Problem 1290: Convert Binary Number in a Linked List to Integer

Problem Information

Difficulty: Easy
Acceptance Rate: 82.31%
Paid Only: No
Tags: Linked List, Math

Problem Description

Given `head` which is a reference node to a singly-linked list. The value of each node in the linked list is either `0` or `1`. The linked list holds the binary representation of a number.

Return the _decimal value_ of the number in the linked list.

The **most significant bit** is at the head of the linked list.

Example 1:

![](https://assets.leetcode.com/uploads/2019/12/05/graph-1.png)

Input: head = [1,0,1] **Output:** 5 **Explanation:** (101) in base 2 = (5) in base 10

Example 2:

Input: head = [0] **Output:** 0

Constraints:

* The Linked List is not empty. * Number of nodes will not exceed `30`. * Each node's value is either `0` or `1`.

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
int getDecimalValue(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public int getDecimalValue(ListNode head) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
```

```
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def getDecimalValue(self, head: Optional[ListNode]) -> int:
```