**Question:** Write a recursive method/function called sumProdAlternate( ) that takes the head of a Singly LinkedList in its parameter and returns the **difference between the sum of the elements at even positions and the product of the elements at odd positions**, starting from the head.

**Note:** The head is considered to be at position 0. You can use as many helper methods/functions. No need to write the Node class.

| **Sample Input** | **Sample Output** |
| --- | --- |
| 2 → 3 → 4 → 5 → 6 | -3 |
| 10 → 1 → 2 | 11 |

**Question:** Write a recursive method/function called weightedSum( ) that takes the head of a Singly Linked List as its parameter and returns the final weighted sum of the list elements.

* For every node at an even position from the head (0th, 2nd, 4th, etc.), multiply the value by 2 and add it to the result.
* For every node at an odd position from the head (1st, 3rd, 5th, etc.), multiply the value by –1 and add it to the result.

**Note:** The head is considered to be at position 0.You can use as many helper methods/functions. No need to write the Node class.

| **Sample Input** | **Sample Output** |
| --- | --- |
| 4 → 3 → 5 → 2 | 13 |
| 1 → 6 → 2 → 7 → 3 | -1 |