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

This challenge is part of a tutorial track by[MyCodeSchool](http://www.youtube.com/mycodeschool)and is accompanied by a video lesson.

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<= n <=1000
* 1<=SingleLinkedListNode[i].data<=1000, where SingleLinkedListNode[i] is the i^th element of the linked list.
* 0<= position<=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 2 which currently has 7 in it. The updated linked list will be 16 13 1 7

using System.CodeDom.Compiler;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Diagnostics.CodeAnalysis;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Runtime.Serialization;

using System.Text.RegularExpressions;

using System.Text;

using System;

class Solution {

class SinglyLinkedListNode {

public int data;

public SinglyLinkedListNode next;

public SinglyLinkedListNode(int nodeData) {

this.data = nodeData;

this.next = null;

}

}

class SinglyLinkedList {

public SinglyLinkedListNode head;

public SinglyLinkedListNode tail;

public SinglyLinkedList() {

this.head = null;

this.tail = null;

}

public void InsertNode(int nodeData) {

SinglyLinkedListNode node = new SinglyLinkedListNode(nodeData);

if (this.head == null) {

this.head = node;

} else {

this.tail.next = node;

}

this.tail = node;

}

}

static void PrintSinglyLinkedList(SinglyLinkedListNode node, string sep, TextWriter textWriter) {

while (node != null) {

textWriter.Write(node.data);

node = node.next;

if (node != null) {

textWriter.Write(sep);

}

}

}

// Complete the insertNodeAtPosition function below.

/\*

\* For your reference:

\*

\* SinglyLinkedListNode {

\* int data;

\* SinglyLinkedListNode next;

\* }

\*

\*/

static SinglyLinkedListNode insertNodeAtPosition(SinglyLinkedListNode head, int data, int position) {

SinglyLinkedListNode head1=head;

SinglyLinkedListNode temp=new SinglyLinkedListNode(data);

//if note chain is null

if(head==null)

head1=temp;

//insert note at first position

else if(position==0)

{

temp.next=head1;

head1=temp;

}

else

{

//from 1 position to insert position

//switch note to insert position

while(position!=1)

{

head=head.next;

position--;

}

//insert node at current position

temp.next=head.next;

head.next=temp;

}

return head1;

}

static void Main(string[] args) {

TextWriter textWriter = new StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT\_PATH"), true);

SinglyLinkedList llist = new SinglyLinkedList();

int llistCount = Convert.ToInt32(Console.ReadLine());

for (int i = 0; i < llistCount; i++) {

int llistItem = Convert.ToInt32(Console.ReadLine());

llist.InsertNode(llistItem);

}

int data = Convert.ToInt32(Console.ReadLine());

int position = Convert.ToInt32(Console.ReadLine());

SinglyLinkedListNode llist\_head = insertNodeAtPosition(llist.head, data, position);

PrintSinglyLinkedList(llist\_head, " ", textWriter);

textWriter.WriteLine();

textWriter.Flush();

textWriter.Close();

}

}

**Congratulations**

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

[Next Challenge](https://www.hackerrank.com/challenges/insert-a-node-into-a-sorted-doubly-linked-list?h_l=interview&playlist_slugs%5B%5D=interview-preparation-kit&playlist_slugs%5B%5D=linked-lists&h_r=next-challenge&h_v=zen)

* **Test case 0**
* **Test case 1**
* **Test case 2**
* **Test case 3**
* **Test case 4**
* **Test case 5**
* **Test case 6**
* **Test case 7**
* **Test case 8**
* **Test case 9**

Compiler Message

**Success**

Input (stdin)

Download

* **3**
* **16**
* **13**
* **7**
* **1**
* **2**

Expected Output

Download

* **16 13 1 7**