This challenge is part of a tutorial track by MyCodeSchool and is accompanied by a video lesson.

You are given the pointer to the head node of a linked list and you need to print all its elements in reverse order from tail to head, one element per line. The head pointer may be null meaning that the list is empty - in that case, do not print anything!

Input Format

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

The first line of input contains t, the number of test cases.
The input of each test case is as follows:

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

Constraints

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

Output Format

Complete the reversePrint function in the editor below and print the elements of the linked list in the reverse order, each in a new line.

Sample Input

```
3
5
16
12
4
2
5
3
7
3
9
5
5
1
18
3
13
```

Sample Output

```
5
2
4
12
16
9
3
7
13
3
18
1
5
```

Explanation

There are three test cases.
The first linked list has 5 elements: `16 -> 12 -> 4 -> 2 -> 5`. Printing this in reverse order will produce: `5 -> 2 -> 4 -> 12 -> 16`.
The second linked list has 3 elements: `7 -> 3 -> 9`. Printing this in reverse order will produce: `9 -> 3 -> 7`.
The third linked list has 5 elements: `5 -> 1 -> 18 -> 3 -> 13`. Printing this in reverse order will produce: `13 -> 3 -> 18 -> 1 -> 5`.