This project was about implementing a First-Come-First-Served (FCFS) Scheduler and Round-Robin (RR) Scheduler. The FCFS scheduler was fairly simple to implement. All that was needed was the start time to be checked. I used an array to store all the jobs. The program read all the jobs from the file and stored it in the array. In FCFS the program uses an iterator to parse through the array. Whenever a job has been finished, the iterator is incremented to the next job. The next job only runs if the timer reaches the start time. Otherwise, a space is left. The Round-Robin was a bit trickier since an iterator could no longer be used. Instead, I used the duration of the job as an iterator by creating a new field in the Job struct called remaining time. I decremented this value each time the job was scheduled to run. I used a queue struct in order to facilitate queue behaviors and to properly order how the jobs would be scheduled. The simply needed to take in the job index as I would still access the job through the jobs array, so whenever a job was going to run at that time, I would dequeue the index from the queue and use it to print out an X for that job. Then I would check if the job at that index had any remaining time to run, and if so, the index would be enqueue back into the queue, otherwise, I decrement a counter which stored the number of jobs. This counter was used so that I don’t exit my while loop for the Round-Robin early. This was the biggest hurdle. The while loop always checked if the queue was empty and if it wasn’t it would continue. But there are times when the queue is empty but there are jobs that are going to be scheduled, they just haven’t arrived yet. So, the counter was used to know if all the jobs had run fully. Once the counter reached 0, all the jobs were run fully, and the program could exit the while loop. I learned about job scheduling through the project and more importantly, how to implement a scheduler and see visually, the difference between FCFS and RR. It was interesting to also see how easy it was to print out everything. The project document gave a recommendation to use some java package for graphics but with C, a few well-written lines of code gave a pretty good visualization of the scheduling. Overall this project was a fun and easy project.