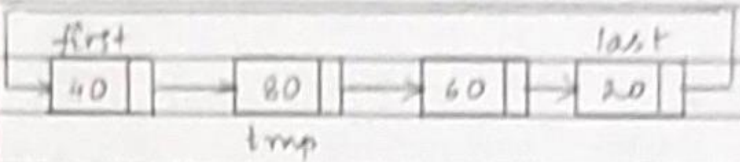
tmp = first x not found

Node found

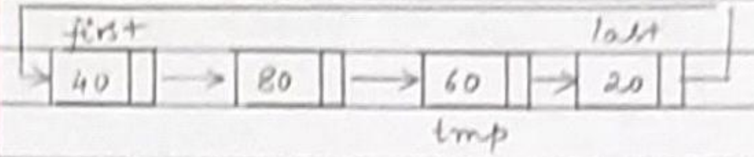
flag = 0 x = 60



tmp->data != x



tmp->data != x



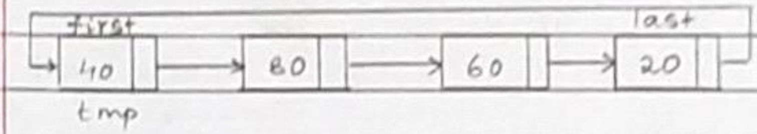
tmp->data == x

flag = 1

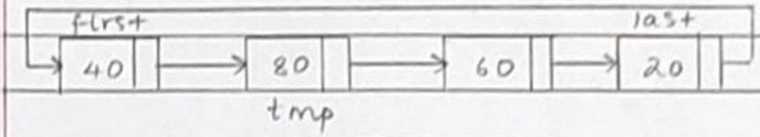
5. DELETE A NODE

Node not found

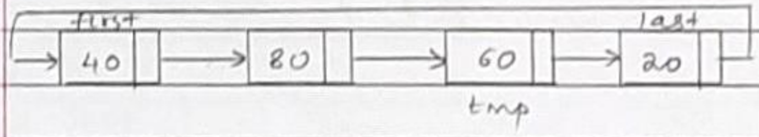
x = 10



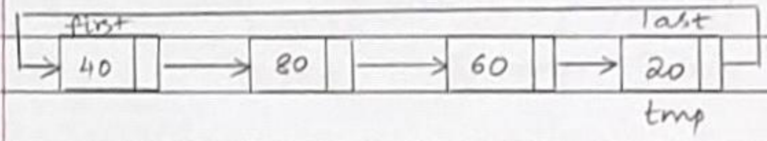
tmp->data != x



tmp->data != x



tmp->data != x

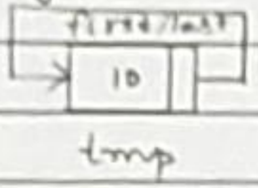


tmp->data != x

tmp = first

x not found.

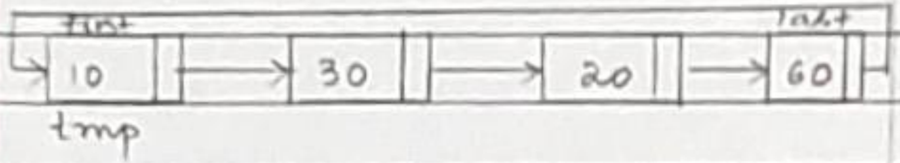
Single Node Deletion

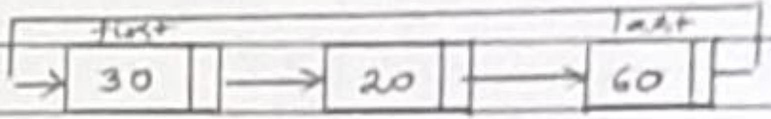


$x = 10$

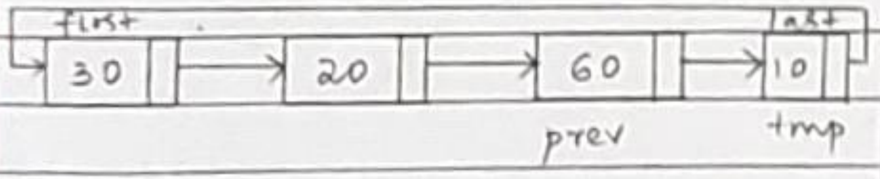
first = last = NULL

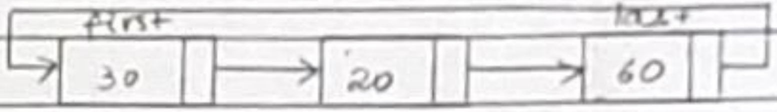
Head Node Deletion



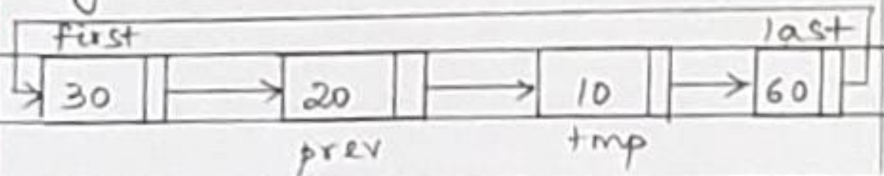


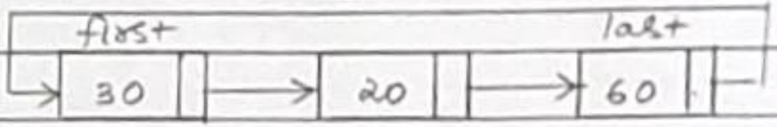
Last Node Deletion





Any other Node


first
last
30
20
10
60
prev
tmp


first
last
30
20
60