Assignment:04

1.Odd String Difference You are given an array of equal-length strings words. Assume that the length of each string is n.

Sol:- words = ["hello", "world", "python", "code"]

n = len(words[0])

result = ["".join([words[i][j] for i in range(len(words)) if i % 2 == 1]) for j in range(n)]

print(result)

2. Words Within Two Edits of Dictionary You are given two string arrays, queries and dictionary. All words in each array comprise of lowercase English letters and have the same length.

Sol:- from collections import defaultdict

def is\_within\_two\_edits(word1, word2):

if word1 == word2:

return True

if len(word1) != len(word2):

return False

edits = 0

for c1, c2 in zip(word1, word2):

if c1 != c2:

edits += 1

if edits > 2:

return False

return True

def find\_words\_within\_two\_edits(queries, dictionary):

dict\_set = set(dictionary)

result = []

for query in queries:

if query in dict\_set:

result.append(query)

else:

for word in dictionary:

if is\_within\_two\_edits(query, word):

result.append(query)

break

return result

queries = ["word", "note", "ants", "wood"] 4. Minimum Addition to Make Integer Beautiful You are given two positive integers n and target. An integer is considered beautiful if the sum of its digits is less than or equal to target. Return the minimum non-negative integer x such that n + x is beautiful. The input will be generated such that it is always possible to make n beautiful.

dictionary = ["wood", "joke", "moat"]

output = find\_words\_within\_two\_edits(queries, dictionary)

print(output)

3. Next Greater Element IV You are given a 0-indexed array of non-negative integers nums. For each integer in nums, you must find its respective second greater integer

Sol:- def nextGreaterElement(nums):

stack, res = [], [-1] \* len(nums)

for i in range(2 \* len(nums)):

while stack and nums[stack[-1]] < nums[i % len(nums)]:

res[stack.pop()] = nums[i % len(nums)]

stack.append(i % len(nums))

return res

4. 4. Minimum Addition to Make Integer Beautiful You are given two positive integers n and target. An integer is considered beautiful if the sum of its digits is less than or equal to target. Return the minimum non-negative integer x such that n + x is beautiful. The input will be generated such that it is always possible to make n beautiful.

Sol:- def minAddToMakeBeautiful(n, target):

def sum\_of\_digits(num):

return sum(int(digit) for digit in str(num))

x = 0

while sum\_of\_digits(n + x) > target:

x += 1

return x

n = 16

target = 6

result = minAddToMakeBeautiful(n, target)

print(result)

5. Sort Array by Moving Items to Empty Space You are given an integer array nums of size n containing each element from 0 to n - 1 (inclusive). Each of the elements from 1 to n - 1 represents an item, and the element 0 represents an empty space.

Sol:- def sort\_array(nums):

n = len(nums)

for i in range(n):

while nums[i] != i:

if nums[i] == 0:

break

nums[nums[i]], nums[i] = nums[i], nums[nums[i]]

return nums

6.