1) Why does a stack (LIFO) work well for *step history / undo* but not for waiting lines?

- **Undo / step history:** Undo operations are naturally LIFO: the last action you performed is the first one you want to undo. A stack exactly models that behavior: push every new action;P`pop()` removes the most recent, restoring the previous state. This matches user expectations (you undo your last step first), and it's simple to implement and reason about.
- **Waiting lines:** Real-world waiting lines (queues) require fairness and temporal order: those who arrived first expect to be served first. LIFO would let the newest arrival cut the line (served first), producing chaos and unfairness. This LIFO violates the fairness and predictability requirements of service lines. In short: undo needs last-in-first-out behavior; waiting lines need first-in-first-out.

2) Why does FIFO (queues) prevent conflicts in food services?

- **Fairness:** Serving in arrival order means there is a clear and agreed rule everyone understands (first-come, first-served). This reduces disputes about "who is next."
- **Predictability:** People can estimate waiting time because the queue order is stable. Staff and customers both know the next person to serve.
- **Simplicity & low contention:** FIFO reduces attempts to jump the line and minimizes conflict points; it sets an objective criterion for service order.
- **Throughput and service modeling:** Queueing theory (e.g., M/M/1 models) shows that FIFO with disciplined service often yields acceptable average wait times and is easier to analyze and plan around (staffing, peak times).
- **Psychological fairness:** People view FIFO as morally fair; perceived fairness reduces tension and complaints in busy canteens.