You are going to implement one version of the program covered in the video lectures, involving ticket sellers. Represent the number of sellers and total number of tickets as global variables that are easily changeable (in his lectures, there were 10 sellers and 150 total tickets). Note that, in the video lectures, there are three versions of the program covered. The first version does not use concurrency, so each seller sells 15 tickets all at once before the next seller sells any tickets. The second version uses concurrency, and the ticket sellers appear to sell in parallel; however, it is still assumed that each seller sells the same number of tickets (again, 15 each). In the third version, each seller keeps on selling tickets until all of the tickets are gone, so it is likely that the different sellers will sell different numbers of tickets. This version requires a semaphore. You are going to implement this third version of the program, which was discussed at the end of lecture #15 and the start of lecture #16.

Of course, as we discussed at our meetings, the code he covers in class assume access to a particular library that was created for that specific class. The library allows programming to create and instantiate multiple threads, and it also provides semaphores, including a wait and a signal routine. We do not have access to that library, and even if we did, we should not expect it to work on our systems. You will need to find, learn, and use a library that works for whatever programming environment you are using. I will implement the program myself using Ubuntu and gcc, and I will use the POSIX library for threads and semaphores. If you use another library, I may not be able to run your code, and I will need screenshots of your running program when you submit it.

When you are finished, email me your code to [carl.sable@cooper.edu](mailto:carl.sable@cooper.edu). If you do not use the POSIX library, also send me screenshots of the running program. Once I grade this, I won't allow re-submissions; so, if you want to talk to me about your code before submitting it, set up an appointment with me. Of course, you are also welcome to email me questions related to the assignment.