Implementing Custom Stack & Queue in C#

Reflection

Implementing both stack and queue in this project really highlighted how different their logic is in practice. The stack comes together faster, as you push to add, pop to undo, and its one directional flow is easy to visualize. The queue took a bit more brainpower since elements enter at the back and leave from the front. This makes testing order especially important to prove FIFO behavior. If I’m building something like an undo history, recursion, or call stack, I’d reach for a stack every time. But for print jobs, task scheduling, or anything that needs fairness and order, a queue is the better fit.