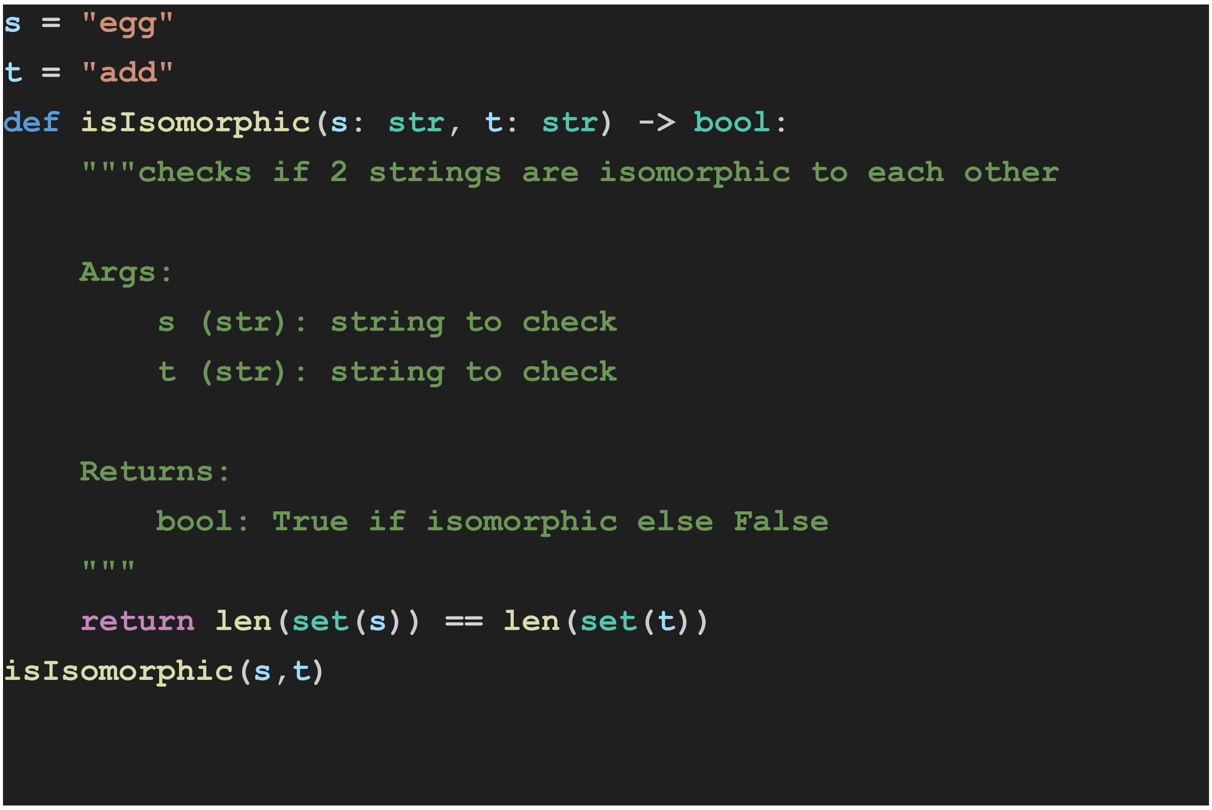
Assignment-7

Ans1-

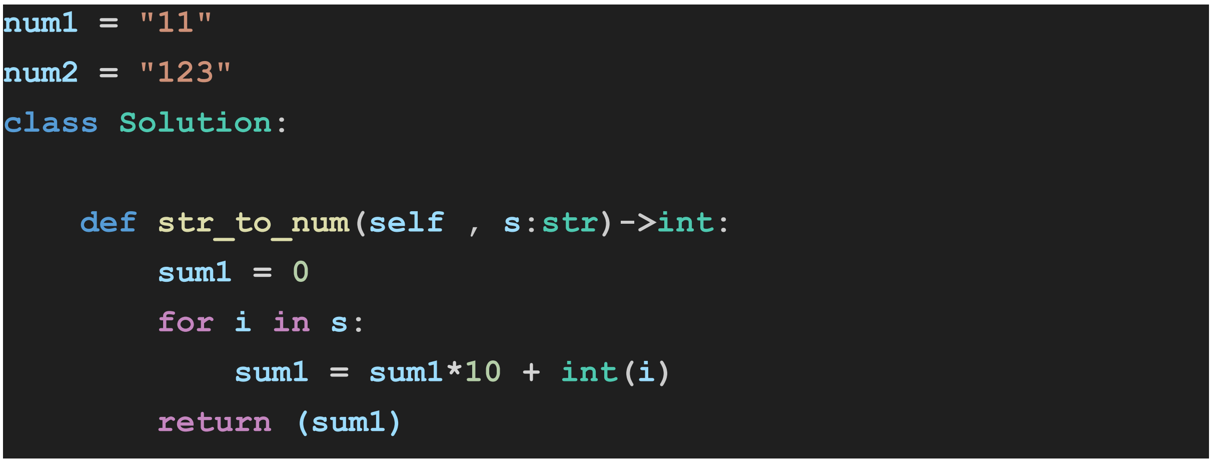


```
s = "egg"
t = "add"
def isIsomorphic(s: str, t: str) -> bool:
    """checks if 2 strings are isomorphic to each other

    Args:
        s (str): string to check
        t (str): string to check

    Returns:
        bool: True if isomorphic else False
    """
    return len(set(s)) == len(set(t))
isIsomorphic(s,t)
```

Ans3-



```
num1 = "11"
num2 = "123"
class Solution:

    def str_to_num(self , s:str)->int:
        sum1 = 0
        for i in s:
            sum1 = sum1*10 + int(i)
        return (sum1)
    def addStrings(self , num1: str, num2: str) -> str:
        """Takes 2 non negative numbers as strings and adds
        them together and returns a string representation of
the result

        Args:
            num1 (str): string representation of the first
number
            num2 (str): string representation of the second
number

        Returns:
            str: string representation of the sum
        """
        a = self.str_to_num(num1)

        b = self.str_to_num(num2)

        return str(a+b)

obj = Solution()
obj.addStrings(num1 , num2)
```

Ans 4-

```
s = "Let's take LeetCode contest"

def stringWordReversal(s:str)->str:
    """takes a string s and returns it
    with all the words reversed
    Args:
        s (str): string

    Returns:
        str: string with all the words reversed
    """
    return " ".join([i[::-1] for i in s.split()])
stringWordReversal(s)
```

Ans 5-

```
s = "abcdefg"
k = 2
def reverseStr(s, k):
        a = list(s)
        for i in range(0, len(a), 2*k):
            a[i:i+k] = reversed(a[i:i+k])
        return "".join(a)
reverseStr(s,k)
```

Ans 6-

```
s = "abcde"
goal = "cdeab"
def strShift(s:str, goal:str)->bool:
    """checks if a given string s can become
    string goal after some number of shifts

    Args:
        s (str): string
        goal (str): goal string
    Returns:
        bool: True if s can become goal else False
    """
    for i in range(len(s)):
        s = s[-1]+s[:-1]
        if s == goal:
            return True
    return False
strShift(s,goal)
```

