

Compare two linked lists

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. If all data attributes are equal and the lists are the same length, return **1**. Otherwise, return **0**.

Example

llist1 = 1 → 2 → 3 → *NULL*

llist2 = 1 → 2 → 3 → 4 → *NULL*

The two lists have equal data attributes for the first **3** nodes. *llist2* is longer, though, so the lists are not equal. Return **0**.

Function Description

Complete the *compare_lists* function in the editor below.

compare_lists has the following parameters:

- *SinglyLinkedListNode llist1*: a reference to the head of a list
- *SinglyLinkedListNode llist2*: a reference to the head of a list

Returns

- *int*: return 1 if the lists are equal, or 0 otherwise

Input Format

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

Each of the test cases has the following format:

The first line contains an integer *n*, the number of nodes in the first linked list.

Each of the next *n* lines contains an integer, each a value for a data attribute.

The next line contains an integer *m*, the number of nodes in the second linked list.

Each of the next *m* lines contains an integer, each a value for a data attribute.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 1000$
- $1 \leq llist1[i], llist2[i] \leq 1000$

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

There are $t = 2$ test cases, each with a pair of linked lists.

- 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