Problem A. Reverse a linked list

OS Linux

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

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.

Example
$head$ references the list $1 \rightarrow 2 \rightarrow 3 \rightarrow NULL$

Manipulate the $next$ pointers of each node in place and return $head$, now referencing the head of the list $3 \rightarrow 2 \rightarrow 1 \rightarrow NULL$.

Function Description

Complete the *reverse* function in the editor below.

reverse has the following parameter:

- *SinglyLinkedListNode pointer head:* a reference to the head of a list

Returns

- *SinglyLinkedListNode pointer:* a reference to the head of the reversed list

Input Format

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

Each test case has the following format:

The first line contains an integer n, the number of elements in the linked list.
Each of the next n lines contains an integer, the $data$ values of the elements in 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.

Input	Output
1 5 1 2 3 4 5	5 4 3 2 1

Explanation

The initial linked list is: $\mathbf{1} \rightarrow \mathbf{2} \rightarrow \mathbf{3} \rightarrow \mathbf{4} \rightarrow \mathbf{5} \rightarrow \boldsymbol{NULL}$.

The reversed linked list is: $\mathbf{5} \rightarrow \mathbf{4} \rightarrow \mathbf{3} \rightarrow \mathbf{2} \rightarrow \mathbf{1} \rightarrow \boldsymbol{NULL}$.