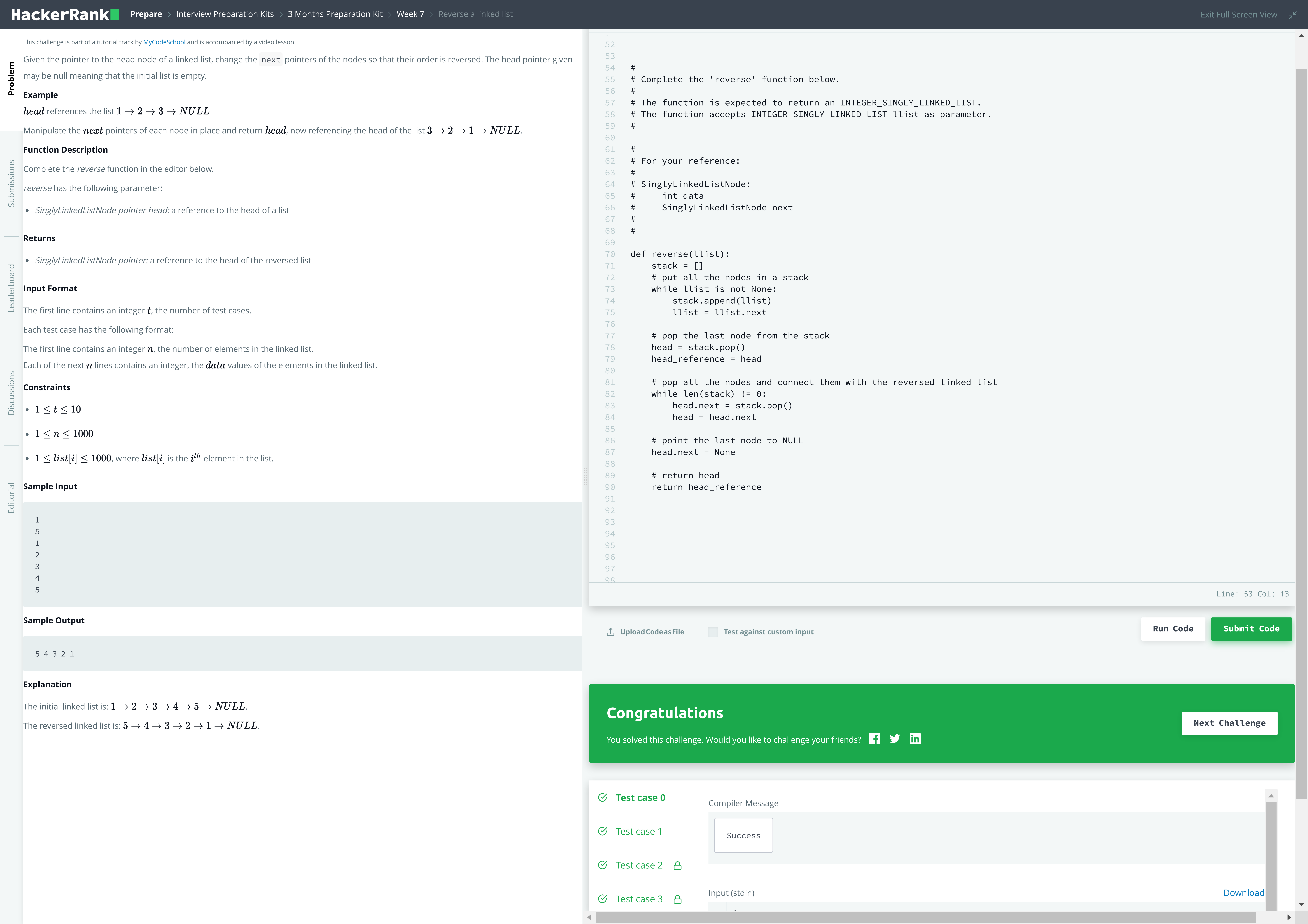
HackerRank
Prepare > Interview Preparation Kits > 3 Months Preparation Kit > Week 7 > Reverse a linked list
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Editorial

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.

Sample Input

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

Sample Output

```
5 4 3 2 1
```

Explanation

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

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

```


#
# Complete the 'reverse' function below.
#
# The function is expected to return an INTEGER_SINGLY_LINKED_LIST.
# The function accepts INTEGER_SINGLY_LINKED_LIST llist as parameter.
#

#
# For your reference:
#
# SinglyLinkedListNode:
#     int data
#     SinglyLinkedListNode next
#
#

def reverse(llist):
    stack = []
    # put all the nodes in a stack
    while llist is not None:
        stack.append(llist)
        llist = llist.next

    # pop the last node from the stack
    head = stack.pop()
    head_reference = head

    # pop all the nodes and connect them with the reversed linked list
    while len(stack) != 0:
        head.next = stack.pop()
        head = head.next

    # point the last node to NULL
    head.next = None

    # return head
    return head_reference
```

Line: 53 Col: 13

Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Input (stdin)