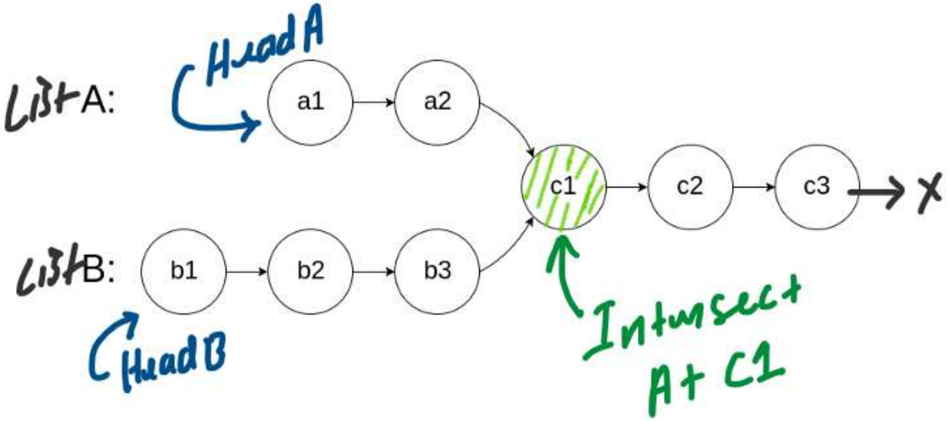
HW 03: Intersection of Two Linked Lists (Leetcode-160)

PROBLEM STATEMENT:

*Given the heads of two singly linked-lists **headA** and **headB**, return the node at which the two lists intersect. If the two linked lists have no intersection at all, return **null**.*

For example,

the following two linked lists begin to intersect at node c1:

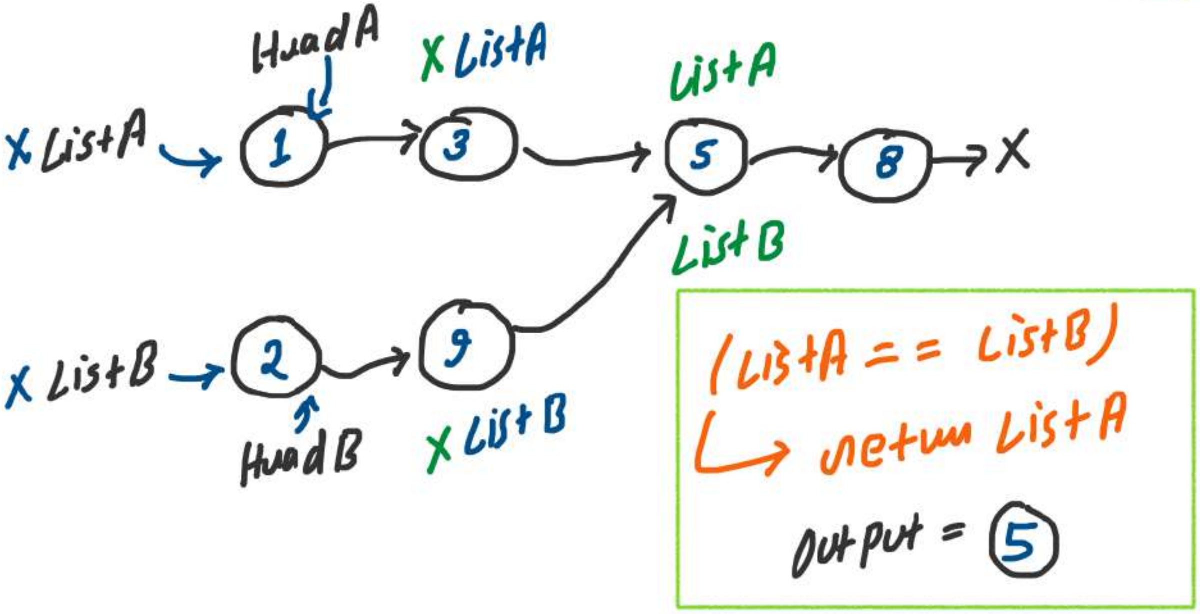


***Note** that the linked lists must retain their original structure after the function returns.*

Ex:1 DRY RUN

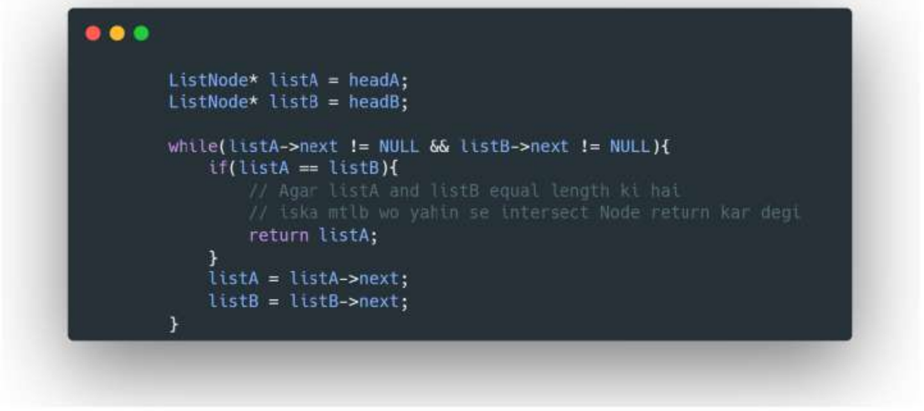
Equal Length of ListA & ListB

ListALength = ListBLength = 4



(ListA == ListB)
↳ return ListA

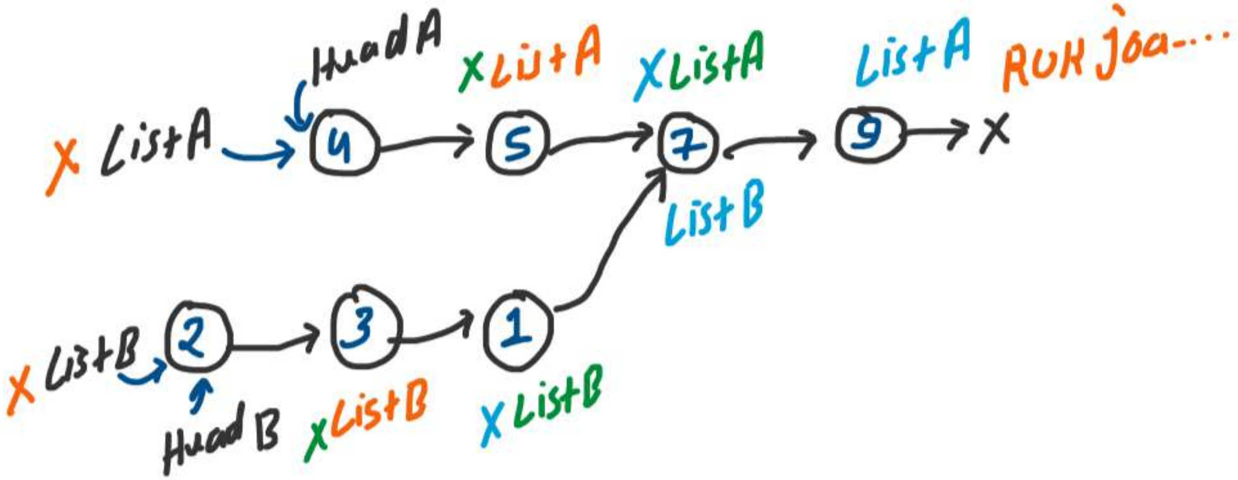
Output = 5

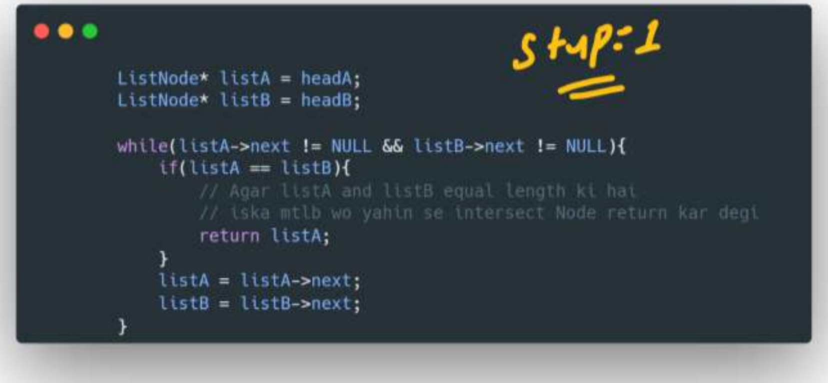


```
ListNode* listA = headA;
ListNode* listB = headB;

while(listA->next != NULL && listB->next != NULL){
    if(listA == listB){
        // Agar listA and listB equal length ki hai
        // iska mtlb wo yahin se intersect Node return kar degi
        return listA;
    }
    listA = listA->next;
    listB = listB->next;
}
```