Ans7-

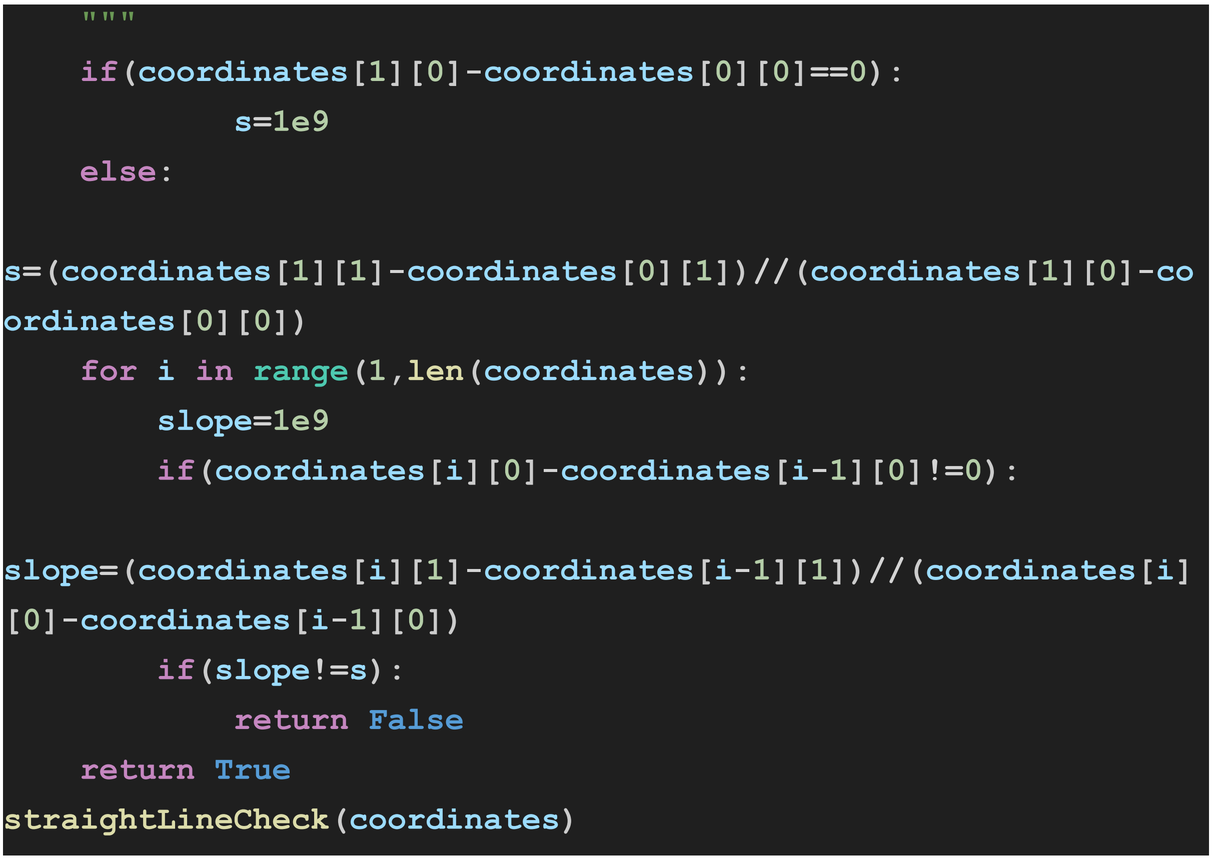
```
s = "ab#c"
t = "ad#c"
def backspaceCompare(S, T):
        def build(S):
            ans = []
            for c in S:
                if c != '#':
                    ans.append(c)
                elif ans:
                    ans.pop()
            return "".join(ans)
        return build(S) == build(T)
backspaceCompare(s,t)
```

Ans 8-

```
coordinates = [[1,2],[2,3],[3,4],[4,5],[5,6],[6,7]]
def straightLineCheck(coordinates: list[list[int]])->bool:
    """check if the given coordinates are on a straight line

    Args:
        coordinates (list[list[int]]): points

    Returns:
        bool: True if points are on a straight line
```



```
    """
    if(coordinates[1][0]-coordinates[0][0]==0):
            s=1e9
    else:

s=(coordinates[1][1]-coordinates[0][1])//(coordinates[1][0]-co
ordinates[0][0])
    for i in range(1,len(coordinates)):
        slope=1e9
        if(coordinates[i][0]-coordinates[i-1][0]!=0):

slope=(coordinates[i][1]-coordinates[i-1][1])//(coordinates[i]
[0]-coordinates[i-1][0])
        if(slope!=s):
            return False
    return True
straightLineCheck(coordinates)
```