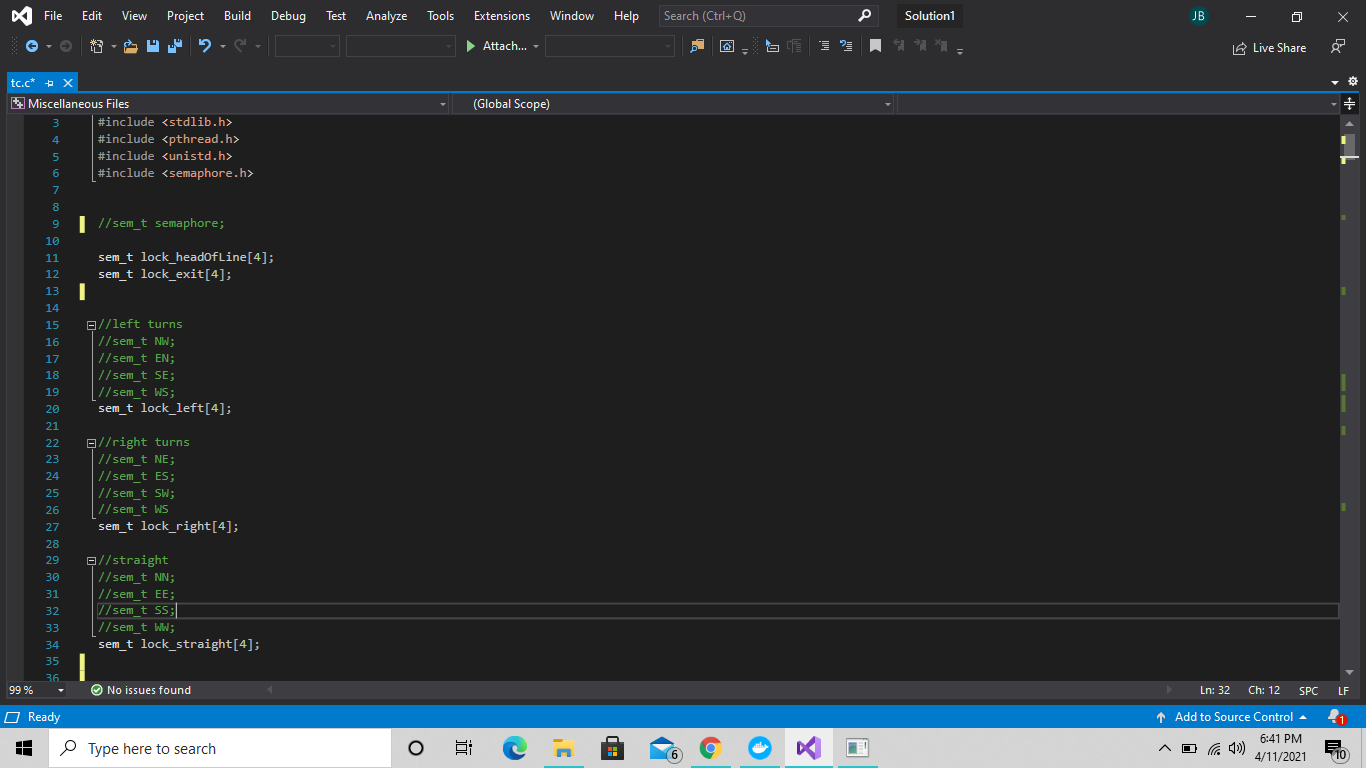
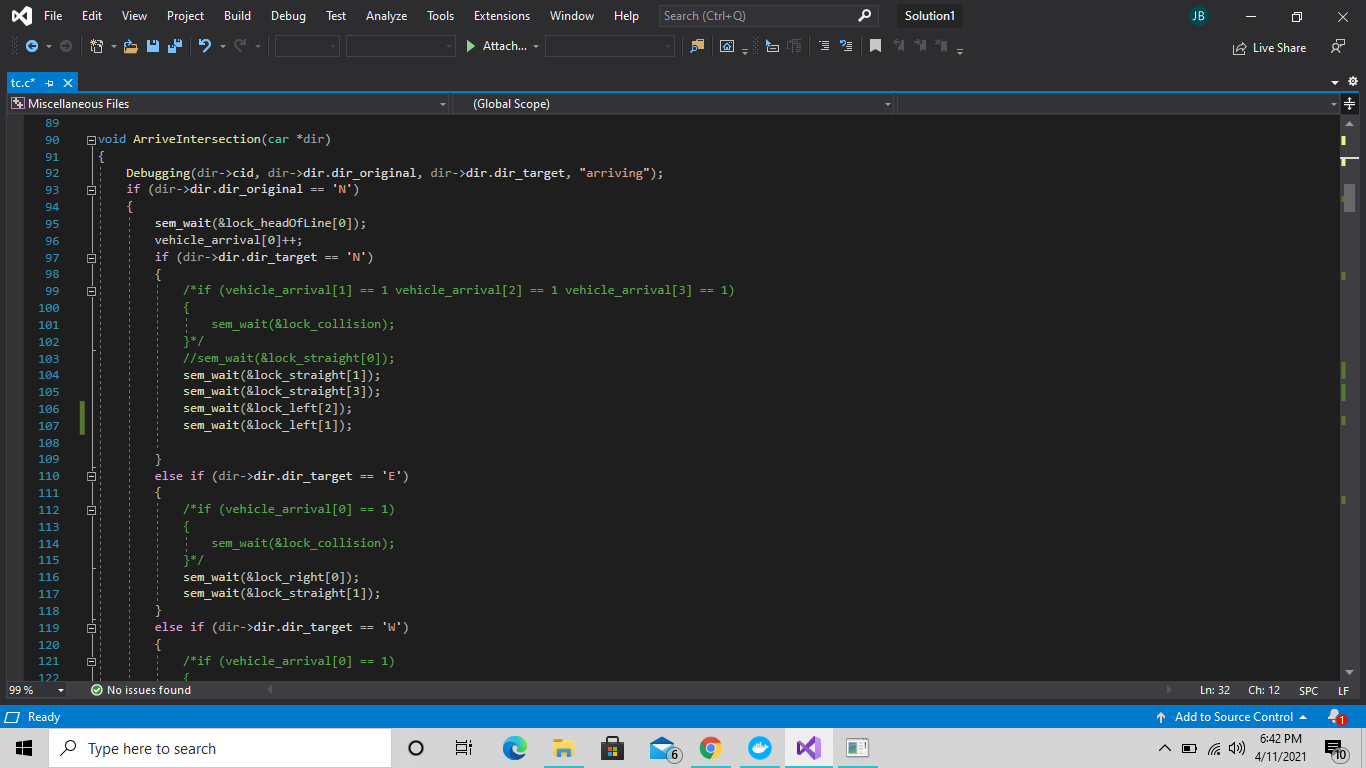


To simulate the intersection we used a system of if statements for each starting direction and each possible direction they can go from there. For example if you started in the north lane you could either go east (right), north (straight), or west (left). In order to prevent a collision we used multiple semaphores for each possible direction the car could go. We have a semaphore for a car first arriving at the intersection (headOfLine) (N, E, S, W) one for exiting the intersection (exit) (N, E, S, W), one