**Tasks for practicing arrays, stacks, and lists:**

**Array Tasks**

**1. Sum and Average:** Create an array of integers with 10 elements. Write a program to calculate and display the sum and average of the array elements.

**2. Max and Min:** Given an array of 5 integers, write a program to find the maximum and minimum values in the array.

**3. Reverse an Array:** Write a program to reverse the elements of an array with 8 elements and display the reversed array.

**4. Count Occurrences:** Create an array of characters. Write a program to count how many times a specific character (input by the user) appears in the array.

**5. Remove Duplicates:** Write a program that removes duplicate elements from an array of integers and prints the unique values.

**Stack Tasks**

**1. Palindrome Check:** Use a stack to check if a word entered by the user is a palindrome (a word that reads the same backward and forward).

**2. Balanced Parentheses:** Write a program using a stack to check if a string of parentheses (`(), {}, []`) is balanced. For example, `(())` is balanced, but `(()` is not.

**3. Browser History:** Simulate a basic browser history using a stack. Allow the user to "visit" a new page (push) and "go back" to the previous page (pop).

**4. Evaluate Postfix Expression:** Given a postfix expression (e.g., "5 1 2 + 4 \* + 3 -"), write a program to evaluate it using a stack.

**5. Undo Function:** Implement an "Undo" function for a text editor. Each action (input by the user) is pushed to the stack, and when the user chooses "Undo," the last action is popped.

**List Tasks**

**1. Shopping List:** Create a program that allows the user to add items to a shopping list (using a linked list), remove items, and display the current list.

**2. Task Manager:** Write a program to manage tasks using a list. Each task has a priority (e.g., high, medium, low). The user can add, remove, and display tasks in priority order.

**3. Student Roll Call:** Create a list of student names. Allow the user to add or remove names, and print the list in alphabetical order.

**4. FIFO Queue Simulation**: Using a list, simulate a first-in-first-out (FIFO) queue for a ticketing system. Allow users to "take a ticket" (add to the end) and "serve a ticket" (remove from the front).

**5. Playlist Manager:** Create a music playlist manager where the user can add songs at the beginning, end, or specific position, remove songs, and display the playlist.