Experiment No: 8 - Menu-Driven List Operations and Analysis Program

Aim

To design and implement a menu-driven Python program that performs various operations on a list such as insertion, deletion, searching, sorting, and analysis using built-in list methods, until the user chooses the Exit option.

Problem Statement

Write a Python program that displays a menu repeatedly and performs the selected operation on a list of integers until the user selects the Exit option.

The menu should include the following choices:

1. **Create List**
   Note:
   - Accept list elements from the user
   - If an existing list is present, display a **warning message** and **overwrite** the old list with a new list
2. **Display List**
3. **Analyze List (Find Sum, Maximum, Minimum, Average, and Sorted View)**
   Note:
   - Display sum, max, min, and average
   - Display sorted list in ascending and descending order
   - **Sorting should not affect the original list**
4. **Insert an Element at a Specified Position**
5. **Delete an Element from the List**
6. **Search for an Element**
7. **Exit the Program**

Concepts Used

- Lists
- Menu-driven programming
- Built-in list methods
- Conditional statements
- Looping constructs

Result

Thus, a menu-driven Python program to perform various list operations and analysis using built-in list methods was successfully designed and executed.

Sample Input / Output:

```
----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
```

```
Enter your choice: 1
Enter number of elements: 4
Enter element: 32
Enter element: 43
Enter element: 54
Enter element: 65
List created successfully.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 1
Warning: Existing list will be cleared.
Enter number of elements: 3
Enter element: 54
Enter element: 64
Enter element: 23
List created successfully.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 2
Original List: [54, 64, 23]

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 3
Sum           : 141
Maximum   : 64
Minimum   : 23
Average      : 47.0
```

Sorted (Ascending) [View Only]: [23, 54, 64]
Sorted (Descending) [View Only]: [64, 54, 23]
Note: Original list remains unchanged.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 2
Original List: [54, 64, 23]

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 4
Enter element to insert: 44
Enter position: 1
Element inserted successfully.
Updated List: [54, 44, 64, 23]

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 5
Enter element to delete: 33
Element not found.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element

7. Exit
Enter your choice: 5
Enter element to delete: 23
Element deleted successfully.
Updated List: [54, 44, 64]

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 6
Enter element to search: 34
Element not found in the list.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 6
Enter element to search: 54
Element found in the list.

----- MENU -----
1. Create List
2. Display List
3. Analyze List
4. Insert an Element
5. Delete an Element
6. Search an Element
7. Exit
Enter your choice: 7
Exiting the program. Thank you!