List – 7 Questions

1. **Create a list of 5 integers entered by the user. Print the sum of all elements.**
   Hint: Use a loop to take 5 inputs and `sum()` function.

2. **Write a program to count how many times a number appears in a list.**
   Hint: Use `list.count(x)` method.

3. **Write a program that removes all even numbers from a list.**
   Hint: Use a loop with `if number % 2 != 0`.

4. **Find the largest and smallest elements in a list.**
   Hint: Use `max()` and `min()` functions.

5. **Sort a list of numbers in ascending order.**
   Hint: Use `list.sort()` or `sorted(list)`.

6. **Create a list of fruits. Ask user to enter a fruit and check if it's in the list.**
   Hint: Use `in` keyword to check membership.

7. **Reverse the elements of a list without using reverse() function.**
   Hint: Use slicing: `list[::-1]` or loop to reverse manually.

Tuple – 6 Questions

8. **Create a tuple with 5 elements and display its length.**
   Hint: Use `len()` function.

9. **Ask the user to enter 5 numbers and store them in a tuple. Then display their average.**
   Hint: Convert input to int, use `sum()` and `len()`.

10. **Check if a specific number is present in a tuple.**
    Hint: Use `in` keyword.

11. **Convert a list of numbers into a tuple.**
    Hint: Use `tuple(your_list)`.

12. **Write a program to find the index of a number in a tuple.**
    Hint: Use `tuple.index(x)` method.

13. **Try to change the value of an element in a tuple and observe the result.**
 Hint: This will raise an error – explain immutability.

Set – 7 Questions

14. **Create a set of 5 integers. Try adding a duplicate element and print the set.**
 Hint: Sets ignore duplicates.

15. **Take two sets from user input and display: union, intersection, and difference.**
 Hint: Use `|`, `&`, and `-` operators or `set.union()` etc.

16. **Write a program to find common elements from two sets.**
 Hint: Use intersection: `set1 & set2`.

17. **Remove a specific element from a set.**
 Hint: Use `set.remove(x)` or `discard(x)` (avoid crash if not found).

18. **Check if one set is a subset of another.**
 Hint: Use `set1.issubset(set2)`.

19. **Convert a list with repeated items into a set and print unique elements only.**
 Hint: `set(list)` removes duplicates.

20. **Take a string from user and display all unique characters.**
 Hint: Convert string to set using `set(string)`.

Scenario: Grocery Billing System

Question:
You're building a simple grocery billing program. Ask the user to enter the names and prices of 5 items. Store the names in one list and the prices in another. Finally, display the total bill and all items purchased.

Hint to Students:

- Use two lists: one for item names (`item_names`), one for prices (`item_prices`).
- Use a `for` loop (5 times) to take item name and price from user.
- Use `sum(item_prices)` to get total bill.
- You can also use `zip()` to nicely display item with its price.

Bonus Tip: Show output like:

```
Apple - Rs.100

Milk - Rs.70

Total: Rs.170
```

Scenario: Student Grades (Immutable Records)

Question:
You are storing student grade records which should not be changed once entered. Ask the user to input 3 subjects' marks and store them as a tuple. Then calculate and display the average mark.

Hint to Students:

- Tuples are good when data shouldn't be changed.
- Use `tuple()` to store the marks.
- Use `sum()` and `len()` to calculate average.

Bonus Tip:
Explain why tuples are good for marks: they are fixed once entered, and cannot be accidentally changed.

🧾 Scenario: Unique Visitors Tracker

Question:
Imagine you're tracking website visitors by their names (or IDs). Ask the user to enter visitor names (with possible duplicates). Store them in a set to get only unique visitors. Finally, display how many unique people visited.

Hint to Students:

- Use a `set()` to store names.
- Use a loop to take input multiple times.
- Sets automatically remove duplicates.
- Use `len(visitor_set)` to count unique visitors.

Bonus Challenge:
Allow user to enter names until they type `"done"`.

```
visitor_set = set()

while True:

    name = input("Enter visitor name (or 'done' to finish): ")

    if name.lower() == 'done':

        break

    visitor_set.add(name)
```