Reverse a doubly linked list ☆

Problem Submissions Leaderboard Editorial

This challenge is part of a tutorial track by MyCodeSchool

You're given the pointer to the head node of a doubly linked list. Reverse the order of the nodes in the list. The head node might be NULL to indicate that the list is empty. Change the next and prev pointers of all the nodes so that the direction of the list is reversed. Return a reference to the head node of the reversed list.

Function Description

Complete the reverse function in the editor below. It should return a reference to the head of your reversed list.

reverse has the following parameter(s):

- head: a reference to the head of a DoublyLinkedList

Input Format

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

Each test case is of the following format:

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

Constraints

- $1 \leq t \leq 10$
- $0 \leq n \leq 1000$
- $0 \leq DoublyLinkedListNode.data \leq 1000$

Output Format

Return a reference to the head of your reversed list. The provided code will print the reverse array as a one line of space-separated integers for each test case.

Sample Input

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

Sample Output

```
4 3 2 1
```

Explanation

The initial doubly linked list is: $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \rightarrow NULL$

The reversed doubly linked list is: $4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1 \rightarrow NULL$

Change Theme C++

```
 *
 */
DoublyLinkedListNode* reverse(DoublyLinkedListNode* head) {
    if (!head) return head;
    DoublyLinkedListNode* prev = head;
    DoublyLinkedListNode* current = head->next;
    head->next = nullptr;
    //https://www.youtube.com/watch?v=8oMklEW4WfU&ab_channel=nexTRIE
    //above link is for singlely linked, use for reference

    while(current){
        DoublyLinkedListNode* next = current-> next;
        current->prev = next;
        current->next = prev;
        prev = current;
        current = next ;
    }
    return prev;
}

int main() ···
```

Line: 79 Col: 26

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Sample Test case 2

Input (stdin) Download

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

Your Output (stdout)

```
4 3 2 1
```

Expected Output Download

```
4 3 2 1
```