**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.**

**Example 3:**

**Input:**

**N = 4**

**value[] = {1,2,3,4}**

**X = 1**

**Output: 1**

**Explanation: The link list looks like**

**1 -> 2 -> 3 -> 4  
^ |**

**|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|**

**A loop is present.**

**If you remove it successfully,**

**the answer will be 1.**

**Your Task:  
You don't need to read input or print anything. Your task is to complete the function removeLoop() which takes the head of the linked list as the input parameter. Simply remove the loop in the list (if present) without disconnecting any nodes from the list.  
Note: The generated output will be 1 if your submitted code is correct.**

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

**Constraints:  
1 ≤ N ≤ 10^4**

**class Solution**

**{**

**public:**

**//Function to remove a loop in the linked list.**

**void removeLoop(Node\* head)**

**{**

**// code here**

**// just remove the loop without losing any nodes**

**if(head->next==NULL || head==NULL){**

**return;**

**}**

**bool flag=true;**

**Node\* slow=head;**

**Node\* fast=head;**

**while(slow!=fast || flag){**

**if(fast->next==NULL || fast->next->next==NULL){**

**return;**

**}**

**fast=fast->next->next;**

**slow=slow->next;**

**flag=false;**

**}**

**//if we start prev from head and shift both the pointers prev and slow one step ahead they will point to the start of the loop always**

**Node\* prev=head;**

**while(prev!=slow){**

**slow=slow->next;**

**prev=prev->next;**

**}**

**//prev for pointing to the end of the loop**

**prev=slow;**

**while(prev->next!=slow){**

**prev=prev->next;**

**}**

**prev->next=NULL;**

**}**

**};**

**Link :** [**https://www.geeksforgeeks.org/problems/remove-loop-in-linked-list/1?page=1&sortBy=submissions**](https://www.geeksforgeeks.org/problems/remove-loop-in-linked-list/1?page=1&sortBy=submissions)