**Exercise 3: Car Parking**

Write a program that simulates a car parking system. The program should have the following features:

1. Maintain a list or dictionary to represent the available parking spaces. Each parking space can be represented by a unique identifier.

2. Implement a function to check the availability of a parking space. This function should take the parking space identifier as input and return True if the space is available or False if it is occupied.

3. Implement a function to park a car in a specific parking space. This function should take the parking space identifier and the car's registration number as input. It should update the parking space status to occupied and store the car's information.

4. Implement a function to remove a car from a parking space. This function should take the parking space identifier as input, update the parking space status to available, and remove the car's information.

5. Allow the user to interact with the program by providing a menu with options to check availability, park a car, or remove a car.

6. Continuously display the current status of parking spaces, indicating which ones are available and which ones are occupied.

Here's a code skeleton to get you started:

| parking\_spaces = {  "A1": {"occupied": False, "car": None},  "A2": {"occupied": False, "car": None},  "A3": {"occupied": False, "car": None}  *# Add more parking spaces as needed* }  def check\_availability(space\_id):  *# Write your code*  def park\_car(space\_id, registration\_number):  *# Write your code*  def remove\_car(space\_id):  *# Write your code*  def display\_parking\_status():  *# Write your code*  *# Example usage:* display\_parking\_status()  print(check\_availability("A1")) *# Output: False*  park\_car("A1", "ABC123") print(check\_availability("A1")) *# Output: True*  remove\_car("A1") print(check\_availability("A1")) *# Output: False*  display\_parking\_status() |
| --- |