Reverse a Linked list

Maintain 3 pointers prev, head and next. head→next should point to prev for reversing the linkedlist.

temp always points to the next element of head so till head is not null temp=head→next. Then we move the prev, and head by one location assigning prev=head ; and head=temp.

We know that as the loop breaks head points to NULL but prev points to first elem of reversed list so we return prev.

```
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        if(head==NULL || head->next==NULL)
            return head;
        ListNode* prev=NULL;
        while(head!=NULL)
        {
            ListNode* temp=head->next;
            head->next=prev;
            prev=head;
            head=temp;
        }
        return prev;
    }
};
```

- Time Complexity : O(N) only traversed once.
- Space Complexity : O(1)