CSE 2216 : Data Structures and Algorithm 1 Lab

Problem Set for Stack and Queue

Stack Problems

1. **Implement a Stack Using Array**
   - **Problem:** Create a stack using an array. Implement `push`, `pop`, `peek`, and `isEmpty` operations.
   - **Example Input:**

     ```
     Operations: push(10), push(20), push(30), pop()
     ```

   - **Example Output:**

     ```
     Stack: 10 → 20
     ```

2. **Reverse a String Using Stack**
   - **Problem:** Use a stack to reverse a given string.
   - **Example Input:**

     ```
     String: "HELLO"
     ```

   - **Example Output:**

     ```
     Reversed String: "OLLEH"
     ```

3. **Check for Balanced Parentheses**
   - **Problem:** Write a program to check if a string has balanced parentheses using a stack.
   - **Example Input:**

     ```
     String: "(a+b)*(c+d)"
     ```

   - **Example Output:**

     ```
     Balanced: Yes
     ```

4. **Sort a Stack**

   - **Problem:** Write a function to sort the elements of a stack such that the smallest elements are on the top. You can only use one additional stack for temporary storage.
   - **Example Input:**

     ```
     Stack: 30 → 10 → 20 → 50 → 40
     ```

   - **Example Output:**

     ```
     Sorted Stack: 10 → 20 → 30 → 40 → 50
     ```

5. **Find the Minimum Element in a Stack**

   - **Problem:** Implement a stack with an additional operation `getMin` that retrieves the minimum element in constant time.
   - **Example Input:**

     ```
     Operations: push(10), push(20), push(5), getMin()
     ```

   - **Example Output:**

     ```
     Minimum Element: 5
     ```

Queue Problems

1. **Implement a Queue Using Array**

   - **Problem:** Create a queue using an array. Implement `enqueue`, `dequeue`, `peek`, and `isEmpty` operations.
   - **Example Input:**

     ```
     Operations: enqueue(10), enqueue(20), dequeue()
     ```

   - **Example Output:**

     ```
     Queue: 20
     ```

2. **Reverse a Queue**

   - **Problem:** Write a function to reverse the elements of a queue.
   - **Example Input:**

     ```
     Queue: 10 → 20 → 30 → 40
     ```

   - **Example Output:**

     ```
     Reversed Queue: 40 → 30 → 20 → 10
     ```

3. **Implement a Circular Queue**

   - **Problem:** Design a circular queue using an array with support for `enqueue` and `dequeue`.
   - **Example Input:**

     ```
     Capacity: 3, Operations: enqueue(10), enqueue(20),

     enqueue(30), enqueue(40), dequeue()
     ```

   - **Example Output:**

     ```
     Queue: 20 → 30 → 40
     ```

4. **Generate Binary Numbers Using Queue**

   - **Problem:** Write a program to generate the first `n` binary numbers using a queue.
   - **Example Input:**

     ```
     n = 5
     ```

   - **Example Output:**

     ```
     Binary Numbers: 1, 10, 11, 100, 101
     ```

5. **Implement Stack using two Queues and Queue using two stacks.**