Problems section of the report:

In this lab we started by separating our work into different sections of the lab. Our initial idea was to have each group member do a section of the code and someone work on the assembly. This division of labor works well but had a few issues. The first issue came when some parts of the code were made before the others, which led to temporary pseudo code and mistakes. This had to be almost completely rewritten to fit the assembly and digital display portions of the lab. This issue was easily fixed since the framework was made prior, but the actual code needed improvements, so our group put the code together and fixed all the issues with the code including how to call the assembly that turns the LEDs on and off. The next challenge we had during this lab was with the Digital display. Initially we were confused that the 7-segment display had only one shift register. We began troubleshooting with the driver, but our main problem persisted. We got one digit to display properly with interrupts but needed double digits for the assignment. In the end our solution stemmed from the elegoo instructions using all the same pins. The code part was solved by using interrupts and the built in timer. The timer was used to trigger an interrupt every second and decrement another variable used to count the seconds down to then display on the 7 segment digital display. The other major issue we had was during our brainstorming process earlier into the project, we initially made the entire project minus the digital display without interruptions integrated into the code. The LEDs all worked properly with the correct timing, however there was no code for interrupts between the traffic lights for when they switch. Instead, it was made purely from timing commands. This definitely helped showcase that the general system was working properly, however the interrupts are a bit part of the lab and led to a lot recoding and changing the general set up of the whole system. We fixed the timing issue by adding some initial interrupts using the “\*” key to poll for the keypad inputs. The final issue we had to overcome was our keypad wasn’t reading in the data properly. We created the keypad using a 2D array set up like the keypad in real life. The original design had an function for reading in the data, this function as the crux of the data reading issue. The original function called a character variable to get the input from the keypad, then checks an if statement to see if the variable is assigned a value. If that’s true, the function checked if the first character was “#” indicating the start of the keypad sequence. The flaw in this code was that the whole function had to loop to get the full command, rather than have a loop within the function. Without the loop it only found the first character and checked if it was the start, this would be checked each time the function was called, which made it so the only time the code would go through was for the “#” symbol rather than the whole sequence. However, in the new code we initialize the character variable, and set up a infinite while loop. In the loop it sets the variable to the read value from the keypad, then returns the character as a string. This works now because the code isn’t looking for the first character in the read function, instead it looks for it later in the code when it needs to take that data and convert it to different states of the lights.

Personal Contribution:

I originally was given the task of making the keypad, however our jobs got mixed up so someone else took that job. After that confusion I was tasked with making the main function. Since this was early in the project, I had no idea how to implement some of the code, and some of the other parts weren’t finished yet. So instead, I made my best attempt at the code. It didn’t include interrupts but rather based off time and input data. When I didn’t know how to call things properly, I added pseudo code so it could be used as a template for how the process would go. Overall, my code was very flawed, but I provided a skeleton for the future code renditions. I was also in charge of writing the problems section of the report.