

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

You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. The lists are equal only if they have the same number of nodes and corresponding nodes contain the same data. Either head pointer given may be null meaning that the corresponding list is empty.

### Input Format

You have to complete the `int CompareLists(Node* headA, Node* headB)` method which takes two arguments - the heads of the two linked lists to compare. 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  $t$ , 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 first linked list.

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

The next line contains an integer  $m$ , denoting the number of elements in the second linked list.

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

### Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 1000$
- $1 \leq list_i \leq 1000$ , where  $list_i$  is the  $i^{th}$  element in the list.

### Output Format

Compare the two linked lists and return 1 if the lists are equal. Otherwise, return 0. Do NOT print anything to stdout/console.

The output is handled by the code in the editor and it is as follows:

For each test case, in a new line, print **1** if the two lists are equal, else print **0**.

### Sample Input

```
2
2
1
2
1
1
2
1
2
2
1
2
```

### Sample Output

```
0
1
```

### Explanation

In the first case, linked lists are: 1 -> 2 -> NULL and 1 -> NULL

In the second case, linked lists are: 1 -> 2 -> NULL and 1 -> 2 -> NULL