

This challenge is part of a tutorial track by [MyCodeSchool](#)

You're given the pointer to the head node of a sorted linked list, where the data in the nodes is in ascending order. Delete as few nodes as possible so that the list does not contain any value more than once. The given head pointer may be null indicating that the list is empty.

Input Format

You have to complete the `SinglyLinkedListNode* removeDuplicates(SinglyLinkedListNode* head)` method which takes one argument - the head of the sorted linked list. You should NOT read any input from stdin/console.

The input is handled by the code in the editor and the format is as follows:

The first line contains an integer t , denoting the number of test cases. The format for each test case is as follows:

The first line contains an integer n , denoting the number of elements in the linked list. The next n lines contain an integer each, denoting the elements of the linked list.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 1000$
- $1 \leq list_i \leq 1000$

Output Format

Delete as few nodes as possible to ensure that no two nodes have the same data. Adjust the next pointers to ensure that the remaining nodes form a single sorted linked list. Then return the head of the sorted updated linked list. Do NOT print anything to stdout/console.

The output is handled by the code in the editor and the format is as follows: For each test case, print in a new line, the data of the linked list after removing the duplicates separated by space.

Sample Input

```
1
5
1
2
2
3
4
```

Sample Output

```
1 2 3 4
```

Explanation

The initial linked list is: 1 -> 2 -> 2 -> 3 -> 4 -> NULL

The final linked list is: 1 -> 2 -> 3 -> 4 -> NULL