Assignment Questions

---

1. List Assignments

1. Write a Python program to input 5 numbers from the user and store them in a list. Then:
   - Print the list.
   - Calculate and display the sum and average of the numbers.
2. Write a Python program to create a list of fruits. Perform the following operations:
   - Add a new fruit to the list.
   - Remove a specific fruit (entered by the user).
   - Sort the list in alphabetical order.
3. Write a Python program to create a list of 10 numbers. Then:
   - Display the even and odd numbers separately.
   - Replace all negative numbers in the list with their absolute values.
4. Create a program to accept a list of names from the user and print the longest name.

---

2. Tuple Assignments

1. Create a tuple of 10 integers. Write a Python program to:
   - Find the minimum and maximum values in the tuple.
   - Count how many times a specific number (entered by the user) occurs in the tuple.
2. Write a Python program to create a tuple with mixed data types (e.g., integers, strings, floats). Extract and display:
   - All integers in the tuple.
   - All strings in the tuple.
3. Write a program to accept a list from the user and convert it into a tuple. Ensure the tuple remains immutable and try modifying it to demonstrate its immutability.
4. Create a tuple of 5 city names and display them in reverse order.

---

3. Set Assignments

1. Write a Python program to create two sets of integers and perform the following operations:
   - Union
   - Intersection
   - Difference
2. Write a program to input 10 numbers from the user. Store them in a set to eliminate duplicates and display the unique numbers.
3. Create a set of your favorite books and another set of your friend's favorite books. Find:
   - The common books you both like.
   - Books that are unique to you.
4. Write a program to check if a given element exists in a set.

---

4. Dictionary Assignments

1. Write a program to create a dictionary where the keys are the names of students, and the values are their marks in a subject. Perform the following operations:
   - Add a new student and their marks.
   - Update marks for an existing student.
   - Remove a student from the dictionary.
2. Write a Python program to count the frequency of each word in a user-provided sentence using a dictionary.
3. Create a dictionary of products and their prices. Write a program to:
   - Display all products and their prices.
   - Find the most expensive product.
   - Calculate the average price of all products.
4. Write a program to merge two dictionaries into one. If both dictionaries have the same key, add their values.

Combined Real-Life Implementation Questions

1. **Shopping Cart System**:
   Create a program to simulate a simple shopping cart system.
   - Use a dictionary to store product names as keys and their prices as values.
   - Use a list to keep track of items added to the cart.
   - Perform the following operations:
     - Add an item to the cart.
     - Remove an item from the cart.
     - Calculate the total price of items in the cart.
2. **Student Data Management**:
   Write a program to manage student data for a school. Use the following structures:
   - **Dictionary**: To store each student's name as the key and their details (age, class, marks as another dictionary) as values.
   - **Tuple**: To store immutable data like their date of birth.
   - **Set**: To keep track of unique extracurricular activities students are involved in.
     Perform operations like adding a new student, updating marks, and displaying all unique extracurricular activities.
3. **Movie Ticket Booking System**:
   Create a Python program for a ticket booking system using:
   - **List**: To store available seats.
   - **Tuple**: To store details of booked seats (seat number, customer name).
   - **Set**: To track seat numbers already booked.
   - **Dictionary**: To store customer details (name, contact information, and booked seat).
     Perform the following operations:
   - Book a seat.
   - Cancel a booking.
   - Display all available and booked seats.
4. **Inventory Management**:
   Write a program to manage an inventory system for a small store:
   - **List**: To track the list of items currently in stock.
   - **Set**: To keep track of unique suppliers.

- **Dictionary**: To store item details (item name as the key, and stock quantity and price as values).
  Perform operations such as:
- Adding a new item to the inventory.
- Updating the stock quantity of an item.
- Displaying items with low stock.

5. **Event Organizer System**:
   Build a program to help organize an event:
   - **List**: To store the names of attendees.
   - **Tuple**: To store fixed event details like the event date, time, and venue.
   - **Set**: To track unique activities available at the event.
   - **Dictionary**: To store attendee details (name as the key and their preferences as values).
     Perform operations such as:
   - Adding a new attendee.
   - Updating attendee preferences.
   - Displaying all activities and preferences.