**📌 Challenge Name:**

**InstaReel Feed Designer**

**📖 Description:**

Design a system to manage user reels, follow relationships, and retrieve the 10 most recent personalized reel feed.

**🧩 Problem Statement:**

You are tasked with designing a simplified social media reel system. Users can post short videos (reels), follow or unfollow other users, and retrieve a personalized news feed consisting of the 10 most recent reel IDs from themselves and the users they follow.

Implement the following class:

class Instagram:

def \_\_init\_\_(self): pass

def postReel(self, userId: int, reelId: int) -> None: pass

def getNewsFeed(self, userId: int) -> List[int]: pass

def follow(self, followerId: int, followeeId: int) -> None: pass

def unfollow(self, followerId: int, followeeId: int) -> None: pass

**🔽 Input Format:**

A sequence of function calls to the Instagram class.

Each call may be one of: "postReel", "getNewsFeed", "follow", "unfollow".

Arguments will be provided as arrays.

**📈 Constraints:**

1 <= userId, reelId <= 10^4

No duplicate reelIds will be posted.

At most 10^4 function calls.

follow and unfollow will not involve self-following.

User follows/unfollows only valid user IDs.

**📤 Output Format:**

For every getNewsFeed call, output a list of up to 10 reel IDs ordered from most recent to least recent.

**🏷️ Tags:**

design, hashmap, linked-list, heap, data structures, social media, reel feed, simulation

**🔍 Test Cases:**

**Test Case 1**

**Input**

["Instagram", "postReel", "getNewsFeed"]

[[], [1, 1001], [1]]

**Output**

[null, null, [1001]]

**Test Case 2**

**Input**

["Instagram", "postReel", "postReel", "follow", "postReel", "getNewsFeed"]

[[], [1, 1002], [2, 1003], [1, 2], [2, 1004], [1]]

**Output**

[null, null, null, null, null, [1004, 1003, 1002]]

**Test Case 3**

**Input**

["Instagram", "postReel", "follow", "postReel", "unfollow", "getNewsFeed"]

[[], [3, 1010], [3, 4], [4, 1011], [3, 4], [3]]

**Output**

[null, null, null, null, null, [1010]]

**Test Case 4**

**Input**

["Instagram", "postReel", "postReel", "postReel", "follow", "getNewsFeed"]

[[], [5, 2001], [6, 2002], [5, 2003], [6, 5], [6]]

**Output**

[null, null, null, null, null, [2003, 2002, 2001]]

**Test Case 5**

**Input**

["Instagram", "getNewsFeed", "postReel", "getNewsFeed"]

[[], [7], [7, 3001], [7]]

**Output**

[null, [], null, [3001]]

**📌 Challenge Name:**

**Sum of Two Reversed Numbers**

**✏️ Description:**

Add two numbers represented by reversed linked lists and return the sum as a reversed linked list.

**📖 Problem Statement:**

You are given two non-empty singly linked lists, each representing a non-negative integer. The digits are stored in reverse order, and each node contains a single digit. Your task is to add the two numbers and return the resulting sum as a linked list in reverse order.

You must implement the following function:

def addTwoNumbers(l1: ListNode, l2: ListNode) -> ListNode:

# return the head of the linked list representing the sum

**🔽 Input Format:**

Two linked lists l1 and l2, where each node contains a single digit.

Input values will be provided as arrays representing the linked list contents in reverse order.

**📈 Constraints:**

1 <= len(l1), len(l2) <= 100

0 <= Node.val <= 9

No leading zeros, unless the number is exactly 0.

**📤 Output Format:**

Return the head of a new linked list representing the sum of the numbers, also in reverse order.

**🏷️ Tags:**

linked list, math, simulation, data structures, digit-wise addition, carry tracking

**🧪 Sample Test Cases:**

**✅ Test Case 1**

**Input**

l1 = [2, 4, 3]

l2 = [5, 6, 4]

**Output**

[7, 0, 8]

**Explanation**  
342 + 465 = 807 → [7, 0, 8]

**✅ Test Case 2**

**Input**

l1 = [0]

l2 = [0]

**Output**

[0]

**Explanation**  
0 + 0 = 0

**✅ Test Case 3**

**Input**

l1 = [9,9,9,9,9,9,9]

l2 = [9,9,9,9]

**Output**

[8,9,9,9,0,0,0,1]

**Explanation**  
9999999 + 9999 = 10009998

**✅ Test Case 4**

**Input**

l1 = [1, 8]

l2 = [0]

**Output**

[1, 8]

**Explanation**  
81 + 0 = 81

**✅ Test Case 5**

**Input**

l1 = [5]

l2 = [5]

**Output**

[0, 1]

**Explanation**  
5 + 5 = 10

**📌 Challenge Name:**

**Binary List to Decimal Conversion**

**✏️ Description (140 characters):**

Convert a binary number stored in a linked list (MSB at head) into its decimal equivalent using efficient traversal.

**📖 Problem Statement:**

You are given a non-empty singly linked list representing a binary number. Each node contains either a 0 or 1, and the list represents the number such that the most significant bit (MSB) is at the head of the list. Your task is to compute and return the decimal (base-10) value of the binary number.

Implement the following function:

def getDecimalValue(head: ListNode) -> int:

# return the decimal value of the binary number

**🔽 Input Format:**

The input consists of a list of integers [b1, b2, ..., bn], representing node values from head to tail in a linked list.

Each integer is either 0 or 1.

**📈 Constraints:**

1 <= n <= 30 — where n is the number of nodes

Each node’s value is either 0 or 1

The linked list is not empty

**📤 Output Format:**

Print a single integer — the decimal equivalent of the binary number represented in the list.

**🏷️ Tags:**

linked list, bit manipulation, binary, data structures, conversion, math, simulation

**🧪 Test Cases**

**✅ Test Case 1**

**Input**

head = [1, 0, 1]

**Output**

5

**Explanation**  
Binary 101 = Decimal 5

**✅ Test Case 2**

**Input**

head = [0]

**Output**

0

**✅ Test Case 3**

**Input**

head = [1, 1, 1, 1]

**Output**

15

**Explanation**  
Binary 1111 = Decimal 15

**✅ Test Case 4**

**Input**

head = [1, 0, 0, 0, 1]

**Output**

17

**Explanation**  
Binary 10001 = Decimal 17

**✅ Test Case 5**

**Input**

head = [1] \* 30

**Output**

1073741823

**Explanation**  
Binary of 30 ones = ( 2^{30} - 1 )