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 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 0 indicates 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* in the editor below. It must return a reference to the head node of your finished list.

insertNodeAtPosition has the following parameters:

- *head*: a SinglyLinkedListNode pointer to the head of the list
- *data*: an integer value to insert as data in your 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 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

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

Sample Input

```
3
16
13
7
1
2
```

Sample Output

```
16 13 1 7
```

Explanation

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

Change Theme Language Python 3

```
# The function is expected to return an INTEGER_SINGLY_LINKED_LIST.
# The function accepts following parameters:
#  1. INTEGER_SINGLY_LINKED_LIST llist
#  2. INTEGER data
#  3. INTEGER position
#

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

def insertNodeAtPosition(head, data, position):
    node = head

    while position > 1:
        node = node.next
        position -= 1

                class SinglyLinkedListNode
    next = node.next
    node.next = SinglyLinkedListNode(data)
    node.next.next = next

    return head
```

Line: 92 Col: 5

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

Fetching Results

Test case 0 Test case 4 Test case 8

Test case 1 Test case 5 Test case 9

Test case 2 Test case 6

Test case 3 Test case 7