### Remove duplicate element from sorted Linked List

Given a singly linked list consisting of **N** nodes. The task is to remove duplicates (nodes with duplicate values) from the given list (if exists).  
**Note:** Try not to use extra space. Expected time complexity is **O(N)**. The nodes are arranged in a **sorted**way.

**Example 1:**

**Input:**

LinkedList: 2->2->4->5

**Output:** 2 4 5

**Explanation:** In the given linked list

2 ->2 -> 4-> 5, only 2 occurs more

than 1 time.

**Example 2:**

**Input:**

LinkedList: 2->2->2->2->2

**Output:** 2

**Explanation:** In the given linked list

2 ->2 ->2 ->2 ->2, 2 is the only element

and is repeated 5 times.

**Expected Time Complexity** : O(N)  
**Expected Auxilliary Space** : O(1)

**Constraints:**  
1 <= Number of nodes <= 104

**Company Tags**

[**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**OYO Rooms**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=OYO%20Rooms) [**Oracle**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Oracle) [**Visa**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Visa) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**Myntra**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Myntra)

//{ Driver Code Starts

/\* package whatever; // don't place package name! \*/

import java.util.\*;

class Node

{

int data;

Node next;

Node(int d) {data = d; next = null; }

}

class Remove\_Duplicate\_From\_LL

{

Node head;

Node tail;

public void addToTheLast(Node node)

{

if (head == null)

{

head = node;

tail = node;

}

else

{

tail.next = node;

tail = node;

}

}

void printList()

{

Node temp = head;

while (temp != null)

{

System.out.print(temp.data+" ");

temp = temp.next;

}

System.out.println();

}

/\* Drier program to test above functions \*/

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int t=sc.nextInt();

while(t>0)

{

int n = sc.nextInt();

Remove\_Duplicate\_From\_LL llist = new Remove\_Duplicate\_From\_LL();

int a1=sc.nextInt();

Node head= new Node(a1);

llist.addToTheLast(head);

for (int i = 1; i < n; i++)

{

int a = sc.nextInt();

llist.addToTheLast(new Node(a));

}

CodingMaxima g = new CodingMaxima ();

llist.head = g.removeDuplicates(llist.head);

llist.printList();

t--;

}

}}

// } Driver Code Ends

class CodingMaxima

{

//Function to remove duplicates from sorted linked list.

Node removeDuplicates(Node head)

{

if(head==null)

return null;

Set<Integer> hm=new HashSet<>();

Node current=head;

Node prev=null;

while(current != null){

if(hm.contains(current.data)){

prev.next=current.next;

}

else{

hm.add(current.data);

prev=current;

}

current=current.next;

}

return head;

}

}