SAMPLE DATA STRUCTURE CHITS FOR MOCK EXAM 2025

1. Write a Python program to manage the **borrowing records of books in a library.** Implement the following functionalities:
   - Compute the average number of books borrowed by all library members.
   - Find the book with the highest and lowest number of borrowings in the library.
   - Count the number of members who have not borrowed any books (denoted by a borrow count of 0).
   - Display the most frequently borrowed book (i.e., the mode of borrow counts).

2. Write a Python program to store marks scored in **subject "Data Structure" by N students** in the class. Write functions to compute following:
   a) The average score of class
   b) Highest score and lowest score of class
   c) Count of students who were absent for the test
   d) Display mark with highest frequency

3. In an **e-commerce system**, customer account IDs are stored in a list, and you are tasked with writing a program that implements the following:
   - **Linear Search:** Check if a particular customer account ID exists in the list. improving the search efficiency over the basic linear.

4. In an **e-commerce system**, customer account IDs are stored in a list, and you are tasked with writing a program that implements the following:
   - **Binary Search:** Implement Binary search to find if a customer account ID exists, improving the search efficiency over the basic linear.

5. Write a Python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using **Linear search.**

6. Write a Python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using **Binary search**

7. In a **company,** employee salaries are stored in a list as floating-point numbers. Write a Python program that sorts the employee salaries in ascending order using the following two algorithms:
   - **Selection Sort**: Sort the salaries using the selection sort algorithm.

   After sorting the salaries, the program should display top five highest salaries in the company

8. In a **company**, employee salaries are stored in a list as floating-point numbers. Write a Python program that sorts the employee salaries in ascending order using the following two algorithms:
    - **Bubble Sort:** Sort the salaries using the bubble sort algorithm.

   After sorting the salaries, the program should display top five highest salaries in the company.

9. Write a Python program to store **first year percentage of students in array**. Write function for sorting array of floating point numbers in ascending order using **a) Selection Sort and display top five scores.**

10. Write a Python program to store **first year percentage of students in array**. Write function for sorting array of floating point numbers in ascending order using **a) Bubble Sort and display top five scores.**

11. Implementing a **real-time undo/redo system** for a text editing application using a **Stack data structure.** The system should support the following operations:
- Make a Change: A new change to the document is made.
- Undo Action: Revert the most recent change and store it for potential redo.
- Redo Action: Reapply the most recently undone action.
- Display Document State: Show the current state of the document after undoing or redoing an action.

12. Implement a **real-time event processing system** using a **Queue data structure**. The system should support the following features:
    - Add an Event: When a new event occurs, it should be added to the event queue.
    - Process the Next Event: The system should process and remove the event that has been in the queue the longest.
    - Display Pending Events: Show all the events currently waiting to be processed.
    - Cancel an Event: An event can be canceled if it has not been processed.

13. A **call center** receives incoming calls, and each call is assigned a unique customer ID. The calls are answered in the order they are received. Your task is to simulate the call queue of a call center using a **queue data structure**.
    - addCall(customerID, callTime): Add a call to the queue with the customer ID and the call time (in minutes).
    - answerCall(): Answer and remove the first call from the queue.
    - viewQueue(): View all calls currently in the queue without removing them.
    - isQueueEmpty(): Check if the queue is empty.

14. **Queues** are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write Python program for simulating job queue. **Write functions to add job and delete job from queue.**

15. Implement a hash table of size 10 and **use the division method** as a hash function. In case of a collision, **use chaining**. Implement the following operations:
    - Insert(key): Insert key-value pairs into the hash table.
    - Search(key): Search for the value associated with a given key.
    - Delete(key): Delete a key-value pair from the hash table

16. Design and implement a hash table of fixed size. **Use the division method** for the hash function and resolve **collisions using linear probing**. Allow the user to perform the following operations:
    - Insert a key
    - Search for a key
    - Delete a key
    - Display the table

17. Consider a particular area in your city. Note the popular locations 0, 1, 2, 3 . . . in that area. Assume these locations represent nodes of a graph. If there is a route between two locations, it is represented as connections between nodes. Find out the sequence in which you will visit these locations, starting from (say 0) using (i) **BFS and Represent a given graph using an adjacency list to perform BFS.**

18. Consider a particular area in your city. Note the popular locations 0, 1, 2, 3 . . . in that area. Assume these locations represent nodes of a graph. If there is a route between two locations, it is represented as connections between nodes. Find out the sequence in which you will visit these locations, starting from (say 0) using **(i) DFS. Represent a given graph using an adjacency matrix to perform DFS.**

19. Implement various operations on a **Binary Search Tree**, such as insertion, deletion, display, and search.

20. Construct an **expression tree from the given prefix expression**, e.g., +--a*bc/def, traverse it using **post-order traversal (non-recursive),** and then delete the entire tree.