Machine Learning and Data Science Module 02 (Exam)

Problem 1: Find Pairs with Sum

You are given a list of tuples, where each tuple contains two integers. Your task is to write a Python function, that takes in the list of tuples and a target sum. The function should return a list of tuples, each representing a pair of numbers whose sum is equal to the target sum. Sample Input: tuples_list = [(1, 2), (2, 3), (4, 5), (6, 7), (8, 9)] target_sum = 10 Output: [(1, 9), (2, 8)] Hint: The function should iterate through each tuple in the list and check if the sum of the two integers in the tuple equals the target sum. If a pair is found, it should be added to the result list. Note that a number can participate in multiple pairs.

Solution:

The goal is to find pairs of integers from a list of tuples that sum up to a given target. Here's a revised solution:

```
def find_pairs_with_sum(tuples_list, target_sum):
    result = []
    for num1, num2 in tuples_list:
        if num1 + num2 == target_sum:
            result.append((num1, num2))
    return result

# Sample Input
tuples_list = [(1, 2), (2, 3), (4, 5), (6, 7), (8, 9)]
target_sum = 10
# Output
print(find_pairs_with_sum(tuples_list, target_sum))  # Output: [(1, 9), (2, 8)]
```

Problem 2: Sum of Diagonal Elements

Write a Python program to input a square numpy array of shape (n, n), where n is a positive integer. Take the value of n from the user and input the user's array values. Then write a Python function, diagonal_sum, that takes in the numpy array and calculates the sum of the diagonal elements. Sample Input: [[1, 2, 3], [4, 5, 6], [7, 8, 9]] Output: 15 Hint: The diagonal elements are defined as the elements that lie on the main diagonal of the array, i.e., the elements with

Answer corrected by: Md. Iquball Hossain, NACTAR MLDS Bach-03, Roll -12 (Contact: 01777044834)

indices (i, i) for i ranging from 0 to n-1. Your function should iterate through the array using a loop and calculate the sum of the diagonal elements.

Solution:

To sum the diagonal elements of a square numpy array:

```
import numpy as np

def diagonal_sum(arr):
    n = arr.shape[0]
    diag_sum = 0
    for i in range(n):
        diag_sum += arr[i, i]
    return diag_sum

# Taking input from the user for the size of the array
n = int(input("Enter the size of the square array: "))
print(f"Enter {n}x{n} array elements:")

# Taking input for the array elements
user_array = []
for i in range(n):
    row = [int(x) for x in input().split()]
    user_array.append(row)

# Converting the user input list into a NumPy array
numpy_array = np.array(user_array)
result = diagonal_sum(numpy_array)
print("Sum of diagonal elements:", result)
```

Problem 3: Subtract Two Matrices

Write a Python program to subtract two matrices. Take two numpy arrays, arr1, and arr2, both of shape (m, n), where m and n are positive integers (you should input each m and n value for each matrix and all matrix elements). Your task is to write a Python function, calculate_subtraction, that performs element-wise subtraction between the two arrays and returns the resulting numpy array

Answer corrected by: Md. Iquball Hossain, NACTAR MLDS Bach-03, Roll -12 (Contact: 01777044834)

Solution:

For element-wise subtraction of two matrices:

```
import numpy as np
def calculate_subtraction(arr1, arr2):
    return arr1 - arr2
# Taking input for the dimensions of the matrices
m1 = int(input("Enter the number of rows for matrix 1: "))
n1 = int(input("Enter the number of columns for matrix 1: "))
m2 = int(input("Enter the number of rows for matrix 2: "))
n2 = int(input("Enter the number of columns for matrix 2: "))

if m1 != m2 or n1 != n2:
    print("Matrices should have the same shape for subtraction.")
else:
    # Taking input for elements of matrix 1
    print("Enter elements for matrix 1:")
    matrix1 = []
    for i in range(m1):
        row = [int(x) for x in input().split()]
        matrix1.append(row)

    # Taking input for elements of matrix 2
    print("Enter elements for matrix 2:")
    matrix2 = []
    for i in range(m2):
        row = [int(x) for x in input().split()]
        matrix2.append(row)

    # Converting the input lists into NumPy arrays
    arr1 = np.array(matrix1)
    arr2 = np.array(matrix2)

    # Calculating the subtraction
    result_array = calculate_subtraction(arr1, arr2)
    print("Result after subtraction:\n", result_array)
```

Answer corrected by: Md. Iquball Hossain, NACTAR MLDS Bach-03, Roll -12 (Contact: 01777044834)