Reverse Operations

You are given a singly-linked list that contains **N** integers. A *subpart* of the list is a contiguous set of even elements, bordered either by either end of the list or an odd element. For example, if the list is [1, 2, 8, 9, 12, 16], the *subparts* of the list are [2, 8] and [12, 16].

Then, for each subpart, the order of the elements is reversed. In the example, this would result in the new list, [1, 8, 2, 9, 16, 12].

The goal of this question is: given a resulting list, determine the original order of the elements.

Implementation detail:

You must use the following definition for elements in the linked list:

```
class Node {
    int data;
    Node next;
}
```

Signature
Node reverse(Node head)

Constraints
1 <= **N** <= 1000, where **N** is the size of the list
1 <= **Li** <= 10^9, where **Li** is the ith element of the list

Example
Input:
N = 6
list = [1, 2, 8, 9, 12, 16]

Output:
[1, 8, 2, 9, 16, 12]