**Remove loop in Linked List**

Given a linked list of **N** nodes such that it may contain a loop.

A loop here means that the last node of the link list is connected to the node at position X(1-based index). If the link list does not have any loop, X=0.

Remove the loop from the linked list, if it is present, i.e. unlink the last node which is forming the loop.

**Example 1:**

**Input:**

N = 3

value[] = {1,3,4}

X = 2

**Output:** 1

**Explanation:** The link list looks like

1 -> 3 -> 4

^ |

|\_\_\_\_|

A loop is present. If you remove it

successfully, the answer will be 1.

**Example 2:**

**Input:**

N = 4

value[] = {1,8,3,4}

X = 0

**Output:** 1

**Explanation:** The Linked list does not

contains any loop.

**Expected time complexity:** O(N)  
**Expected auxiliary space:**O(1)

**Constraints:**  
1 ≤ N ≤ 10^4

**Company Tags**

[**VMWare**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=VMWare) [**Morgan Stanley**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Morgan%20Stanley) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Snapdeal**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal) [**MakeMyTrip**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=MakeMyTrip) [**Oracle**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Oracle) [**Walmart**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Walmart) [**Goldman Sachs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Goldman%20Sachs) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**Qualcomm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Qualcomm) [**Kuliza**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Kuliza) [**Netskope**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Netskope)

Java code

//{ Driver Code Starts

// driver code

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class Node

{

int data;

Node next;

}

class CodingMaxima

{

public static Node newNode(int data){

Node temp = new Node();

temp.data = data;

temp.next = null;

return temp;

}

public static void makeLoop(Node head, int x){

if (x == 0)

return;

Node curr = head;

Node last = head;

int currentPosition = 1;

while (currentPosition < x)

{

curr = curr.next;

currentPosition++;

}

while (last.next != null)

last = last.next;

last.next = curr;

}

public static boolean detectLoop(Node head){

Node hare = head.next;

Node tortoise = head;

while( hare != tortoise )

{

if(hare==null || hare.next==null) return false;

hare = hare.next.next;

tortoise = tortoise.next;

}

return true;

}

public static int length(Node head){

int ret=0;

while(head!=null)

{

ret += 1;

head = head.next;

}

return ret;

}

public static void main (String[] args){

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while(t--> 0)

{

int n = sc.nextInt();

int num = sc.nextInt();

Node head = newNode(num);

Node tail = head;

for(int i=0; i<n-1; i++)

{

num = sc.nextInt();

tail.next = newNode(num);

tail = tail.next;

}

int pos = sc.nextInt();

makeLoop(head, pos);

Solution x = new Solution();

x.removeLoop(head);

if( detectLoop(head) || length(head)!=n )

System.out.println("0");

else

System.out.println("1");

}

}

}

// } Driver Code Ends

class Solution

{

public static void removeLoop(Node head){

Node slow = head;

Node fast = head;

while(fast!=null && fast.next!=null){

slow = slow.next;

fast = fast.next.next;

if(slow == fast){

break;

}

}

if(slow!=fast)

return ;

fast=head;

while(slow.next!=fast){

slow = slow.next;

fast = fast.next;

}

slow.next=null;

}

}