Question #1:
The MinStack class is meant to be a stack that supports the following operations:

- push(int val): Pushes a value onto the stack.
- pop(): Removes the top value from the stack.
- getMin(): Retrieves the minimum value in the stack.
- getLast(): Retrieves the last (most recent) value pushed onto the stack.

To efficiently support the getMin operation, you can use an auxiliary stack (minStack) to keep track of the minimum value:

1. **Main Stack (myStack)**: Holds all the values.
2. **Auxiliary Stack (minStack)**: Keeps track of the current minimum value.

Implementation Details

1. **Push Operation**:
    - Push the value onto the main stack.
    - Update the minimum stack: If minStack is empty or the new value is smaller than the current minimum, push the new minimum onto minStack.
2. **Pop Operation**:
    - Pop the value from the main stack.
    - If the value matches the top of minStack, pop from minStack as well.
3. **Get Minimum**:
    - Return the top value of minStack, which is the current minimum.
4. **Get Last**:
    - Return the top value of the main stack.

Question # 2:
The StackCust class implements a basic stack with operations to push, pop, peek, and check if the stack is full or empty. Here's a quick overview and some improvements to ensure the class behaves correctly:

Key Points and Corrections

1. **Constructor**: Initializes the stack with a given size and sets the top index to -1.
2. **isEmpty()**: Checks if the stack is empty.
3. **isFull()**: Checks if the stack is full.
4. **push(int elem)**: Adds an element to the stack if it's not full.
5. **peek()**: Returns the top element of the stack without removing it.
6. **pop()**: Removes and returns the top element of the stack.

Issues in the Current Implementation

- **pop() Method**: The pop() method should return the element that was removed. The current implementation returns arr[top] after decrementing top, which could be incorrect if the stack was empty.
- **peek() Method**: The peek() method correctly checks if the stack is empty before accessing arr[top].