**1. Using Queue: "Patient Registration and Checkup" (FIFO)**

* **Problem:** In a hospital, patients arrive and register at the reception. The first patient registered should be checked by the doctor first.
* **Solution:** Use a **queue**, where patients are **added** at the end when they register and **removed** from the front when the doctor checks them.

**2. Using Stack: "Managing Patient Records (Undo Operations)" (LIFO)**

* **Problem:** If a receptionist enters incorrect patient details, they should be able to undo the last entry.
* **Solution:** Use a **stack**, where each patient’s details are **pushed** when entered and **popped** if an undo is needed.

**3. Using Linked List: "Managing Appointments Dynamically"**

* **Problem:** Appointments are booked dynamically, and patients may cancel or reschedule.
* **Solution:** Use a **linked list**, where each appointment is a **node** linked to the next one. If a patient cancels, the node is deleted.

**4. Using Deque: "Emergency and Normal Patient Handling"**

* **Problem:** Hospitals must prioritize emergency patients while still handling normal cases.
* **Solution:** Use a **deque**, where emergency patients are added at the **front**, and normal patients are added at the **back**.