Reverse a Singly Linked List

Given a singly linked list, return the linked list in reversed  
  
Input: Linked List  
Output: Linked List

# Example

Input: {1} -> {2} -> {3} -> {4}

Output: {4} -> {3} -> {2} -> {1}

# Constraints

Time Complexity: O(N)  
Auxiliary Space Complexity: O(1)

* The linked list has the following properties
  + head : pointer to the head node
* Each node in the linked list has the following properties
  + next: pointer to the next node in the linked list, the default would be null
  + value: integer value of the node
* Values of the nodes will be integers

# Solution

1. Instantiate three node pointers
   1. ‘prev’ set as null
   2. ‘current’ set as the head node
   3. ‘next’ set as the node after head
2. While ‘current’ has not reached null
   1. Set ‘current’ next pointer to ‘prev’
   2. Then shift the three values forward by
      1. Set ‘prev’ to ‘current’
      2. Set ‘current’ to ‘next’
      3. As long as next is not null, set to next.next
3. Once outside the while loop, set the head to ‘prev’
4. Return the linked list

# Notes

There is a way to use recursion however, the call stack will use O(N) auxiliary space.

Must ensure the head property is updated.

# Resources

http://www.geeksforgeeks.org/write-a-function-to-reverse-the-nodes-of-a-linked-list/