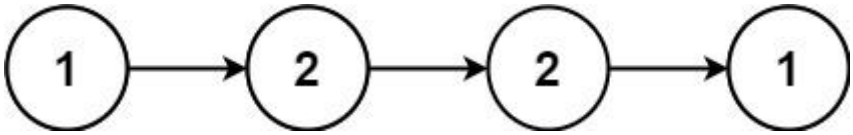
Given the head of a singly linked list, return true *if it is a*

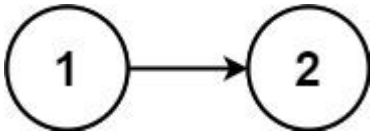
palindrome
***or* false *otherwise*.**

Example 1:



Input: head = [1,2,2,1]
Output: true

Example 2:



Input: head = [1,2]
Output: false
Constraints:

- **The number of nodes in the list is in the range [1, 10^5].**
- **0 <= Node.val <= 9**

Follow up: Could you do it in O(n) time and O(1) space?

Solution:

```
class Solution {
    public ListNode midnode(ListNode head){
        ListNode s=head;
        ListNode f=head;
        while(f!=null&&f.next!=null){
            s=s.next;
            f=f.next.next;
        }
        return s;
    }
    public boolean isPalindrome(ListNode head) {
        if(head==null||head.next==null){
            return true;
        }
        ListNode m= midnode(head);
        ListNode curr=m;
        ListNode prev=null;
        ListNode n;
        while(curr!=null){
            n=curr.next;
            curr.next=prev;
            prev=curr;
            curr=n;
        }
        ListNode r=prev;
        ListNode l=head;
        while(r!=null){
            if(l.val!=r.val){
                return false;
            }
            l=l.next;
            r=r.next;
        }
        return true;
    }
}
```