**LLD.01**
Design and implement in Go a parking lot system. There are multiple floors in the parking lot. Each floor has parking spots for different vehicle types: bicycles, motorcycles and automobiles. Parking spots are arranged in rows and columns. Some of the parking spots are inactive.

Note: The parking lot has multiple gates, ensure the code is thread-safe for concurrent access.

**Constraints**

- 1 <= floors <= 8
- 1 <= rows <= 1000
- 1 <= column <= 1000
- Each floor will have the same number of rows
- Each rows will have the same number of columns
- Each parking spot is of the following type:
    - “B-1”, active for bicycles
    - “M-1”, active for motorcycles
    - “A-1”, active for automobiles
    - “X-0”, inactive

**Requirements**

- Park vehicle. Given a vehicle type, assign an empty parking spot id and map the vehicleNumber. **spotId** is **floor-row-column**. If no free spot is found, return an error.
    - park(vehicleType, vehicleNumber)
- Unpark vehicle. Removes vehicle from parking spot. Return an error for failure to unpark a vehicle.
    - unpark(spotId, vehicleNumber)
- Available spot. Display the free spots for each vehicle type.
    - availableSpot(vehicleType)
- Search vehicle. If the vehicle has been **unparked**, get its last **spotId**.
    - searchVehicle(vehicleNumber)