Assignment 3

Process Synchronization

Due: Wednesday October 30 by 11:55pm.

Instructions:

This programming assignment has three parts. You have been provided a Makefile, and source files for each part in the attached zip file "**ProgAssignment3.zip**". Make sure that you go through the comments in the source files carefully, and print the output in the described format. Otherwise, you will get a 0 in your assignment.

You have to use the pthread library. Make sure that you do not change main file or any of the function headers provided in part1.h, part2.h and part3.h. However, you can make use of any extra helper functions you want to make.

For synchronization purposes, you will use semaphores. Note that you are not allowed to use any other construct (e.g. mutexes, condition variables etc.).

For compiling your program, run the command "make". For running your program, type ./test.

Question 1:

You've been hired by XYZ University to build a controller for their elevator system. The university has two elevators. For simplicity, one elevator (Elevator 1) starts from the ground floor and goes to the top floor and then comes back to the ground floor (and so on). The other elevator (Elevator 2) starts from the top floor and goes down to the ground floor and then again goes up (and so on). Though, the system will work fine with one elevator but we do not want people to keep waiting for/ in the elevator therefore, the university has two elevators. You must design the system which will use both the elevators.

The elevator user is represented by a thread – when the user gets to the elevator, the user calls "GoingFromTo(int fromFloor, int toFloor)". The thread shouldn't return until the elevator has gone from the fromFloor to the toFloor. For simplicity, you can assume each elevator holds only 20 people.

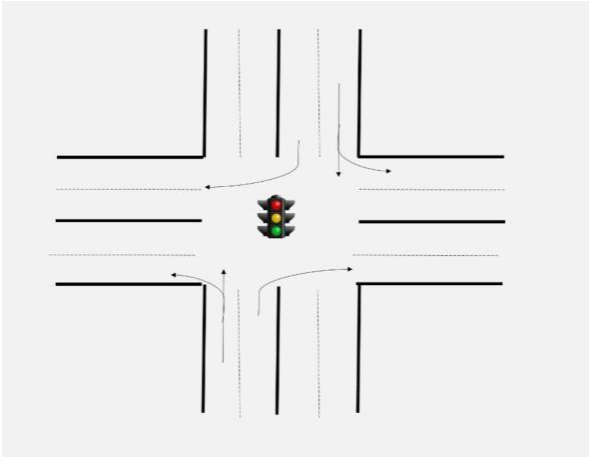
Note that the test cases might pass with only one elevator. But a system with two elevators take lesser time than a system using only one elevator. Your submission will be checked with extra (hidden) test cases for this purpose. So, use both the elevators.

Question 2:

You have been hired by the Metropolitan Transit authority (MTA) to design controller of their automated trains around the city in a circular loop. The stations in this loop are numbered 1 through N. Each station has a boarding area where passengers wait for the train in a queue. To enter the boarding area, passengers need to scan their ticket on an automated machine. Similarly, just before entering the train passengers scan their tickets on a 2nd scanning machine. The scanned information includes the source station and destination station of the passenger. Suppose that the capacity of each train is 50 (i.e., only 50 passengers can be in the train at any given point in time). There are 5 trains running in the loop with an inter-arrival time of 5 minutes. When a train arrives at the station passengers waiting in the queue board the train. If the train becomes full, it does not take further passengers and the remaining passengers in the queue wait for the next train. A train stops at a given station only, if any of the following two conditions hold: There are one or more passengers waiting for the train in the boarding area of the given station. There are one or more passengers who need to get off from the train at the given station (i.e., these passengers have declared the given station as their destination station). Each passenger is represented by a thread. The passenger thread shouldn't exit until the passenger has reached the destination station.

Question 3

You have been hired by Uber to design the operating systems of their automated cars running in the downtown area. Specifically, Uber has asked you to design a system that enables their cars to safely cross intersections with traffic lights. This system includes a traffic light controller and car controller. As depicted in the Figure below, suppose that all intersections in the given downtown area have traffic lights controlling 4-way traffic. The traffic flow is North-bound, South-bound, East-bond, and West-bound. Each road on the intersection has 4 lanes (2 lanes for North-bound traffic and 2 lanes for South-bound traffic) or (2 lanes for East-bound traffic and 2 lanes for Westbound traffic). A car arriving at the intersection may go straight, or turn left or right. Following rules need to be followed for crossing the intersection.



1. There are four traffic lights, one for each of the North-bound, South-bound, East-bound, and West-bound Road. Only one traffic light becomes green at any given time; the remaining traffic light will be red.
2. A car going straight or turning left must be in the left lane.
3. A car turning right must be in the right lane.
4. A car going straight or turning right cannot cross the intersection if the corresponding traffic light is red and must wait for the traffic light to go green. The traffic light for going straight or turning right becomes green at the same time.
5. A car may turn left on Red light if there is no car waiting in front of it for green traffic light.
6. A traffic light remains green until 5 waiting cars from each lane cross the intersection or turn right. In case there are less than 5 waiting cars in each lane, the traffic light becomes red as soon as all the waiting cars cross the intersection or turn right.
7. Each car is represented by a thread. The car thread shouldn't exit until the car has crossed the intersection or made the turn.

Submission Instructions:

1. Please follow the print format as stated in the code files.
2. You only need to submit cpp and header (.h) files for all three questions. There will be total 6 files.
3. Make a zip file containing your submission files and name it A3_<R>.zip where <R> is your roll number e.g A3_20100192.zip
4. Failure to comply with the submission protocol can result in 0 marks.