Insert a node at a specific position in a linked list *





Your Insert a node at a specific position in a linked list submission got 5.00 points.

Share

Post

X

You are now 4 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

Given a pointer to the head node of a linked list and an integer to insert at a certain position, create a new node with the given integer as its *data* attribute, insert this node at the desired position, and return the head node.

A position of O indicates the head, a position of 1 indicates one node away from the head, and so on. The head pointer given may be null, meaning that the initial list is empty.

Example

head refers to the first node in the list $1 \rightarrow 2 \rightarrow 3$

data = 4

position = 2

Insert a node at position 2 with data = 4. The new list is $1 \rightarrow 2 \rightarrow 4 \rightarrow 3$

Function Description

Complete the function *insertNodeAtPosition* with the following parameters:

- SinglyLinkedListNode pointer llist: a reference to the head of the list
- data: an integer value to insert as data in the new node
- position: an integer position to insert the new node, zero-based indexing

Returns

• SinglyLinkedListNode pointer: a reference to the head of the revised list

Input Format

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

Each of the next ${\pmb n}$ lines contains an integer SinglyLinkedListNode[i].data.

The next line contains an integer data, the data of the node that is to be inserted.

The last line contains an integer **position**.

Constraints

- $1 \le n \le 1000$
- $1 \leq SinglyLinkedListNode[i]. data \leq 1000$, where SinglyLinkedListNode[i] is the i^{th} element of the linked list.
- $0 \le position \le n$.

Sample Input

STDIN	Function
3	n = 3
16	llist = 16->13->7

```
13
7
1 data = 1
2 position = 2
```

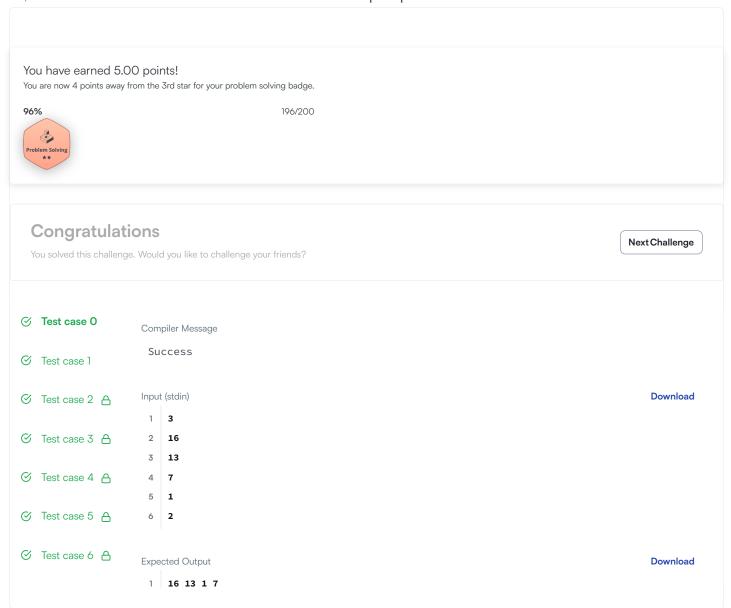
Sample Output

16 13 1 7

Explanation

The initial linked list is $16 \to 13 \to 7$. Insert 1 at the position 2 which currently has 7 in it. The updated linked list is $16 \to 13 \to 1 \to 7$.

```
Change Theme Language Python 3
                                                                                                         1
                                                                                                             22
         # Complete the 'insertNodeAtPosition' function below.
    40
    41
    42
         # The function is expected to return an INTEGER_SINGLY_LINKED_LIST.
         # The function accepts following parameters:
    43
     44
          # 1. INTEGER_SINGLY_LINKED_LIST llist
            2. INTEGER data
     45
     46
         # 3. INTEGER position
     47
    48
    49
    50
         # For your reference:
    51
         # SinglyLinkedListNode:
    52
    53
                int data
                SinglyLinkedListNode next
    54
    55
    56
    57
    58
         def insertNodeAtPosition(llist, data, position):
    59
              # Write your code here
              if not llist:
    60
     61
                  return SinglyLinkedListNode(data)
    62
              cur = llist
    63
    64
              pre = None
              for _ in range(position):
    65
    66
                  pre = cur
    67
                  cur = cur.next
              new_node = SinglyLinkedListNode(data)
    68
    69
              pre.next = new_node
              new_node.next = cur
    70
     71
              return llist
     72
     73
         if
                           main ':--
              name
EMACS
                                                                                                      Line: 48 Col: 1
                                                                                                 Run Code
                                                                                                            Submit Code
 Test against custom input
```



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