# Definition for singly-linked list.

# class ListNode:

#     def \_\_init\_\_(self, val=0, next=None):

#         self.val = val

#         self.next = next

class Solution:

    def addTwoNumbers(self, l1: Optional[ListNode], l2: Optional[ListNode]) -> Optional[ListNode]:

        dummy = ListNode(0)

        temp = dummy

        carry = 0

        while l1 or l2 or carry != 0:

            val1 = l1.val if l1 else 0

            val2 = l2.val if l2 else 0

            num = val1 + val2 + carry

            val = num % 10

            carry = num // 10

            temp.next = ListNode(val)

            temp = temp.next

            l1 = l1.next if l1 else None

            l2 = l2.next if l2 else None

        return dummy.next

class Solution:

    def lengthOfLongestSubstring(self, s: str) -> int:

        max\_len = 0

        i = 0

        while(i < len(s)):

            j = i

            while(j < len(s)):

                sub\_str = s[i : j + 1]

                if(len(sub\_str) == len(set(sub\_str))):

                    if(len(sub\_str) > max\_len):

                        max\_len = len(sub\_str)

                else:

                    break

                j += 1

            i += 1

        return max\_len