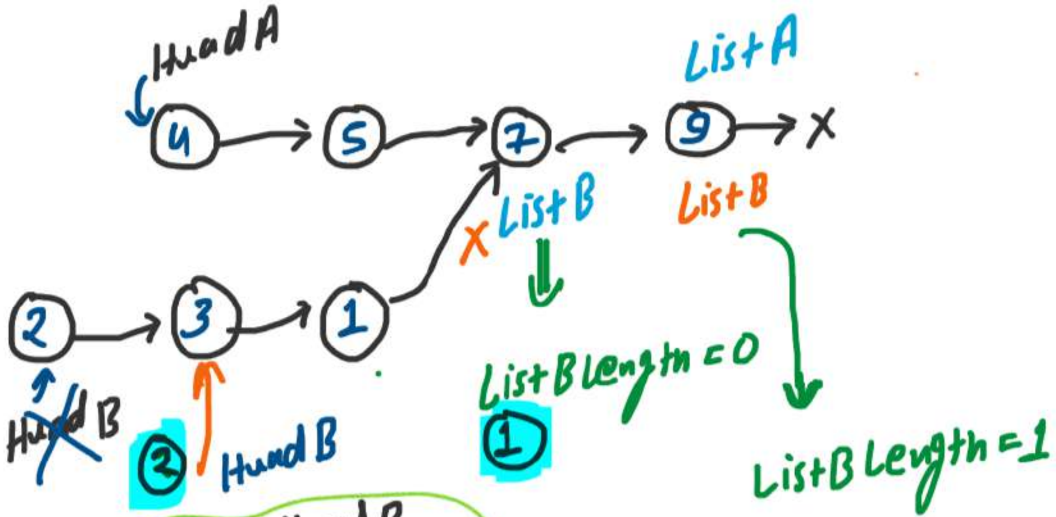
Ex: 2   DRY RUN

NO Equal Length of ListA & ListB





```
ListNode* listA = headA;
ListNode* listB = headB;

while(listA->next != NULL && listB->next != NULL){
    if(listA == listB){
        // Agar listA and listB equal length ki hai
        // iska mtlb wo yahin se intersect Node return kar degi
        return listA;
    }
    listA = listA->next;
    listB = listB->next;
}
```

