//*Ober Cab Services*//

There are **n** cabs, **m** riders and **k** payment servers. n, m and k are user inputs and rest of the variables are randomly generated using `***rand()***` function in C with seed current time. Each rider arrives at a random time and books a cab (either premier or pool). All riders and payment servers are threads. **One mutex lock per cab** and **payment server is used**. One mutex lock is used to keep track of remaining riders.

Cabs :

- Each cab is a struct.

- cab_status denotes the current status of the cab.
- 0 : can accept both pool and premier customers.
- 1 : currently riding a premier passenger
- 2 : currently riding a single pool passenger and can accept another pool passenger (priority will be given to it in case a person books a pool cab)
- 3 : filled pool cab

Riders :

Implemented using struct `**riders[]**` and for each of the M riders, a new thread is created.

- Rider's arrival time is stimulated using sleep.

- Arrival time is a random number between 1 and 50.

- Wait time is a random number between 1 and 8.

- Ride time is a random number between 1 and 8.

- If the rider is pool, it searches for a 2 type cab and if it does not find any it searches for wait cab.

- If he does not find any cab he sleeps for a second and repeat the same search until it finds a cab or times out if current time exceeds his **max_Wait_time**(Note: max_Wait_time does not include time taken by payment server).

- After getting a cab he sleeps for ride time and then frees the cab.

- It then invokes **Kiraya_Lo()** function and searches for a free server. On finding a free server, it activates that server(changes status from 0 to 1).

Payment Servers :

Implemented using struct `**Pservers[]**` and for each of the k servers, a new thread is created.

- It waits for it to be activated. status to change from 0 to 1.

- It then sleeps for 2 seconds

- Payment is then accepted. status is changed from 1 to 0.

- It then decrements the variable total_no_of_riders(keeps track of remaining riders).

- If no rider is left, all functions return and all threads are killed.