Q1

class Solution:

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

if len(s) != len(t):

return False

s\_to\_t = {}

t\_to\_s = {}

for i in range(len(s)):

if s[i] in s\_to\_t:

if s\_to\_t[s[i]] != t[i]:

return False

else:

if t[i] in t\_to\_s:

return False

s\_to\_t[s[i]] = t[i]

t\_to\_s[t[i]] = s[i]

return True

Q2

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

Q3

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)

Q4

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

Q5

class Solution:

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

temp = []

while s:

temp.append(s[:k])

s = s[k:]

for i in range(0, len(temp), 2):

temp[i] = temp[i][::-1]

return ''.join(temp)

Q6

class Solution:

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

if s == goal:

return True

s, goal = [\*s], [\*goal]

for x in range(len(s)):

a = s[0]

s.pop(0); s.append(a)

if s == goal:

return True

return False

Q7

class Solution:

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

x=[]

y=[]

for i in range(len(s)):

if x and s[i]=="#":

x.pop()

else:

if s[i]!="#":

x.append(s[i])

for i in range(len(t)):

if y and t[i]=="#":

y.pop()

else:

if t[i]!="#":

y.append(t[i])

if x==y:

return True

return False

Q8

class Solution:

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

x1,y1=coordinates[0][0],coordinates[0][1]

x2,y2=coordinates[1][0],coordinates[1][1]

for x3,y3 in coordinates[2:]:

if (y3-y2)\*(x2-x1)!=(x3-x2)\*(y2-y1):

return False

else:

return True