

## Module 3: Data Structures in Python – Practice Problems

---

### Section 1: Lists – Basics and Manipulation

#### Basic List Operations

1. Create a list of 5 integers.
  2. Print the first and last element of a list.
  3. Add a new element to the end of the list.
  4. Insert an element at the 2nd position.
  5. Remove the last element using `pop()`.
  6. Delete a specific element by value.
  7. Sort a list of numbers in ascending order.
  8. Reverse the list using `.reverse()` and slicing.
  9. Replace the second item in the list with a new value.
  10. Count how many times an element appears in the list.
- 

### Section 2: List Iteration & Aggregation

11. Print each element in a list using a for loop.
  12. Find the sum of all numbers in a list.
  13. Find the maximum and minimum number in a list.
  14. Multiply all items in a list.
  15. Find the average of numbers in a list.
  16. Separate even and odd numbers from a list.
  17. Count how many items in a list are greater than 50.
  18. Create a list of squares from 1 to 10.
  19. Remove all duplicates from a list.
  20. Create a new list with elements in reverse order.
- 

### Section 3: Advanced List Operations

21. Concatenate two lists.
22. Find the common elements between two lists.
23. Find elements that are in list A but not in B.

24. Flatten a 2D list into a 1D list.
  25. Split a list into 2 halves.
  26. Copy a list without using `copy()` method.
  27. Rotate a list to the right by 2 positions.
  28. Check if a list is sorted.
  29. Shuffle a list (use `random.shuffle()`).
  30. Find all indices of a given element in a list.
- 

#### **Section 4: Tuples – Creation and Use Cases**

31. Create a tuple with at least 3 elements.
  32. Access elements by index.
  33. Check if an element exists in a tuple.
  34. Convert a list to a tuple.
  35. Convert a tuple to a list.
  36. Create a tuple of squares from 1 to 10.
  37. Count occurrences of an element in a tuple.
  38. Find the index of an element in a tuple.
  39. Unpack a tuple into variables.
  40. Slice a tuple from the second to the fourth element.
- 

#### **Section 5: List vs Tuple Exercises**

41. Create a program that accepts comma-separated numbers and stores them in a tuple and a list.
  42. Demonstrate the immutability of tuples.
  43. Compare memory usage between a list and a tuple.
  44. Show an example of tuple packing and unpacking.
- 

#### **Section 6: Dictionaries – Basics and Manipulation**

45. Create a dictionary to store a student's name, age, and grade.
46. Access and print a value using its key.
47. Update the age of the student.

48. Add a new key: "email".
  49. Delete a key-value pair using pop().
  50. Use .get() to safely access a value.
  51. Print all keys in a dictionary.
  52. Print all values in a dictionary.
  53. Loop through a dictionary and print key-value pairs.
  54. Check if a key exists in a dictionary.
- 

### Section 7: Dictionary Use Cases

55. Count frequency of characters in a string.
  56. Count word frequency in a sentence.
  57. Create a dictionary from two lists (keys and values).
  58. Sort a dictionary by values.
  59. Find the key with the maximum value.
  60. Merge two dictionaries.
  61. Create a nested dictionary of students and their marks.
  62. Print all students and their total marks.
  63. Invert a dictionary (keys → values, values → keys).
  64. Filter out key-value pairs where value < 50.
- 

### Section 8: Sets – Basics and Set Operations

65. Create a set with duplicate values and observe the result.
66. Add a new element to a set.
67. Remove an element from a set.
68. Check membership of an element in a set.
69. Convert a list with duplicates to a set.
70. Find the union of two sets.
71. Find the intersection of two sets.
72. Find the difference between two sets.
73. Check if one set is a subset of another.
74. Clear all elements from a set.

---

### Section 9: Use Cases with Sets

- 75. Remove duplicates from a list using a set.
- 76. Given two lists, print common elements using sets.
- 77. Count the number of unique words in a sentence.
- 78. Find elements that are in either of two sets but not both.
- 79. Create a set of unique characters from a string.
- 80. Check if two strings are anagrams using sets.

---

### Section 10: Combined Data Structure Tasks

- 81. Convert a list of tuples into a dictionary.
- 82. Convert a dictionary into a list of tuples.
- 83. Create a matrix (list of lists) and print it.
- 84. Transpose a 2D list (matrix).
- 85. Find the sum of each row in a 2D list.
- 86. Create a dictionary where key = number and value = square of the number.
- 87. Create a list of dictionaries representing a class of students.
- 88. Sort the list of students by age.
- 89. Group names by their first letter using a dictionary.
- 90. Create an inventory system using a dictionary.

---

### Section 11: Real-World Simulations

- 91. Create a contact book using a dictionary (name: phone number).
- 92. Implement a shopping cart using a dictionary (item: quantity).
- 93. Create a dictionary to store login credentials.
- 94. Build a system that stores movie ratings (title: rating).
- 95. Make a language dictionary (EN → FR) with 5 words.

---

### Section 12: Mini Challenges & Fun Problems

- 96. Print the top 3 most frequent elements in a list.
- 97. Group words by their lengths using a dictionary.

98. From a list of words, find those with unique letters.
99. Create a dictionary to track frequency of digits in a number.
100.       Combine multiple dictionaries into one (handling duplicate keys).
101.       Write a function that accepts a sentence and returns the most frequent word.
102.       Create a histogram from a list of numbers using dictionaries.
103.       Given a list of email addresses, count how many come from each domain.
104.       Build a basic flashcard quiz system using a dictionary (question: answer).
105.       Create a dictionary of employee records with nested data (emp\_id: {name, salary}).