

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 $1 \rightarrow 3 \rightarrow 7 \rightarrow NULL$

headB refers to $1 \rightarrow 2 \rightarrow NULL$

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

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 1000$
- $1 \leq list[i] \leq 1000$, where *list[i]* is the *ith* element of the list.

Sample Input

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

Sample Output

```
1 2 3 3 4
```

Explanation

The first linked list is: $1 \rightarrow 3 \rightarrow 7 \rightarrow NULL$

The second linked list is: $3 \rightarrow 4 \rightarrow NULL$

Hence, the merged linked list is: $1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow NULL$