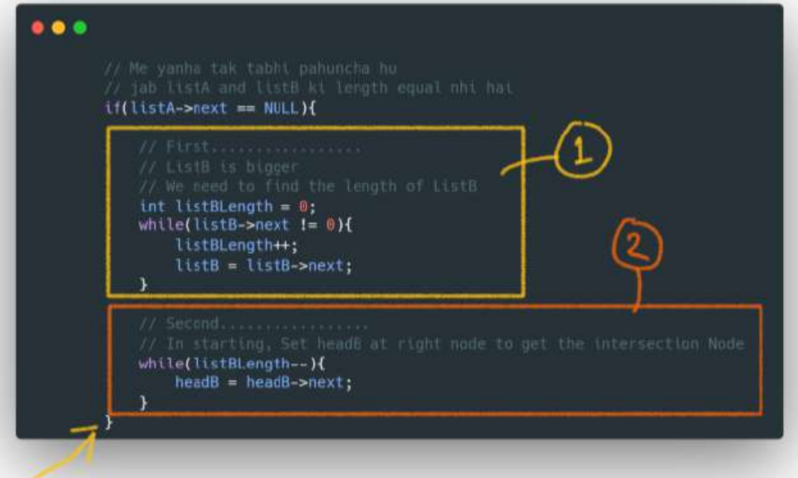


Agar HeadB ko iss ③ node par set kardi To Intersection Node Mil jayega

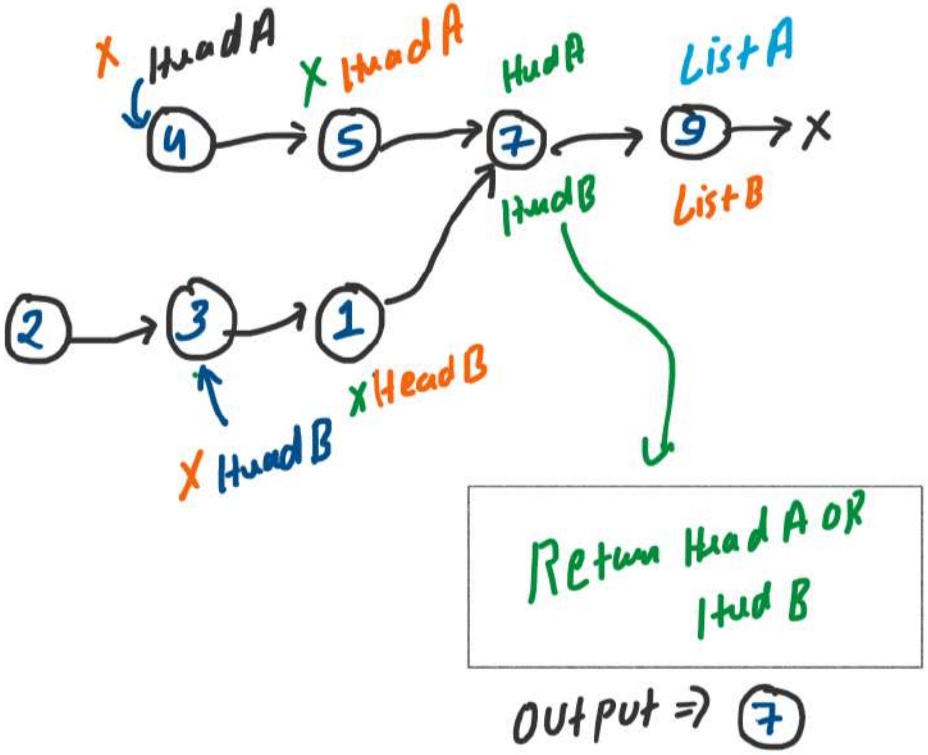
But How

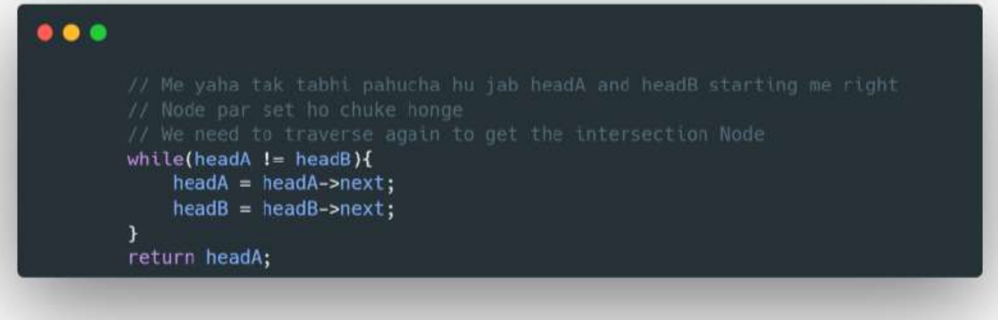
① Find Diff b/w both list

② Set at head at right position.



```
// Me yanha tak tabhi pahuncha hu
// jab listA and listB ki length equal nhi hai
if(listA->next == NULL){
    // First.................
    // ListB is bigger
    // We need to find the length of ListB
    int listBLength = 0;
    while(listB->next != 0){
        listBLength++;
        listB = listB->next;
    }
    // Second.................
    // In starting, Set headB at right node to get the intersection Node
    while(listBLength--){
        headB = headB->next;
    }
}
```

