**Exercise 4: Dating Game**

You are building a Dating Game application that matches people based on their interests. Write a Python program that models the process of matching users and provides information about their compatibility.

The program should include the following functionalities:

1. Define a list of dictionaries, where each dictionary represents a user with attributes such as 'name', 'age', 'gender', and 'interests'. Populate the list with at least 4 user profiles.

2. Define a function called `match\_users` that takes two user dictionaries as arguments and calculates their compatibility score based on common interests. The compatibility score is calculated by counting the number of common interests between the users and returning the score as a percentage.

3. Use a list comprehension to generate a list of tuples, where each tuple represents a pair of matched users along with their compatibility score. The list should include all possible pairs of users in the list.

4. Use the `sorted` function and a lambda function to sort the list of matched user tuples based on their compatibility score in descending order.

5. Define a function called `print\_matches` that takes the sorted list of matched user tuples as an argument and prints the user pairs along with their compatibility scores.

Here's a skeleton code to get you started:

| *# Step 1: Define the list of user profiles* users = [  {'name': 'Alice', 'age': 25, 'gender': 'female', 'interests': ['hiking', 'reading', 'cooking']},  {'name': 'Bob', 'age': 28, 'gender': 'male', 'interests': ['movies', 'sports', 'travel']},  {'name': 'Charlie', 'age': 23, 'gender': 'male', 'interests': ['music', 'cooking', 'photography']},  {'name': 'Diana', 'age': 26, 'gender': 'female', 'interests': ['reading', 'travel', 'painting']},  *# Add more user profiles* ]  *# Step 2: Define the function to match users* def match\_users(user1, user2):  *# Add your code*  return compatibility\_score  *# Step 3: Generate a list of matched user pairs with their compatibility scores* matched\_users = [] *# Add your code*  *# Step 4: Sort the matched user pairs based on compatibility score* sorted\_matches = *# Add your code*  *# Step 5: Print the matched user pairs and their compatibility scores* for user1, user2, score in sorted\_matches:  print(f"{user1} and {user2}: {score}% compatibility") |
| --- |

In this exercise, you will complete the code to implement the required functionalities. You will define the list of user profiles, implement the `match\_users` function to calculate compatibility scores, use a list comprehension to generate matched user pairs, sort the pairs based on compatibility scores, and finally print the matched user pairs.

Note: The compatibility score in this exercise is calculated as the percentage of common interests between two users. You can customize the matching algorithm