Programming Assignment #04

Due: Thursday, 12/06/2018, by 11:30pm

You work as a system engineer team for Universal Studios, which is currently in the process of designing a Jurassic Park ride at the theme park. You are in charge of designing the waiting area, God-forsaken land of ropes, where people going around in circles. Here are specifications:

- The ride consists of putting MAXPERCAR or less people on a Ford Explorer and sending through a small-scale version of Jurassic Park where the dinosaurs run loosely and attack tourists.
- The process is semi-continuous, as an empty Explorer will arrive at the embarkation periodically. You have been told by industrial engineers that it takes 7 seconds to load the people into the car and to send them on their way and another 53 seconds to complete the ride.
- But not every car is completely full, as the number of people waiting on line may be less than MAXPERCAR.
- The en-queuing rate is variable, as people get in line at various rates during the day. You can assume all people arrival at beginning of every minute. The mean value of arrival rate varies as follows:
  - 09:00:00--10:59:59, meanArrival = 25 persons per minute
  - 11:00:00--13:59:59, meanArrival = 45 persons per minute
  - 14:00:00--15:59:59, meanArrival = 35 persons per minute
  - 16:00:00--18:59:59, meanArrival = 25 persons per minute
- At the beginning of every minute, we can decide the number people who will arrive in this minute by calling poissonRandom(meanArrival), provided in file random437.h.
- Assume there is a limit on the number of people kept in the waiting area, MAXWAITPEOPLE(800). If more people come, they will be advised to come back later since the waiting line is too long.
- You are going to design the waiting area where the lines will snake back and forth endlessly between roped-off guideways, before spilling over into the street. Your task is to simulate the waiting lines on a computer with multiple concurrent threads to observe the dynamics of the lines at different times during the day.
- There are total CARNUM Explorers, you need to create a new thread for each of them.
- You need a thread to take care of incoming people and decide that either accept them into the waiting line or reject some or all of them away. It will write a status line into an output file:

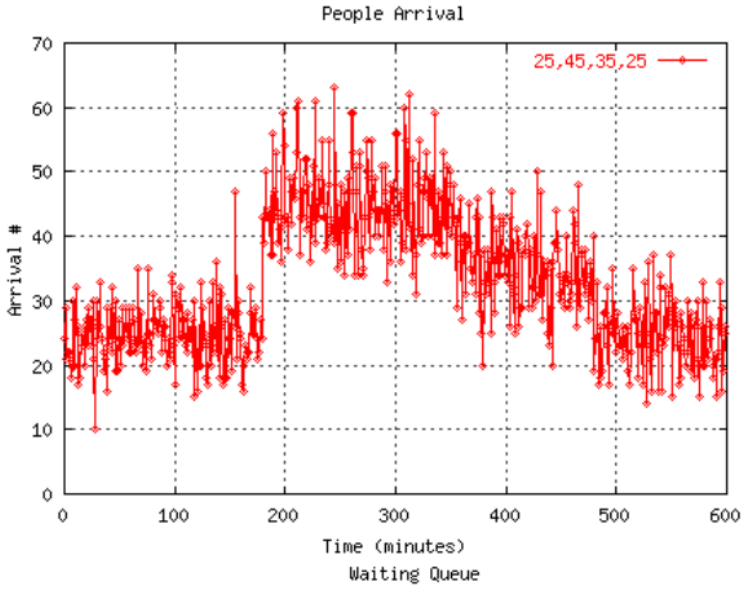
  "XXX arrive YYY reject ZZZ wait-line WWW at HH:MM:SS"

