Get Node Value

This challenge is part of a tutorial track by MyCodeSchool

Given a pointer to the head of a linked list and a specific position, determine the data value at that position. Count backwards from the tail node. The tail is at postion 0, its parent is at 1 and so on.

Example

$head$ refers to $3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow NULL$

$positionFromTail = 2$

Each of the data values matches its distance from the tail. The value 2 is at the desired position.

Function Description

Complete the getNode function in the editor below.

getNode has the following parameters:

- SinglyLinkedListNode pointer head: refers to the head of the list
- int positionFromTail: the item to retrieve

Returns

- int: the value at the desired position

Input Format

The first line contains an integer t, the number of test cases.

Each test case has the following format:

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

The next n lines contains an integer, the data value for an element of the linked list.

The last line contains an integer $positionFromTail$, the position from the tail to retrieve the value of.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 1000$
- $1 \leq list[i] \leq 1000$, where $list[i]$ is the i^{th} element of the linked list.
- $0 \leq positionFromTail < n$

Sample Input

```
2
1
1
0
3
3
2
1
2
```

Sample Output

```
1
3
```

Explanation

In the first case, there is one element in linked list with a value of 1. The last (only) element contains 1.

In the second case, the list is $\mathbf{3} \rightarrow \mathbf{2} \rightarrow \mathbf{1} \rightarrow \boldsymbol{NULL}$. The element with position of 2 from tail contains 3.

Change Theme Language Python 3

```
#
# Complete the 'getNode' function below.
#
# The function is expected to return an INTEGER.
# The function accepts following parameters:
#  1. INTEGER_SINGLY_LINKED_LIST llist
#  2. INTEGER positionFromTail
#

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

def getNode(llist, positionFromTail):
    # Write your code here
    cur = llist
    length = 0
    while cur:
        length += 1
        cur = cur.next

    cnt = length - positionFromTail
    cur = llist
    for _ in range(cnt-1):
        cur = cur.next

    return cur.data

if __name__ == '__main__':…
```

EMACS Line: 40 Col: 1

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

Test case 0

Test case 1

Compiler Message

Success

Input (stdin) Download

Test case 2	2
Test case 3	1
Test case 4	1
Test case 5	0
	3
	3
	2
	1
	2