```
--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 10
Element 10 pushed into stack.

--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter the value to push: 20
Element 20 pushed into stack.

--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements are:
20
10

--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Popped element: 20
```

```
--- Stack Menu ---
1. Push
2. Pop
3. Display
Enter your choice: 2
Popped element: 10

--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Stack Underflow! Cannot pop element.

--- Stack Menu ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 4
Exiting program.
```

```
Enter a valid parenthesized infix expression: (A+B)*(C-D)
Postfix expression: AB+CD-
```

```
--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 10
Inserted 10 into queue.

--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 20
Inserted 20 into queue.

--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements:
10 20

--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 10

--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 20

--- Linear Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Queue Empty! Cannot delete.
```

```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 10
Inserted 10 into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 20
Inserted 20 into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Circular Queue elements:
10 20

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 10

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Circular Queue elements:
20
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at Position
4. Insert at End
5. Display
6. Exit
Enter your choice: 1
Enter number of nodes: 3
Enter data: 10
Enter data: 20
Enter data: 30

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at Position
4. Insert at End
5. Display
6. Exit
Enter your choice: 2
Enter data: 5
Node inserted at beginning.

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at Position
4. Insert at End
5. Display
6. Exit
Enter your choice: 3
Enter position: 3
Enter data: 15
Node inserted at position 3.
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at Position
4. Insert at End
5. Display
6. Exit
Enter your choice: 4
Enter data: 40
Node inserted at end.

--- Singly Linked List Menu ---
1. Create Linked List
2. Insert at Beginning
3. Insert at Position
4. Insert at End
5. Display
6. Exit
Enter your choice: 5
Linked List: 5 -> 10 -> 15 -> 20 -> 30 -> 40 -> NULL
```

```
--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 1
Enter number of nodes: 4
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30
Enter data for node 4: 40

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 10 -> 20 -> 30 -> 40 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 2
First node deleted.

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 20 -> 30 -> 40 -> NULL


--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 3
Enter element to delete: 30
Node with value 30 deleted.

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 20 -> 40 -> NULL

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 4
Last node deleted.

--- Singly Linked List Menu ---
1. Create Linked List
2. Delete First Element
3. Delete Specified Element
4. Delete Last Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 20 -> NULL
```

```
Create first linked list
Enter number of nodes: 4
Enter data for node 1: 40
Enter data for node 2: 10
Enter data for node 3: 30
Enter data for node 4: 20

--- Menu ---
1. Display List
2. Sort List
3. Reverse List
4. Concatenate with another list
5. Exit
Enter your choice: 1
Linked List: 40 -> 10 -> 30 -> 20 -> NULL

--- Menu ---
1. Display List
2. Sort List
3. Reverse List
4. Concatenate with another list
5. Exit
Enter your choice: 2
Linked list sorted.
Linked List: 10 -> 20 -> 30 -> 40 -> NULL

--- Menu ---
1. Display List
2. Sort List
3. Reverse List
4. Concatenate with another list
5. Exit
Enter your choice: 3
Linked list reversed.
Linked List: 40 -> 30 -> 20 -> 10 -> NULL

--- Menu ---
1. Display List
2. Sort List
3. Reverse List
4. Concatenate with another list
5. Exit
Create second linked list
Enter number of nodes: 2
Enter data for node 1: 50
Enter data for node 2: 60
Linked lists concatenated.
Linked List: 40 -> 30 -> 20 -> 10 -> 50 -> 60 -> NULL
```

```
--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 1
Enter value to push: 10
Pushed 10 into stack.

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 1
Enter value to push: 20
Pushed 20 into stack.

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 3
Stack (Top to Bottom): 20 -> 10 -> NULL

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 2
Popped element: 20

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 3
Stack (Top to Bottom): 10 -> NULL

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 4
Enter value to enqueue: 30
Enqueued 30 into queue.
```

```
--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 4
Enter value to enqueue: 40
Enqueued 40 into queue.

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 6
Queue (Front to Rear): 30 -> 40 -> NULL

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 5
Dequeued element: 30

--- Menu ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 6
Queue (Front to Rear): 40 -> NULL
```

```
--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 1
Enter number of nodes: 3
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30

--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 4
Doubly Linked List: 10 <-> 20 <-> 30 <-> NULL

--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 2
Enter value to insert left of: 20
Enter new data: 15
Node inserted to the left of 20.

--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 4
Doubly Linked List: 10 <-> 15 <-> 20 <-> 30 <-> NULL


--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 3
Enter value to delete: 30
Node with value 30 deleted.

--- Doubly Linked List Menu ---
1. Create List
2. Insert Node to Left of Given Value
3. Delete Node by Value
4. Display List
5. Exit
Enter your choice: 4
Doubly Linked List: 10 <-> 15 <-> 20 <-> NULL
```

```
Enter number of nodes: 6
Enter value 1: 50
Enter value 2: 30
Enter value 3: 70
Enter value 4: 20
Enter value 5: 40
Enter value 6: 60

In-order Traversal: 20 30 40 50 60 70
Pre-order Traversal: 50 30 20 40 70 60
Post-order Traversal: 20 40 30 60 70 50
PS C:\Users\Muhammad Rabbani\OneDrive\Desktop\DATA STRUCTURE>
```

```
Enter number of vertices: 5
Enter adjacency matrix:
0 1 1 0 0
1 0 0 1 0
1 0 0 1 1
0 1 1 0 1
0 0 1 1 0
Enter starting vertex: 0
BFS Traversal: 0 1 2 3 4
PS C:\Users\Muhammad Rabbani\OneDrive\Desktop\DATA STRUCTURE>
```

```
Enter number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 1 1
1 1 0 0
0 1 0 0
The given graph is CONNECTED.
Enter number of vertices: 4
Enter adjacency matrix:
0 1 0 0
1 0 0 0
0 0 0 1
0 0 1 0
The given graph is NOT CONNECTED.
```

```
Enter size of Hash Table (m): 10

--- Hashing with Linear Probing ---
1. Insert Employee Record
2. Search Employee Record
3. Display Hash Table
4. Exit
Enter your choice: 1
Enter 4-digit Key: 1234
Enter Employee ID: 101
Enter Employee Name: RAVI
Record inserted at address 4

--- Hashing with Linear Probing ---
1. Insert Employee Record
2. Search Employee Record
3. Display Hash Table
4. Exit
Enter your choice: 1
Enter 4-digit Key: 1244
Enter Employee ID: 102
Enter Employee Name: ANIL
Record inserted at address 5

--- Hashing with Linear Probing ---
1. Insert Employee Record
2. Search Employee Record
3. Display Hash Table
4. Exit
Enter your choice: 1
Enter 4-digit Key: 1254
Enter Employee ID: 103
Enter Employee Name: SITA
Record inserted at address 6
```

```
--- Hashing with Linear Probing ---
1. Insert Employee Record
2. Search Employee Record
3. Display Hash Table
4. Exit
Enter your choice: 3

Hash Table Contents:
Address Key     EmpID   Name
0       ---
1       ---
2       ---
3       ---
4       1234    101     RAVI
5       1244    102     ANIL
6       1254    103     SITA
7       ---
8       ---
9       ---

--- Hashing with Linear Probing ---
1. Insert Employee Record
2. Search Employee Record
3. Display Hash Table
4. Exit
Enter your choice: 2
Enter Key to search: 1244
Record Found at address 5
Employee ID: 102
Name: ANIL
```