

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

You're given the pointer to the head node of a linked list, an integer to add to the list and the position at which the integer must be inserted. Create a new node with the given integer, 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.

As an example, if your list starts as 1->2->3 and you want to insert a node at position 2 with data = 4, your new list should be 1->2->4->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

## 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* denoting the data of the node that is to be inserted.

The last line contains an integer *position*.

## Constraints

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

## Output Format

Return a reference to the list head. Locked code prints the list for you.

## Sample Input

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

### Sample Output

```
16 13 1 7
```

### Explanation

The initial linked list is 16 13 7. We have to insert 1 at the position 1 which currently has 13 in it. The updated linked list will be 16 13 1 7