https://course.acciojob.com/idle?question=dbee5570-e9a9-4ebb-a0dc-1d104f838536

- EASY
- Max Score: 30 Points

Remove Duplicates from Sorted List

You are given the `head` node of sorted linked list, your task is to delete all duplicates such that each element appears only once, return the `head` node once you delete duplicate nodes.

Input Format

The first line of input contains a single integer `n`.

The second line of input contains `n` space seperated integer.

Output Format

Return the head node of linked list after removing duplicates.

Example 1

Input

```
3
1 1 2
```

Output

```
1 2
```

Explanation

```
The linked list given is
```

```
1->1->2

after removing the duplicate element we get 1->2
```

Example 2

Input

```
5
1 1 2 3 3
```

Output

```
1 2 3
```

Explanation

```
The linked list given are

1->1->2->3->3

after removing duplicate elements we get 1->2->3
```

Constraints

1 <= n <= 300

-100 <= node.value <= 100

Topic Tags

- **Linked lists**

My code

// n java

```
import java.util.*;

class Node {
    int data;
    Node next;

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

class Main {
    static Node head;
    static Node temp;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

            int n = sc.nextInt();
            int a1 = sc.nextInt();
            head = null;
            temp = null;
            addToTheLast(a1);

            for (int i = 1; i < n; i++) {
                int data = sc.nextInt();
                addToTheLast(data);
            }
            Solution ans = new Solution();
```

```
            Node node = ans.solve(head);
            printList(node);
            System.out.println();
            sc.close();
    }

    public static void addToTheLast(int data) {
        if (head == null) {
            head = new Node(data);
            temp = head;
            return;
        } else {
            Node new_node = new Node(data);
            temp.next = new_node;
            temp = temp.next;
            return;
        }
    }

    public static void printList(Node node) {
        while (node != null) {
            System.out.print(node.data + " ");
            node = node.next;
        }
    }
}

class Solution {
    public Node solve(Node head) {
        // your code here
```

```
        HashMap<Integer,Integer> hm= new HashMap<>();
        Node ans=head;
        hm.put(head.data,1);
        head=head.next;
        Node p=ans;
        p.next=null;
        while(head!=null)
            {
            if(hm.containsKey(head.data))
                head=head.next;
            else
            {
                hm.put(head.data,1);
                p.next=head;
                head=head.next;
                p=p.next;
           p.next=null;
            }

            }
        return ans;

    }
}
```