LAB04

Given the file text.txt

Exercise 1

Write a Python function that takes two 2D arrays (matrices) as input and returns their sum.

Exercise 2

Write a Python function to find the sum of each row in a 2D array and return the results as a list.

Exercise 3

Write a Python function that takes a dictionary as input and returns a new dictionary where the keys and values are swapped.

Exercise 4

Write a Python function that finds the intersection of two sets.

Exercise 5

Create a function that returns the symmetric difference of two sets.

Exercise 6

Write a Python function to group words that are anagrams of each other using a dictionary.

Ex:

Input: words = ["eat", "tea", "tan", "ate", "nat", "bat"]

Output: [['eat', 'tea', 'ate'], ['tan', 'nat'], ['bat']]

Exercise 7

Write a function that merges two nested dictionaries. If keys overlap and values are dictionaries, merge them recursively.

Ex:

Input:

d1 = {'a': {'b': 1}, 'c': 3}

d2 = {'a': {'d': 4}, 'c': 5}

Output: {'a': {'b': 1, 'd': 4}, 'c': 5}

Exercise 8

Given a 2D array of integers, write a function that returns a dictionary where the keys are the unique integers and the values are their frequencies.

Input:

matrix =
$$\begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$
Output: {1: 3, 2: 3, 3: 2, 4: 1}

Exercise 9

Given two sets, write a Python function to find the elements present in the first set but not in the second.

Input:

set1 = 1, 2, 3, 4, 5

set2 = 2, 4, 6

Output: 1, 3, 5

Exercise 10

Write a function to find common elements among all rows of a 2D array, where each row is treated as a tuple.

Input: matrix =
$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & 2 & 4 \end{bmatrix}$$
Output:

{2}

Exercise 11

There is a language that uses 26 lowercase English letters, but the order of the letters may be different from the usual alphabetical order.

You are given:

1. A list of words written in this language.

2. A string representing the custom order of the alphabet.

Your task is to determine whether the given list of words is sorted according to this custom alphabetical order.

EX:

Input:

words = ["hello", "intelligent", "class"]

order = "hlabicdefgjkmnopqrstuvwxyz"

Output: True

Explanation:

According to the custom order, 'h' comes before 'l', 'l' comes before 'c' so "hello" comes before "intelligent" and "intelligent" comes before "class" in the given order.