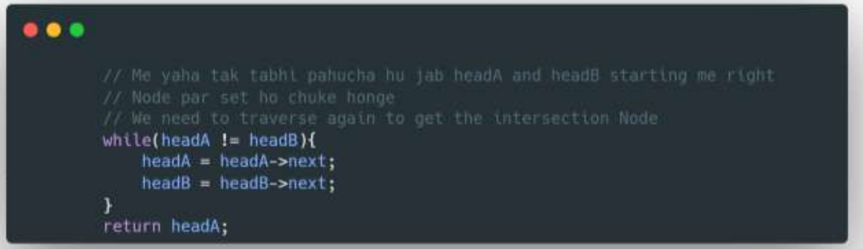




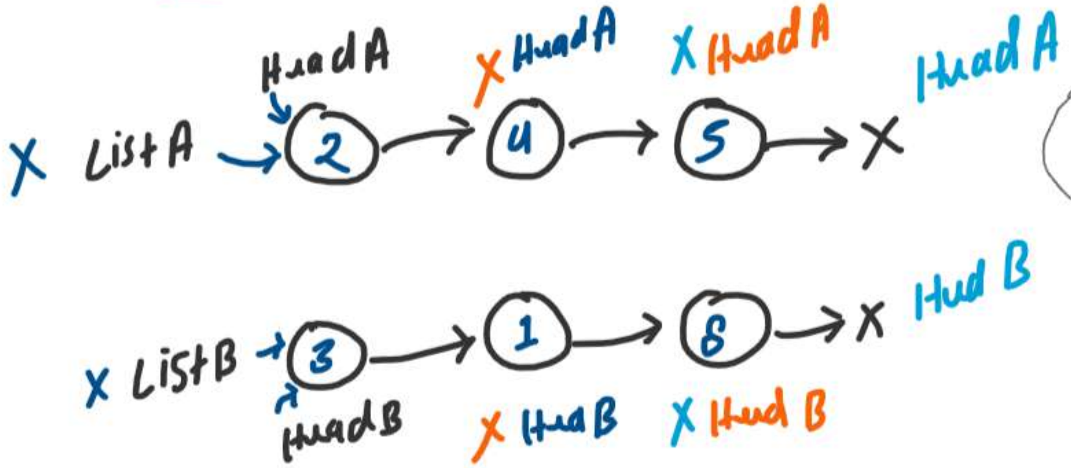
```
// Me yaha tak tabhi pahucha hu jab headA and headB starting me right
// Node par set ho chuke honge
// We need to traverse again to get the intersection Node
while(headA != headB){
    headA = headA->next;
    headB = headB->next;
}
return headA;
```

Ex:3 DRY RUN

No Intersection b/w both Lists



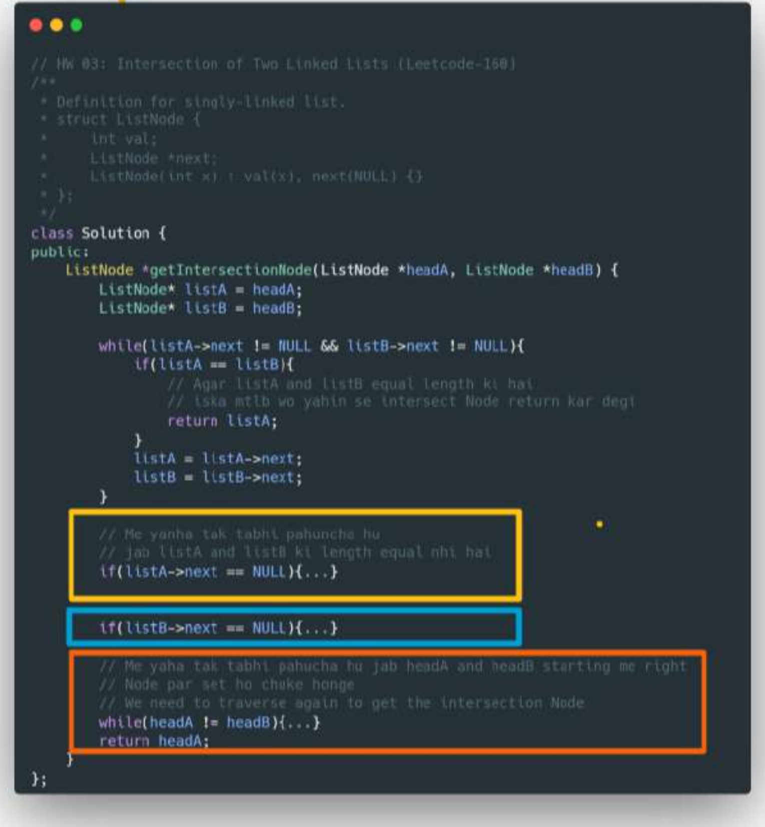
```
// Me yaha tak tabhi pahucha hu jab headA and headB starting me right
// Node par set ho chuke honge
// We need to traverse again to get the intersection Node
while(headA != headB){
    headA = headA->next;
    headB = headB->next;
}
return headA;
```



