

This challenge is part of a tutorial track by [MyCodeSchool](#) and is accompanied by a video lesson.

Youâ€™re given the pointer to the head node of a linked list. Change the `next` pointers of the nodes so that their order is reversed. The head pointer given may be null meaning that the initial list is empty.

### Input Format

You have to complete the `SinglyLinkedListNode reverse(SinglyLinkedListNode head)` method which takes one argument - the head of the linked list. 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 an integer  $t$ , denoting the number of test cases.

Each test case is of the following format:

The first line contains an integer  $n$ , denoting the number of elements in the linked list.

The next  $n$  lines contain an integer each, denoting the elements of the 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

Change the `next` pointers of the nodes that their order is reversed and return the head of the reversed linked list. Do NOT print anything to `stdout/console`.

The output is handled by the code in the editor. The output format is as follows:

For each test case, print in a new line the elements of the linked list after reversing it, separated by spaces.

### Sample Input

```
1
5
1
2
3
4
5
```

### Sample Output

```
5 4 3 2 1
```

### Explanation

The initial linked list is: 1 -> 2 -> 3 -> 4 -> 5 -> NULL

The reversed linked list is: 5 -> 4 -> 3 -> 2 -> 1 -> NULL