for each straight combination (NN, EE, SS, WW), one for each possible left turn combination (NW, EN, SE, WS), and finally one for each right turn combination (NE, ES, SW, WS). Using these all we had to do was lock the semaphore that would result in a collision of the arriving car while it was crossing. We then unlock these semaphores when we reach exit in our subroutine.

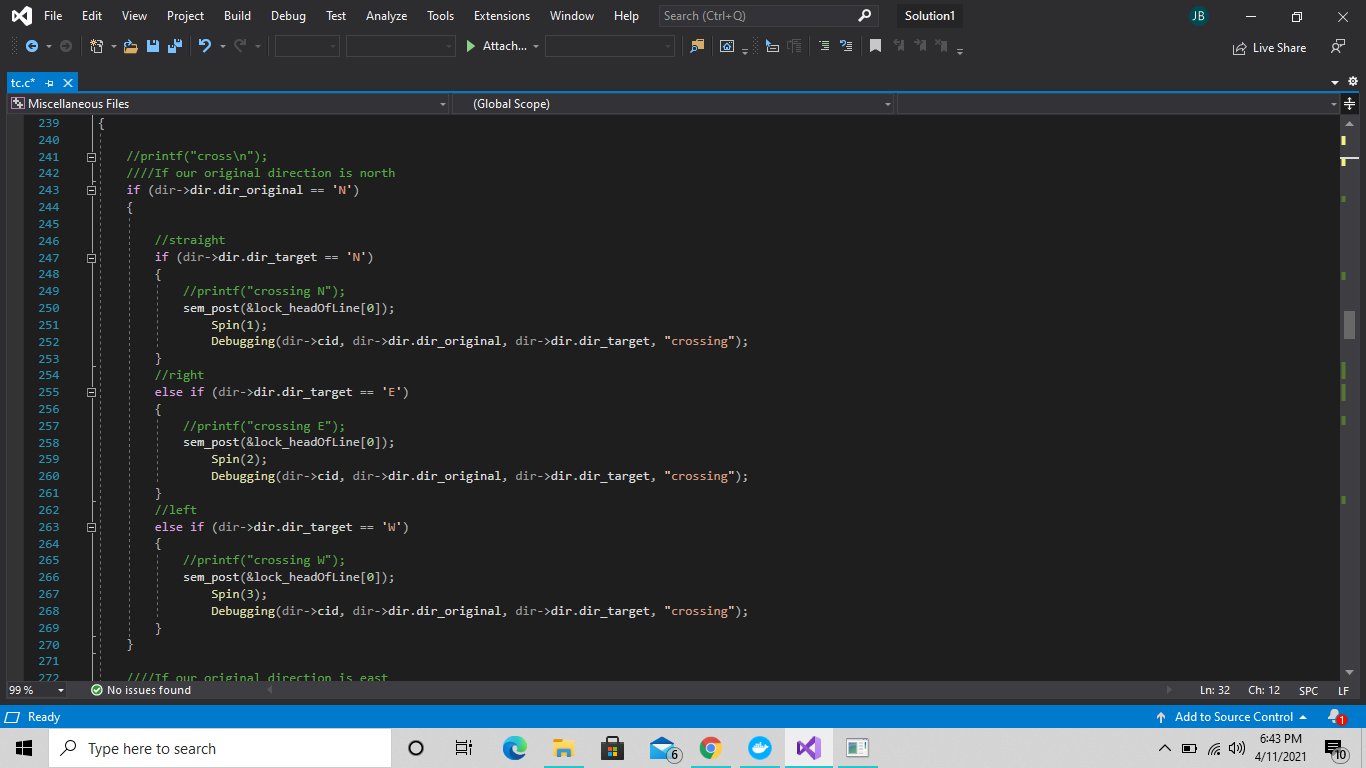
Each car is represented by a different thread. We create an array of cars and use a for loop to create a new thread for each car. Each thread then runs through our subroutine.



