Level: Easy

Reverse a linked list. You should use O(1) space.

Questions to Clarify:

Q. Is is a singly linked list?
A. Yes

Q. How do you want the output?
A. Return the new head of the linked list.

Solution:

This question is banned at most companies. It was very common at one point, which is why no one asks it. However, it is still good to know the trick.

We simply rearrange the next pointers to do the reversal.

Pseudocode:

(Note: Never write pseudocode in an actual interview. Unless you're writing a few lines quickly to plan out your solution. Your actual solution should be in a real language and use good syntax.)

```
prev = null
curr = head
while curr != null
     Node next = curr.next
     curr.next = prev
     prev = curr
     curr = next

prev is the new head, return prev
```

Test Cases:

Edge Cases: Empty list
Base Cases: Single node, 2 nodes
Regular Cases: > 2 nodes

Time Complexity: O(n)

Space Complexity: O(1)

```
public static Node reverse(Node head) {
    Node prev = null, curr = head;
    while (curr != null) {
        Node next = curr.getNext();
        curr.setNext(prev);
        prev = curr;
        curr = next;
    }
    return prev;
}
```