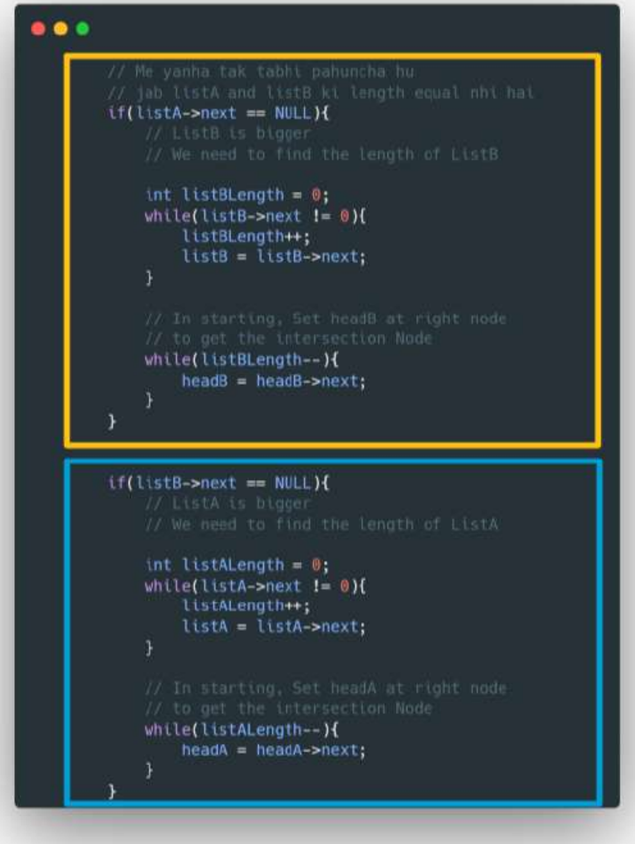
Return HeadA
OR
HeadB

Output = Null

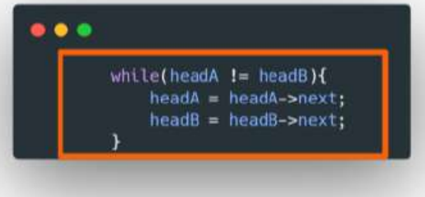
Complete code



```
// HW 03: Intersection of Two Linked Lists (Leetcode-160)
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode *getIntersectionNode(ListNode *headA, ListNode *headB) {
        ListNode* listA = headA;
        ListNode* listB = headB;

        while(listA->next != NULL && listB->next != NULL){
            if(listA == listB){
                // Agar listA and listB equal length ki hai
                // iska mtlb wo yahin se intersect Node return kar degi
                return listA;
            }
            listA = listA->next;
            listB = listB->next;
        }

        // Me yanha tak tabhi pahuncha hu
        // jab listA and listB ki length equal nhi hai
        if(listA->next == NULL){...}

        if(listB->next == NULL){...}

        // Me yaha tak tabhi pahucha hu jab headA and headB starting me right
        // Node par set ho chuke honge
        // We need to traverse again to get the intersection Node
        while(headA != headB){...}
        return headA;
    }
};
```



```
    // Me yanha tak tabhi pahuncha hu
    // jab listA and listB ki length equal nhi hai
    if(listA->next == NULL){
        // ListB is bigger
        // We need to find the length of ListB

        int listBLength = 0;
        while(listB->next != 0){
            listBLength++;
            listB = listB->next;
        }

        // In starting, Set headB at right node
        // to get the intersection Node
        while(listBLength--){
            headB = headB->next;
        }
    }

    if(listB->next == NULL){
        // ListA is bigger
        // We need to find the length of ListA

        int listALength = 0;
        while(listA->next != 0){
            listALength++;
            listA = listA->next;
        }

        // In starting, Set headA at right node
        // to get the intersection Node
        while(listALength--){
            headA = headA->next;
        }
    }
```



```
    while(headA != headB){
        headA = headA->next;
        headB = headB->next;
    }
```

T.C. ⇒ O(N)

S.C. ⇒ O(1)