M ASSIGNMENT 3

1. Odd String Difference

PROGRAM:

def findOddString(w):

def d(wrd):

return [ord(wrd[i+1]) - ord(wrd[i]) for i in range(len(wrd) - 1)]

diff\_arrays = [d(x) for x in w]

for i, diff in enumerate(diff\_arrays):

if diff\_arrays.count(diff) == 1:

return w[i]

print(findOddString(["adc", "wzy", "abc"]))

OUTPUT:



2. Words Within Two Edits of Dictionary

PROGRAM:

def two(q, d):

def f(a, b):

return sum(1 for x, y in zip(a, b) if x != y) <= 2

r = []

for a in q:

if any(f(a, b) for b in d):

r.append(a)

return r

print(two(["word", "note", "ants", "wood"], ["wood", "joke", "moat"]))

OUTPUT:



3.Next Greater Element IV

PROGRAM:

def targets(n, s):

from collections import defaultdict

c = defaultdict(int)

for x in n:

c[x % s] += 1

m = max(c.values())

v = float('inf')

for x in n:

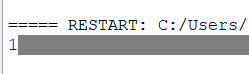
if c[x % s] == m:

v = min(v, x)

return v

print(targets([3, 7, 8, 1, 1, 5], 2))

OUTPUT:



4. Minimum Addition to Make Integer Beautiful

PROGRAM:

def minAdd(n, t):

def d(x):

return sum(int(c) for c in str(x))

x = 0

while d(n + x) > t:

x += 1

return x

print(minAdd(16, 6))

OUTPUT:



5. Sort Array by Moving Items to Empty Space

PROGRAM:

def minOperationsToSort(n):

l = len(n)

z = n.index(0)

o = 0

v = [False] \* l

def c(s):

k = 0

x = s

while not v[x]:

v[x] = True

x = n[x]

k += 1

return k - 1

for i in range(l):

if not v[i] and n[i] != i:

o += c(i)

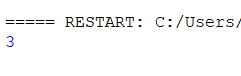
if z != 0 and z != l - 1:

o += 1

return o

print(minOperationsToSort([1, 0, 2, 4, 3]))

OUTPUT:



6. Apply Operations to an Array

PROGRAM:

def applyOperations(n):

l = len(n)

for i in range(l - 1):

if n[i] == n[i + 1]:

n[i] \*= 2

n[i + 1] = 0

r = [x for x in n if x != 0] + [0] \* n.count(0)

return r

print(applyOperations([1, 2, 2, 1, 1, 0]))

OUTPUT:



ANALYTICAL PROBLEMS

