For this project, we used Dr. Ligon’s code for the base of this project, as we were more confident in that than our own. Most of the changes that were made for this part were in the switch’s cases. They all function similarly in that each case modifies the event type to move to the next case. The arrival case passes the passenger into the airline queue and creates an additional passenger as long as the program has not reached the maximum number of passengers. To handle airlineQ, idQ, and trainQ, we created a dummy variable for each that allows us to check if someone is already at the desk for that event type. If there is, we place it into a FIFO queue, where it will wait for the desk to be cleared. As the desk is cleared, it calls for the next passenger from the queue. If there is no passenger in the queue it says the desk is empty and moves on to the next case.

The scanQ case is somewhat similar to the others, but with an additional component. Instead, there is an array of dummy variables that allows us to see if there is a passenger going through each scanner. We created an additional parameter on the passenger structure that allowed us to see which scanner was being used by that passenger, so that we could open up that specific scanner when the passenger leaves. If all the scanners are being used, or there is a queue already, we check to see which queue is the smallest of the available queues and place the passenger in that queue to wait for the scanner to open. Once one of the scanners opens, a passenger from that scanner’s queue is pulled out, processed, and pushed onto the train queue.

The gate case is where the passenger leaves the airport. At this step, the final time is reported, and the passenger is passed into the passenger\_destroy function to be freed from memory. After the last passenger has left the gate, we exit the loop and free up all the used memory so that there are no memory leaks.

For the extra credit, we implemented the parse\_args function using getopt so that the program will accept command line arguments. We use -p for number of passengers, and -s for the MAX\_SCAN value.