Report for Question 3

Ober Cab Services

Problem has been solved using the concept of multithreading and synchronization between them

Main logic flow:

Creates threads for each rider and joins them in the main function.
i.e. Program ends when all the riders have been served for the day.

Then these steps follow in the thread:

1) Generate cab type , max wait time and ride time as follows:

```
cabType=rand()%2;
maxWaitTime=1+rand()%6;
RideTime=1+rand()%5;
```

2) Wait for a random amount of time and then print that this rider is online right now.

3) Call the function BookCab which searches for a cab.

4) It uses a semaphore of size D = (Number of drivers) to check keep a track on how many cabs are free and allows the rider to search accordingly.

5) If an empty cab is found it is acquired by that rider and it’s status is changed accordingly.

6) The semaphore is checked only till a timeout of maxWaitTime and returns with an error message if that happens. To do this we make use of the function sem_timedwait().

7) It then sleeps for RideTime and then signals the cab waiting on this thread to continue.

8) Then the function MakePayment() is called which checks for empty servers using a semaphore of size = S (Number of Servers) , like 4.

9) If a server is found, then a wait of 2 seconds is initiated and then the function returns printing that the rider has paid for the ride.Also the server id is printed.

Mutex locks have been used appropriately wherever shared memory comes into role.