Code Comments

Your code is split into multiple modules, which is good. Your headers file need to have #include guards. In util_queue.h, you have a suitable structure for a queue. It would be slightly easier on the eye if your function prototype names, for example, all line up in the same column. I can see that you're using the GSL random number generator. In util_queue.c, good error processing if malloc fails. The names push and pop are normally associated with stacks and not queues, but that is fine. You free memory when you take stuff off the queue. You don't seem to have a function called runOneSimulation, but looking at runSim, there is a chunk at the beginning of the for loop that changes the lights. Note that you only really need one variable to record the lights, there are only two lights and the colours are only red and green. You have got a nice comment to say what that does, but why not put it all in a function with a suitable name and then just call the function? If the lights do change, then nothing else should happen within this iteration. Again, you have got you have got two little chunks of code here that essentially do the same things, but just for left and right, so perhaps have a function there and just call it twice, once for each side. And the same again for traffic leaving the queues. Then you have a separate while loop that repeats some of that code just to empty the queues after the end of the 500 iterations. So why not put all of that inside the main loop? And all that is required then, is an extra if statement somewhere. Good commenting, though. In runSimulations.c, Your program does not accept values from the command line. You're prompting the user to enter for values using scanf, but you need to check that scanf returns the right number (which you might be doing - not sure). Even there, you need to verify that, for example, the traffic light period is not zero, because if it is, then your simulations are going to be strange. You seed the random generator just once at the beginning, which is good. You run your simulation 100 times and then you output the results. That is nice. So some repetition of code, which could be replaced by nicely named functions that would cut down the amount of code that is in the main simulation loop and make it slightly easier to read, but good use of comments. Well done.

Report

In the design choices section, the assumptions should really be about technical assumptions you have made in order to write the program, so bicycles are not really relevant here. What is important (and missing) is how you have chosen to use the input parameters to decide whether a car arrives or not. The experiments are interesting, and I like that you have included graphs.

Marks: 79%