**Question 1**

Given two strings s and t, determine if they are isomorphic.

Two strings s and t are isomorphic if the characters in s can be replaced to get t.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

**Example 1:**

**Input:** s = "egg", t = "add"

**Output:** true

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

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

        d={}

        if len(s)!=len(t):

            return False

        else :

            for i in range(len(s)):

                if s[i] in d or t[i] in d.values():

                    continue

                else:

                    d[s[i]]=t[i]

            for i in range(len(s)):

                if s[i] in d:

                    if t[i]!=d[s[i]]:

                        return False

                else:return False

            return True

**Question 2**

Given a string num which represents an integer, return true if num is a ***strobogrammatic number***.

A **strobogrammatic number** is a number that looks the same when rotated 180 degrees (looked at upside down).

**Example 1:**

**Input:** num = "69"

**Output:**

true

**Ans.**

def isStrobogrammatic(n):

    check=['0','1','8']

    check2=['1','0','8','9','6']

    # Write your code here.

    if len(n)==1:

        if n in check:

            return True

        else:

            return False

    start=0

    end=len(n)-1

    while start<end+1:

        if (n[start] not in check2) or (n[end] not in check2):

            return False

        elif n[start]=='6' and n[end] != '9':

            return False

        elif n[start]=='9' and n[end] != '6':

            return False

        elif n[start]=='0' and n[end] != '0':

            return False

        elif n[start]=='1' and n[end] != '1':

            return False

        elif n[start]=='8' and n[end] != '8':

            return False

        start+=1

        end-=1

    return True

**Question 3**

Given two non-negative integers, num1 and num2 represented as string, return the sum of num1 and num2 as a string.

You must solve the problem without using any built-in library for handling large integers (such as BigInteger). You must also not convert the inputs to integers directly.

**Example 1:**

**Input:** num1 = "11", num2 = "123"

**Output:**

"134"

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

    def addStrings(self, num1: str, num2: str) -> str:

        num1 = num1[::-1]

        num2 = num2[::-1]

        # Pad the shorter input string with zeros

        if len(num1) < len(num2):

            num1 += '0' \* (len(num2) - len(num1))

        else:

            num2 += '0' \* (len(num1) - len(num2))

        result = ''

        carry = 0

        # Loop through both strings, starting from the least significant digits

        for i in range(len(num1)):

            # Convert the current digit in each string to an integer and add them together

            digit\_sum = int(num1[i]) + int(num2[i]) + carry

            # Determine the carry for the next iteration, if any

            carry = digit\_sum // 10

            # Append the least significant digit of the sum to the result string

            result += str(digit\_sum % 10)

        # If there is a carry after the last iteration, append it to the result string

        if carry > 0:

            result += str(carry)

        # Reverse the result string to obtain the final sum

        return result[::-1]

**Question 4**

Given a string s, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

**Example 1:**

**Input:** s = "Let's take LeetCode contest"

**Output:** "s'teL ekat edoCteeL tsetnoc"

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

    def reverseWords(self, s: str) -> str:

        if ' ' not in s:

            return s[::-1]

        srt=''

        x=''

        for i in range(len(s)):

            if s[i] ==" ":

                x=x+srt[::-1]+' '

                srt=''

            elif i==len(s)-1:

                srt=srt+s[i]

                x=x+srt[::-1]

            else:

                srt=srt+s[i]

        return x[:]

**Question 5**

Given a string s and an integer k, reverse the first k characters for every 2k characters counting from the start of the string.

If there are fewer than k characters left, reverse all of them. If there are less than 2k but greater than or equal to k characters, then reverse the first k characters and leave the other as original.

**Example 1:**

**Input:** s = "abcdefg", k = 2

**Output:**

"bacdfeg"

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

    def reverseStr(self, s: str, k: int) -> str:

        s = list(s)

        for i in range(0, len(s), 2\*k):

            print(i, i+k)

            s[i:i+k] = reversed(s[i:i+k])

        return ''.join(s)

**Question 6**

Given two strings s and goal, return true *if and only if* s *can become* goal *after some number of* ***shifts*** *on* s.

A **shift** on s consists of moving the leftmost character of s to the rightmost position.

* For example, if s = "abcde", then it will be "bcdea" after one shift.

**Example 1:**

**Input:** s = "abcde", goal = "cdeab"

**Output:**

true

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

    def rotateString(self, s: str, goal: str) -> bool:

        if len(s)!=len(goal):

            return False

        if len(s)>0 and len(goal)>0:

            new\_s = s+s

            return (goal in new\_s)

        else:

            return False

**Question 7**

Given two strings s and t, return true if they are equal when both are typed into empty text editors. '#' means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

**Example 1:**

**Input:** s = "ab#c", t = "ad#c"

**Output:** true

**Explanation:**

Both s and t become "ac".

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

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

        stack1=[]

        stack2=[]

        for i in range(len(s)):

            try:

                if s[i]=="#":

                    stack1.pop()

                else:

                    stack1.append(s[i])

            except:

                continue

        for i in range(len(t)):

            try:

                if t[i]=="#":

                    stack2.pop()

                else:

                    stack2.append(t[i])

            except:

                continue

        return True if stack1==stack2 else False

**Question 8**

You are given an array coordinates, coordinates[i] = [x, y], where [x, y] represents the coordinate of a point. Check if these points make a straight line in the XY plane.

**Example 1:**

**Input:** coordinates = [[1,2],[2,3],[3,4],[4,5],[5,6],[6,7]]

**Output:** true

**Ans.** Solution from my leetcode : - <https://leetcode.com/gopsa2001/>

class Solution:

    def checkStraightLine(self, coordinates: List[List[int]]) -> bool:

        n = len(coordinates)

        x1, y1 = coordinates[0]

        x2, y2 = coordinates[1]

        for i in range(2, n):

            x, y = coordinates[i]

            if (y - y1) \* (x - x2) != (y - y2) \* (x - x1):

                return False

        return True