```java
// Function to remove the loop from the linked list. It takes the head of the
// linked list as input parameter.
// Then it simply remove the loop in the list (if present) without disconnecting
// any nodes from the list.
// Node(int data, Node next)
public static void removeLoop(Node head) {
    // list is empty or list has only one node -> no loop.
    if (head == null || head.next == null) {
        return;
    }
    // Floyd's loop detection algorithm
    Node low = head.next;
    Node high = head.next.next;

    while (low != high) {
        // loop doesn't exist
        if (high == null || high.next == null) {
            return;
        }
        low = low.next;
        high = high.next.next;
    }
    // detect starting node of the loop
    Node start = head;
    while (start != high) {
        start = start.next;
        high = high.next;
    }
    while (true) {
        // remove the next pointer of last node which pointing to the starting node of
        // the loop.
        if (high.next == start) {
            high.next = null;
            break;
        }
        high = high.next;
    }
}
```