ASSIGMNENT -7

STRINGS

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

SOLUTION:

TC: O(n), SC:O(1)

CODE:

```
class Solution:
    def isIsomorphic(self, s: str, t: str) -> bool:
        return len(set(s))==len(set(t))==len(set(zip(s,t)))
```

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

SOLUTIONS:

TC:O(n), SC:O(1)

CODE:

```
class Solution(object):
    def isStrobogrammatic(self, num):

        maps = {("0", "0"), ("1", "1"), ("6", "9"), ("8", "8"), ("9", "6")}
        i,j = 0, len(num) - 1
        while i <= j:
            if (num[i], num[j]) not in maps:
                return False
            i += 1
            j -= 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"

SOLUTION:

TC:O(n), SC:O(1)

CODE:

```
class Solution:
    def addStrings(self, num1: str, num2: str) -> str:
        sys.set_int_max_str_digits(10000)
        n=int(num1)
        n1=int(num2)
        n2=n+n1
        return str(n2)
```

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"

SOLUTION:

TC:O(n), SC:O(1)

CODE:

```
class Solution:
    def reverseWords(self, s: str) -> str:
        words=s.split()
        ans = ""
        for  i in range(len(words)):
            ans += words[i][::-1]
            if i != len(words)-1:
                ans+=" "
        return ans
```

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"

SOLUTION:

TC:O(n), SC:O(n)

CODE:

```
class Solution:
    def reverseStr(self, s: str, k: int) -> str:
        m = ''
        for i in range(0,len(s),2*k):
                m+= (s[i:i+k][::-1]) + s[i+k:i+2*k]
        return m
```

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

SOLUTION:

TC:O(n), SC:O(n)

CODE:

```
class Solution:
    def rotateString(self, s: str, goal: str) -> bool:
        return len(s) == len(goal) and s in goal+goal
```

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".

SOLUTION:

TC: O(n), SC: O(1)

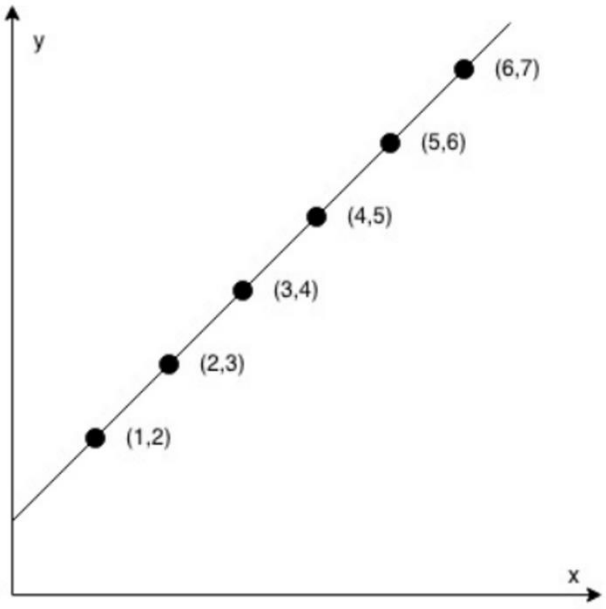
CODE:

```
class Solution:
       def backspaceCompare(self, S, T):
        back = lambda res, c: res[:-1] if c == '#' else res + c
        return reduce(back, S, "") == reduce(back, T, "")
```

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:



SOLUTION:

TC: O(n), SC: O(1)

CODE:

```
class Solution:
    def checkStraightLine(self, coordinates: List[List[int]]) -> bool:
        (x1, y1), (x2, y2) = coordinates[:2]
        for i in range(2, len(coordinates)):
            (x, y) = coordinates[i]
            if((y2 - y1) * (x1 - x) != (y1 - y) * (x2 - x1)):
                return False
        return True
```