This challenge is part of a tutorial track by MyCodeSchool

Given pointers to the heads of two sorted linked lists, merge them into a single, sorted linked list. Either head pointer may be null meaning that the corresponding list is empty.

## Example

headA refers to 1 
ightarrow 3 
ightarrow 7 
ightarrow NULL

headB refers to 1 o 2 o NULL

The new list is  $1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow NULL$ 

#### **Function Description**

Complete the mergeLists function in the editor below.

mergeLists has the following parameters:

- SinglyLinkedListNode pointer headA: a reference to the head of a list
- SinglyLinkedListNode pointer headB: a reference to the head of a list

#### Returns

SinglyLinkedListNode pointer: a reference to the head of the merged list

### Input Format

The first line contains an integer t, the number of test cases.

The format for each test case is as follows:

The first line contains an integer n, the length of the first linked list.

The next n lines contain an integer each, the elements of the linked list.

The next line contains an integer m, the length of the second linked list.

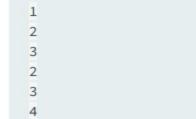
The next m lines contain an integer each, the elements of the second linked list.

### Constraints

- $1 \le t \le 10$
- $1 \le n, m \le 1000$
- $1 \leq list[i] \leq 1000$ , where list[i] is the  $i^{th}$  element of the list.



3



## Sample Output

1 2 3 3 4

# Explanation

The first linked list is: 1 
ightarrow 3 
ightarrow 7 
ightarrow NULL

The second linked list is: 3 o 4 o NULL

Hence, the merged linked list is: 1 o 2 o 3 o 3 o 4 o NULL