**Question:** You are given a non-dummy-headed singly linked list. Write a recursive function **multiply\_middle(head)** that finds the middle element of the linked list. If the list has an even number of elements, return the second of the two middle elements. Afterwards, all the values in the list are multiplied by this middle element.   
**Note:** The function should not use any loops. You may use any number of helper functions as needed.

| **Sample Input (list)** | **Sample Output (modified list)** | **Explanation** |
| --- | --- | --- |
| 1→2→3→4→5→6→7→8 | 5 → 10 → 15 → 20 → 25 → 30 → 35 → 40 | The middle element is 5 (second of two middles: 4 and 5), so all nodes are multiplied by 5. |
| 1→2→3→4→5 | 3→6→9→12→15 | The middle element is 3, so all nodes are multiplied by 3. |

**Question:** You are given a non-dummy-headed singly linked list. Write a recursive function **divide\_middle(head)** that finds the middle element of the linked list. If the list has an even number of elements, return the second of the two middle elements. Afterwards, all the values in the list are divided by this middle element.   
**Note:** The function should not use any loops. You may use any number of helper functions as needed.

| **Sample Input (list)** | **Sample Output (modified list)** | **Explanation** |
| --- | --- | --- |
| 10 → 20 → 30 → 40 | 0 → 0 → 1 → 1 | The middle element is 30 (second of two middles: 20 and 30), so all nodes are divided by 30 using integer division. |
| 36 → 38 → 2 → 34 → 40 | 18 → 19 → 1 → 17 → 20 | The middle element is 2, so all nodes are divided by 2 using integer division. |