Get Node Value

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 linked list and a specific position. Counting backwards from the tail node of the linked list, get the value of the node at the given position. A position of 0 corresponds to the tail, 1 corresponds to the node before the tail and so on.

Input Format

You have to complete the `int GetNode(Node* head, int positionFromTail)` method which takes two arguments - the head of the linked list and the position of the node from the tail. positionFromTail will be at least 0 and less than the number of nodes in the list. You should NOT read any input from stdin/console.

Constraints

Position will be a valid element in linked list.

Output Format

Find the node at the given position counting backwards from the tail. Then `return` the `data` contained in this node. Do NOT print anything to stdout/console.

Sample Input

```
1 -> 3 -> 5 -> 6 -> NULL, positionFromTail = 0
1 -> 3 -> 5 -> 6 -> NULL, positionFromTail = 2
```

Sample Output

```
6
3
```

Submissions:50940

Max Score:5

Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 7

```
/*
  Get Nth element from the end in a linked list of integers
  Number of elements in the list will always be greater than N.
  Node is defined as
  class Node {
```

```
6       int data;
7       Node next;
8     }
9   */
10
11  int GetNode(Node head,int n) {
12      // This is a "method-only" submission.
13      // You only need to complete this method.
14
15
16  }
17
```

Line: 1 Col: 1

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