**INSERT NODE AT i-th POSITION**

* Given a singly linked list of integers and position 'i', delete the node present at the 'i-th' position in the linked list recursively.

**Input Format**

* The first line of each test case or query contains the elements of the singly linked list separated by a single space.
* The second line of input contains a single integer depicting the value of 'i'.

**Constraints**

* Time:- 1sec

**Output Format**

* Output will be printed in a seperate line.

**Sample Input 0**

3 4 5 2 6 1 9 -1

3 100

**Sample Output 0**

3 4 5 100 2 6 1 9

//SOURCE CODE

import java.util.\*;

import java.io.\*;

public class Main{

public static void main(String args[]){

Scanner sc=new Scanner (System.in);

LinkedList list=new LinkedList();

while(true){

int n=sc.nextInt();

if(n<0){

break;

}

else{

list.insertAtEnd(n);

}

}

int pos=sc.nextInt();

int val=sc.nextInt();

list.insertAtPos(pos,val);

list.display();

}

}

class LinkedList{

Node head,tail;

LinkedList(){

head=null;

tail=null;

}

class Node{

int data;

Node next;

Node (int val){

data=val;

next=null;

}

}

public void insertAtEnd(int val){

Node n=new Node(val);

if(head==null){

head=n;

tail=n;

}

else{

tail.next=n;

tail=n;

}

}

public void insertAtPos(int pos,int val){

Node n=new Node(val);

if(pos==0){

n.next=head;

head=n;

return;

}

Node temp=head;

for (int i=1;i<pos;i++){

temp=temp.next;

}

n.next=temp.next;

temp.next=n;

}

public void display(){

Node temp=head;

while(temp!=null){

System.out.print(temp.data+" ");

temp=temp.next;

}

}

}