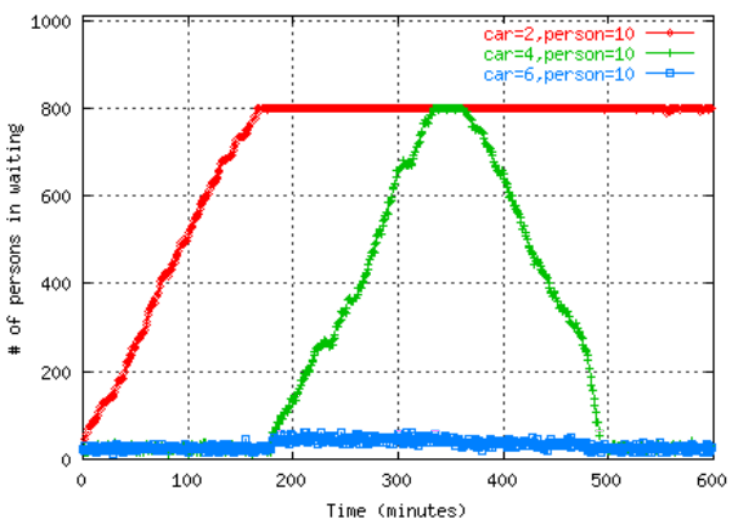
  where, XXX is the time step ranging from 0 to 599, YYY is the number of persons arrived, ZZZ is the number of persons rejected, WWW is number of persons in the waiting line, HH:MM:SS is hours, minutes, and seconds. At the end of the day, it will display:
  - The total number of people arrived
  - The total number of people taking the ride
  - The total number of people going away due to the long waiting line
  - The average waiting time per person (in minutes)
  - Determine the length of the line at its worst case, and the time of day at which that occurs.
- All of your threads need to synchronize every minute.
- You should C to implement your assignment.
- Your implementation should not have any thread doing busy waiting.
- Your program will take two options, "-N" CARNUM and "-M" MAXPERCAR, as its command line arguments, and run simulation from 9am to 7pm.
- You need to simulate 10 hours (600 minutes) between 9:00am – 7:00pm. You could synchronize a virtual minute to a real second (or 10ms or 100ms).

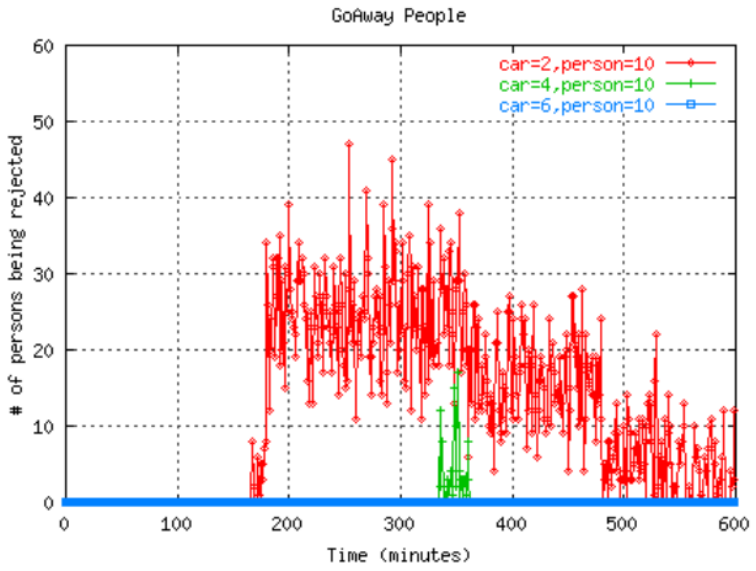
- Run your simulation with MAXPERCAR M=7,9, and CARNUM N=4,6. (We may use different parameters to test your code.)
- From your output, fill out a table

(N,M)	Total Arrival	Total GoAway	Rejection Ratio	Average Wait Time (mins)	Max Waiting Line
N=2,M=7					
N=2,M=9					
N=4,M=7					
N=4,M=9					
N=6,M=7					
N=6,M=9					

- Provide three figures to illustrate for CARNUM=2,4,6, MAXPERCAR=7:
  - The number of persons arrived in every minute. (one curve in one figure)
  - The number of persons reject in every minute. (four curves in one figure)
  - The number of persons waiting in every minute. (four curves in one figure)
- For example, three sample figures are illustrated here (gnuplot is used to generate figures; here, you can use any software you like as long as x y axis and legends properly displayed):







- You have to submit to files, i.e., write report and source code
  1. The write report should include:
     - Description (in pseudo code) of your design of thread coordination of the three parts: the waiting line, explorer cars and the arriving module (One page in length approximately), including major primitives used for thread synchronization and critical section protection;
     - The required table;
     - The three figures to demonstrate your outcome;
  2. The source code should be in one file. If you would use random437.h without change, we will have it in place