Insert a Node at the Tail of a Linked List ★

14 more points to get your next star!

Rank: 1127668 | Points: 186/200 ⓘ

Problem Solving ★★

Your Insert a Node at the Tail of a Linked List submission got 5.00 points. Share Post

You are now 14 points away from the 3rd star for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem   Submissions   Leaderboard   Editorial 🔒

This challenge is part of a tutorial track by MyCodeSchool and is accompanied by a video lesson.

You are given the pointer to the head node of a linked list and an integer to add to the list. Create a new node with the given integer. Insert this node at the tail of the linked list and return the head node of the linked list formed after inserting this new node. The given head pointer may be null, meaning that the initial list is empty.

Function Description

Complete the *insertNodeAtTail* function with the following parameters:

- *SinglyLinkedListNode pointer head*: a reference to the head of a list

- *int data*: the data value for the node to insert

Returns

- *SinglyLinkedListNode pointer*: reference to the head of the modified linked list

Input Format

The first line contains an integer n, the number of elements in the linked list.

The next n lines contain an integer each, the value that needs to be inserted at tail.

Constraints

- $1 \le n \le 1000$

- $1 \le list_i \le 1000$

Sample Input

```
STDIN    Function
-----    --------
5        size of linked list n = 5
141      linked list data values 141..474
302
164
530
474
```

Sample Output

```
141
302
164
530
474
```

Explanation

First the linked list is NULL. After inserting 141, the list is 141 -> NULL.

After inserting 302, the list is 141 -> 302 -> NULL.

After inserting 164, the list is 141 -> 302 -> 164 -> NULL.

After inserting 530, the list is 141 -> 302 -> 164 -> 530 -> NULL. After inserting 474, the list is 141 -> 302 -> 164 -> 530 -> 474 -> NULL, which is the final list.

Change Theme   Language Python 3

```python
31    #
32    # SinglyLinkedListNode:
33    #     int data
34    #     SinglyLinkedListNode next
35    #
36    #
37    def insertNodeAtTail(head, data):
38        if head is None:
39            return SinglyLinkedListNode(data)
40
41        cur = head
42        pre = None
43        while cur:
44            pre = cur
45            cur = cur.next
46        pre.next = SinglyLinkedListNode(data)
47        return head
48
49    if __name__ == '__main__':
50        fptr = open(os.environ['OUTPUT_PATH'], 'w')
51
52        llist_count = int(input())
53
54        llist = SinglyLinkedList()
55
56        for i in range(llist_count):
57            llist_item = int(input())
58            llist_head = insertNodeAtTail(llist.head, llist_item)
59            llist.head = llist_head
60
61        print_singly_linked_list(llist.head, '\n', fptr)
62        fptr.write('\n')
63
```

EMACS   Line: 47 Col: 16

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

You have earned 5.00 points!

You are now 14 points away from the 3rd star for your problem solving badge.

86%   186/200

Congratulations

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

Next Challenge

✓ **Test case 0**

✓ Test case 1

✓ Test case 2 🔒

✓ Test case 3 🔒

✓ Test case 4 🔒

✓ Test case 5 🔒

✓ Test case 6 🔒

Compiler Message

Success

Input (stdin)   Download

1 | **5**
2 | **141**
3 | **302**
4 | **164**
5 | **530**
6 | **474**

Expected Output   Download

1 | **141**

Blog | Scoring | Environment | FAQ | About Us | Helpdesk | Careers | Terms Of Service | Privacy Policy