**Scheduling Algorithms**

* First Come First Serve
  + FCFS didn’t take a lot of effort to put together and was rather straightforward. My implementation ended up just finding the smallest non-flagged (not yet finished) process and would then run it to completion.
* Shortest Remaining Time
  + SRT required a little more thought as I had to consciously keep track of which processes had already arrived. I ended up only needed an additional condition to make sure the process had arrived before searching for the one with the shortest time until completion.
* Round Robin
  + RR was confusing at first but once I wrote down a few examples and understood how we were supposed to implement it I managed to put it together quite quickly. I ended up just moving throughout the array of processes and taking QUANTUM off of their remaining time as long as they had already arrived. Once QUANTUM brought the process to completion I would then take the overage and add it to time so the time would adjust properly.
* Round Robin – Priority
  + RRP took me the longest to figure out. While functioning the same as RR initially, I needed to decide how to handle priorities as they arrived. My solution revolved around looping through the arrived processes, and if none of them could currently be run then I would decrement the priority. Once the time advanced I check if any of the newly arrived processes have a higher priority and if they do, I raise the priority.