Remove Nth Node from List End

Asked in: HCL Amazon

Suggest Edit | Bookmark

Given a linked list, remove the nth node from the end of list and return its head.

For example,
Given linked list: `1->2->3->4->5`, and `n = 2`.
After removing the second node from the end, the linked list becomes `1->2->3->5`.

> ***Note:***
> 1. If n is greater than the size of the list, remove the first node of the list.

Try doing it using constant additional space.

See Expected Output

Seen this question in a real interview before Yes No

```
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     public int val;
 *     public ListNode next;
 *     ListNode(int x) { val = x; next = null; }
 * }
 */
public class Solution {
    public ListNode removeNthFromEnd(ListNode A, int B) {
        ListNode dummy=A;
        int n=0;
        if(A==null || A.next==null && B==1)
        {
            return null;
        }
        while(dummy!=null){
            dummy=dummy.next;
            n++;
        }
        if(B>=n){
            return A.next;
        }
        int k=n-B;

        dummy=A;
```

```
        while(k>1){

            dummy=dummy.next;

            k--;
        }
        dummy.next=dummy.next.next;
        return A;
    }

}
```

Add Two Numbers as Lists

Suggest Edit | Bookmark

Asked in: Amazon | Qualcomm | Microsoft | Facebook

You are given two linked lists representing two non-negative numbers. The digits are stored in **reverse order** and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

Input: `(2 -> 4 -> 3)` + `(5 -> 6 -> 4)`
Output: `7 -> 0 -> 8`

```
342 + 465 = 807
```

Make sure there are no trailing zeros in the output list
So, `7 -> 0 -> 8 -> 0` is not a valid response even though the value is still 807.

See Expected Output

Seen this question in a real interview before Yes No

```
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     public int val;
 *     public ListNode next;
 *     ListNode(int x) { val = x; next = null; }
 * }
 */
import java.util.*;
```

```
public class Solution {
    public ListNode addTwoNumbers(ListNode A, ListNode B) {
        ListNode dummy=new ListNode(0);
        ListNode p=dummy;
        int c=0;
        while(A!=null || B!=null || c!=0){
            if(A!=null){
                c+=A.val;
                A=A.next;
            }
            if(B!=null){
                c+=B.val;
                B=B.next;
            }
            ListNode temp=new ListNode(c%10);
            p.next=temp;
            p=p.next;
            c=c/10;
        }
        return dummy.next;

    }
}
```

Palindrome List

Asked in: Amazon Microsoft

Suggest Edit Bookmark

Given a singly linked list, determine if its a palindrome. Return 1 or 0 denoting if its a palindrome or not, respectively.

Notes:

1. Expected solution is linear in time and constant in space.

For example,

```
List 1-->2-->1 is a palindrome.
List 1-->2-->3 is not a palindrome.
```

See Expected Output

Seen this question in a real interview before Yes No

```
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     public int val;
 *     public ListNode next;
 *     ListNode(int x) { val = x; next = null; }
 * }
 */
public class Solution {
    public int lPalin(ListNode A) {
       ListNode dummy=Reverse_List(A);
       boolean result=isPalindrome(A,dummy);
       if(result){
          return 1;
       }
       return 0;
     }
     ListNode Reverse_List(ListNode A){
        ListNode head=null;
        while(A!=null){
           ListNode temp=new ListNode(A.val);
           temp.next=head;
           head=temp;
           A=A.next;
        }
        return head;
     }

     boolean isPalindrome(ListNode A,ListNode B){
         while(A!=null && B!=null){
            if(A.val!=B.val){
               return false;
            }
            A=A.next;
            B=B.next;
         }
         return A==null && B==null;
     }
}
```