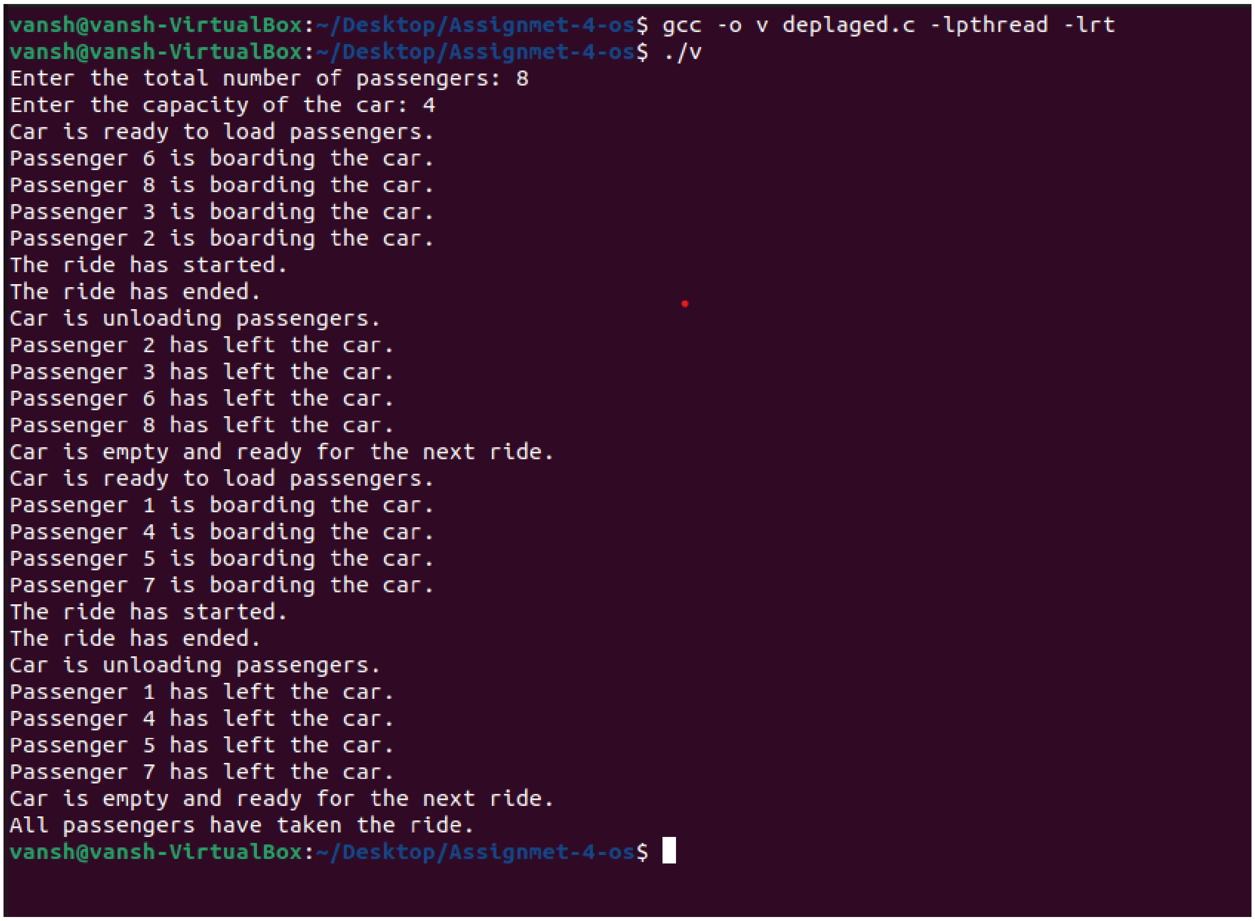
Question 2

The provided code simulates a scenario where a car can only set off when fully occupied, with a maximum capacity less than the total number of waiting passengers. The car and passenger interactions are modeled as threads using semaphores for synchronization. The car thread manages the loading, running, and unloading procedures, while the passenger threads handle boarding and getting off the ride. Semaphores such as carSem, passengerSem, and rideSem are initialized to control access to shared resources. Essential parts, which include actions such as loading and unloading, use semaphores for protection to prevent race conditions. The code makes sure the car and passenger threads work together correctly by using loops until all passengers have had their ride. It uses sleep functions to show the time taken for different actions. It also handles user input for total passengers and car capacity in a suitable way. The code is skillfully geared towards preventing issues that could arise from concurrency, putting emphasis on right synchronization and atomic operations.

Screenshot of sample output for 8 total number of passenger and 4 as the capacity of the car given below:



```
vansh@vansh-VirtualBox:~/Desktop/Assignmet-4-os$ gcc -o v deplaged.c -lpthread -lrt
vansh@vansh-VirtualBox:~/Desktop/Assignmet-4-os$ ./v
Enter the total number of passengers: 8
Enter the capacity of the car: 4
Car is ready to load passengers.
Passenger 6 is boarding the car.
Passenger 8 is boarding the car.
Passenger 3 is boarding the car.
Passenger 2 is boarding the car.
The ride has started.
The ride has ended.
Car is unloading passengers.
Passenger 2 has left the car.
Passenger 3 has left the car.
Passenger 6 has left the car.
Passenger 8 has left the car.
Car is empty and ready for the next ride.
Car is ready to load passengers.
Passenger 1 is boarding the car.
Passenger 4 is boarding the car.
Passenger 5 is boarding the car.
Passenger 7 is boarding the car.
The ride has started.
The ride has ended.
Car is unloading passengers.
Passenger 1 has left the car.
Passenger 4 has left the car.
Passenger 5 has left the car.
Passenger 7 has left the car.
Car is empty and ready for the next ride.
All passengers have taken the ride.
vansh@vansh-VirtualBox:~/Desktop/Assignmet-4-os$
```