Name:T.Harshitha

Date:05/01/2023

Python Codes

Problem Statement 1

 Add Two Numbers

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

**Example 1:**



**Input:** l1 = [2,4,3], l2 = [5,6,4]

**Output:** [7,0,8]

**Explanation:** 342 + 465 = 807.

**Example 2:**

**Input:** l1 = [0], l2 = [0]

**Output:** [0]

**Example 3:**

**Input:** l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9]

SOURCE CODE:

def addTwoNumbers( l1,l2):

        dummy=ListNode(0)

        curr=dummy

        carry=0

        while l1 or l2 or carry:

            v1=l1.val if l1 else 0

            v2=l2.val if l2 else 0

            val=v1+v2+carry

            carry=val//10

            val=val%10

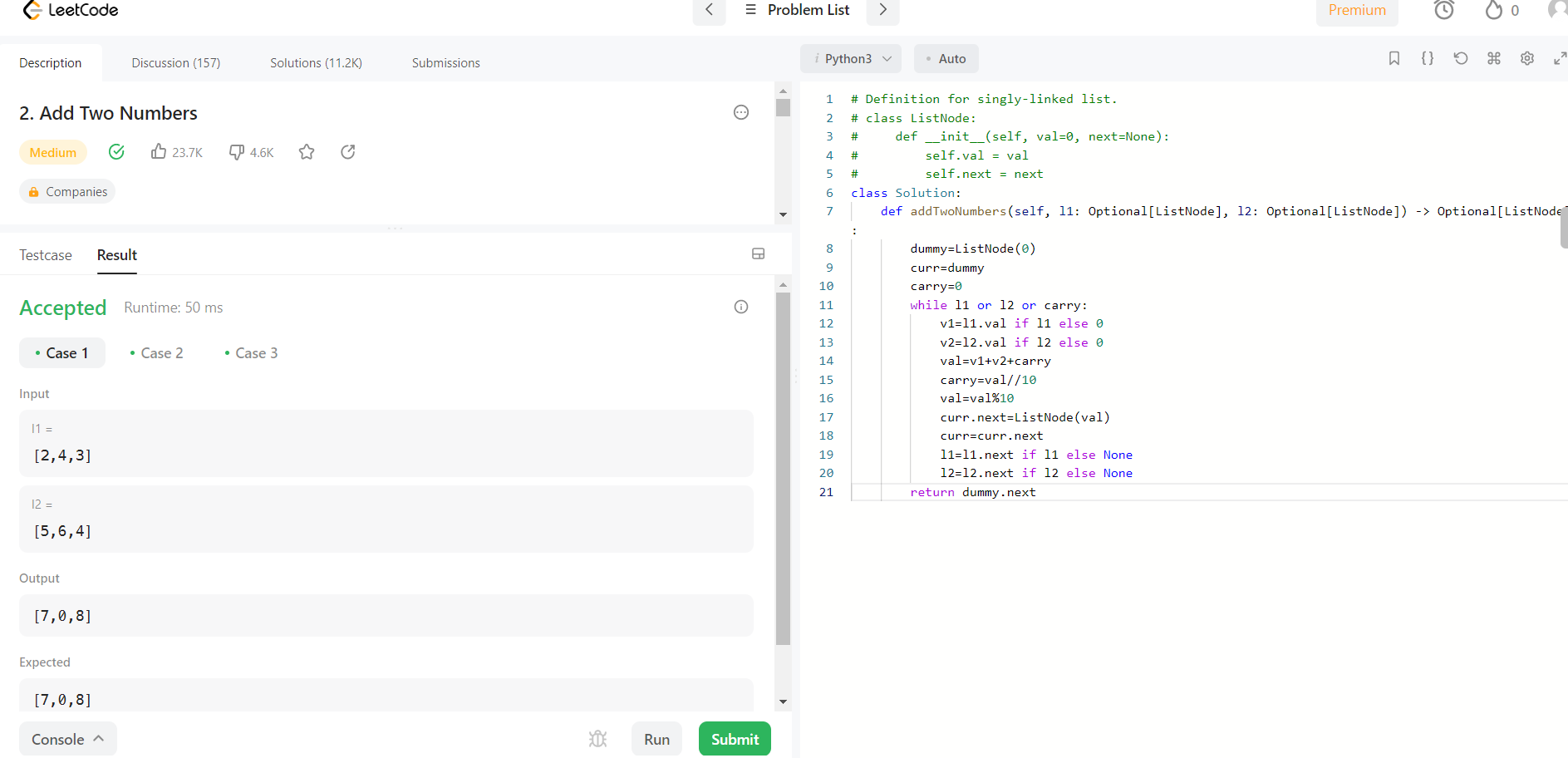
            curr.next=ListNode(val)

            curr=curr.next

            l1=l1.next if l1 else None

            l2=l2.next if l2 else None

        return dummy.next



PROBLEM STATEMENT 2

Valid Anagram

Given two strings s and t, return true if t is an anagram of s, and false otherwise.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

**Example 1:**

**Input:** s = "anagram", t = "nagaram"

**Output:** true

**Example 2:**

**Input:** s = "rat", t = "car"

**Output:** false

SOURCE CODE:

def isAnagram(self, s: str, t: str) -> bool:

        for i in s:

            if i not in t:

                return False

        return True

