/*You are building a task management system where tasks are stored in a singly linked list. Each task has a unique integer taskID. Tasks are arranged in the order they are added. You are required to implement a function that deletes a task at a given position.

Input Format

• The first line contains an integer n, the number of tasks.

• The second line contains n space-separated integers: the task IDs in the initial list.

• The third line contains an integer position, the index (0-based) of the node to delete.

Output Format

• Print the updated task list after deletion, with each task ID separated by a space.

Sample Input:

5

101 102 103 104 105

2

Sample Output

101 102 104 105

Constraints

• 1 ≤ n ≤ 1000

• 0 ≤ position < n

Task list before deletion:

101 -> 102 -> 103 -> 104 -> 105

After deleting the task at position 2 (i.e., task 103):

101 -> 102 -> 104 -> 105*/

```
package hacckerank;
import java.util.*;
class DeleteTask
{
	int task_id;
	DeleteTask next;
	public DeleteTask(int task_id)
	{
		this.task_id=task_id;
```

```
                this.next=null;
                }
}
public class july30hthr4
{
        DeleteTask head;
        void insertAtEnd(int task_id)
        {

                DeleteTask newNode=new DeleteTask(task_id);
                if(head==null)
                {
                        head=newNode;
                        return;
                }
                DeleteTask temp=head;
                while(temp.next!=null)
                {
                        temp=temp.next;
                }
                temp.next=newNode;



        }
        void deleteAtPostion(int pos)
        {
                if(pos<0||head==null)
                {
                        System.out.println("The position is invalid");
                        return;
                }
```

```
		if(pos==0)
		{
			head=head.next;
			return;
		}
		int index=0;
		DeleteTask temp=head;
		while(temp!=null&&index<pos-1)
		{
			temp=temp.next;
			index++;
		}
		if(temp==null || temp.next==null)
{
   System.out.println("index is out of range");
   return;

}
		temp.next=temp.next.next;

	}
	void display()
	{
		DeleteTask temp=head;
		while(temp!=null)
		{
			System.out.print(temp.task_id+" ");
			temp=temp.next;
		}
	}
```

```
	public static void main(String[] args)
	{
		july30hthr4 dl=new july30hthr4();
		Scanner sc=new Scanner(System.in);
		int n=sc.nextInt();
		for(int i=0;i<n;i++)
		{
			int task_id=sc.nextInt();
			dl.insertAtEnd(task_id);
		}
		int pos=sc.nextInt();
		dl.deleteAtPostion(pos);
		dl.display();




	}

}
```