Reverse a doubly linked list

by harsha_s

Problem | Submissions | Leaderboard | Discussions | 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.

Input Format

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

Output Format

Change the `next` and `prev` pointers of all the nodes so that the direction of the list is reversed. Then `return` the head node of the reversed list. Do NOT print anything to stdout/console.

Sample Input

NULL
NULL <-- 2 <--> 4 <--> 6 --> NULL

Sample Output

```
NULL
NULL <-- 6 <--> 4 <--> 2 --> NULL
```

Explanation

1. Empty list, so nothing to do.
2. 2,4,6 become 6,4,2 o reversing in the given doubly linked list.

Current Buffer (saved locally, editable) Java 7

```
/*
  Insert Node at the end of a linked list
  head pointer input could be NULL as well for empty list
  Node is defined as
  class Node {
     int data;
     Node next;
```

```
8       Node prev;
9     }
10  */
11
12  Node Reverse(Node head) {
13
14  }
15
```

Line: 1 Col: 1

Upload Code as File

Run Code | Submit Code