**Stack**

**Theory Questions (Basic to Advanced)**

**🟢 Basic:**

1. What is a stack?
2. What are the basic operations of a stack?
3. What is LIFO? Explain with an example.
4. How is a stack different from a queue?
5. What is the time complexity of push/pop/peek operations?

**🟡 Intermediate:**

1. How is a stack implemented internally (array vs linked list)?
2. What are real-life examples of stacks?
3. What is stack overflow and underflow?
4. What is the use of the call stack in Java?
5. What are the advantages and disadvantages of stack?

**🔴 Advanced:**

1. Can we implement multiple stacks in a single array?
2. How do stacks help in parsing expressions (infix/postfix)?
3. Explain how recursion uses a stack internally.
4. Explain memory usage when using stack vs heap.
5. What is the role of stack in DFS (Depth First Search)?

**💻 Coding Questions (Basic to Advanced)**

**🟢 Basic:**

1. Implement a stack using array
2. Implement a stack using linked list
3. Implement a stack using Java's built-in Stack class

**🟡 Intermediate:**

1. Reverse a string using stack
2. Check for balanced parentheses using stack
3. Evaluate postfix expression using stack
4. Sort a stack using another stack
5. Design a stack that supports getMin() in O(1) time
6. Implement two stacks in one array

**🔴 Advanced:**

1. Implement a stack with O(1) getMin() and getMax()
2. Largest rectangle in a histogram (using stack)
3. Max area in binary matrix (based on histogram logic)
4. Next Greater Element using stack
5. Remove K digits to make the smallest number (uses monotonic stack)
6. Design a custom browser history (forward/back using stacks)