```
// C++ program to detect and remove loop
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    int key;
    struct Node* next;
} Node;

Node* newNode(int key)
{
    Node* temp = (Node*)malloc(sizeof(Node));
    temp->key = key;
    temp->next = NULL;
    return temp;
}

// A utility function to print a linked list
void printList(Node* head)
{
    while (head != NULL) {
        printf("%d ", head->key);
        head = head->next;
    }
    printf("\n");
}

// Function to detect and remove loop in a linked list that
// may contain loop
void detectAndRemoveLoop(Node* head)
{
    // If list is empty or has only one node without loop
    if (head == NULL || head->next == NULL)
        return;

    Node *slow = head, *fast = head;

    // Move slow and fast 1 and 2 steps ahead respectively.
    slow = slow->next;
    fast = fast->next->next;

    // Search for loop using slow and fast pointers
    while (fast && fast->next) {
        if (slow == fast)
            break;
        slow = slow->next;
        fast = fast->next->next;
    }

    /* If loop exists */
    if (slow == fast) {
        slow = head;

        // this check is needed when slow and fast both meet
        // at the head of the LL eg: 1->2->3->4->5 and then
        // 5->next = 1 i.e the head of the LL
        if (slow == fast)
            while (fast->next != slow)
                fast = fast->next;
        else {
            while (slow->next != fast->next) {
                slow = slow->next;
                fast = fast->next;
            }
        }

        /* since fast->next is the looping point */
        fast->next = NULL; /* remove loop */
    }
}

/* Driver program to test above function*/
int main()
{
    Node* head = newNode(50);
    head->next = head;
    head->next = newNode(20);
    head->next->next = newNode(15);
```

```c
    head->next->next->next = newNode(4);
    head->next->next->next->next = newNode(10);

    /* Create a loop for testing */
    head->next->next->next->next->next = head;

    detectAndRemoveLoop(head);

    printf("Linked List after removing loop \n");
    printList(head);

    return 0;
}

// This code is contributed by Aditya Kumar (adityakumar129)

```