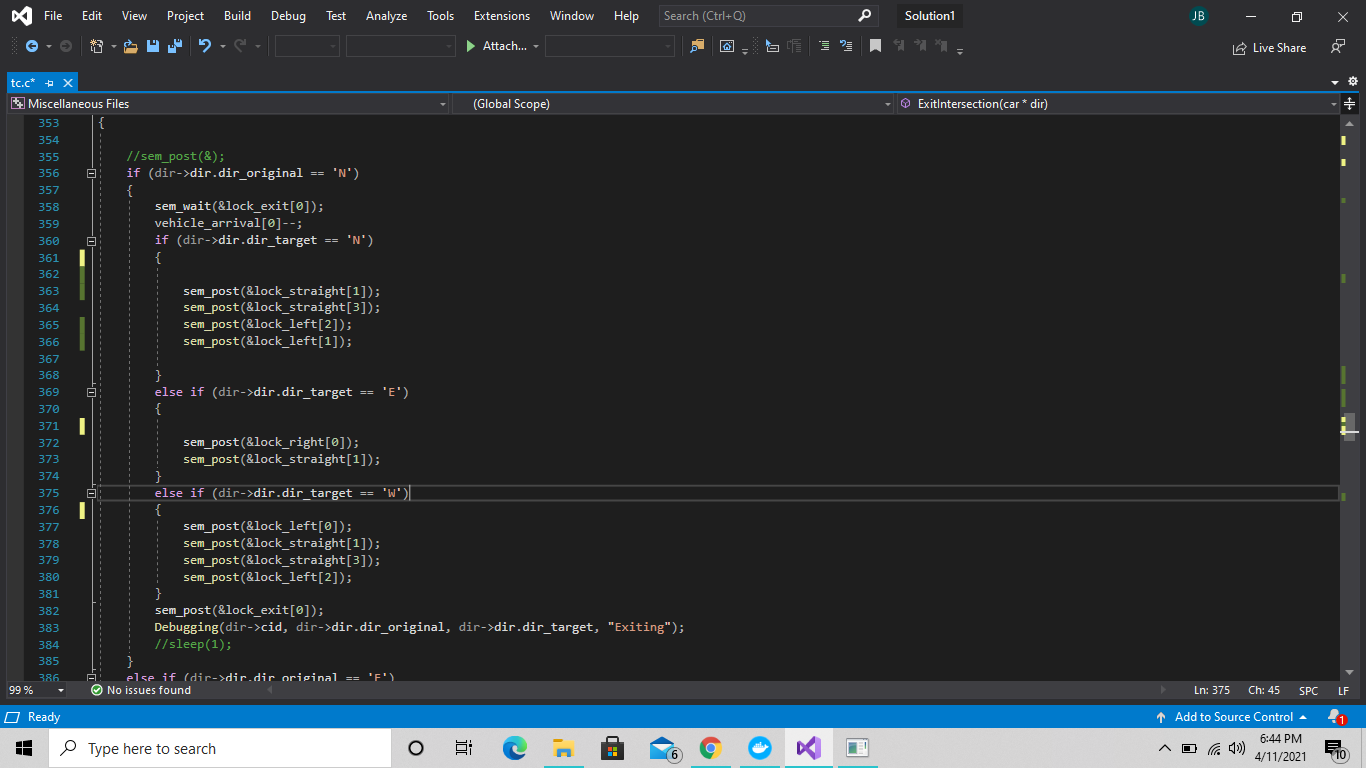
An idea of the semaphores we used



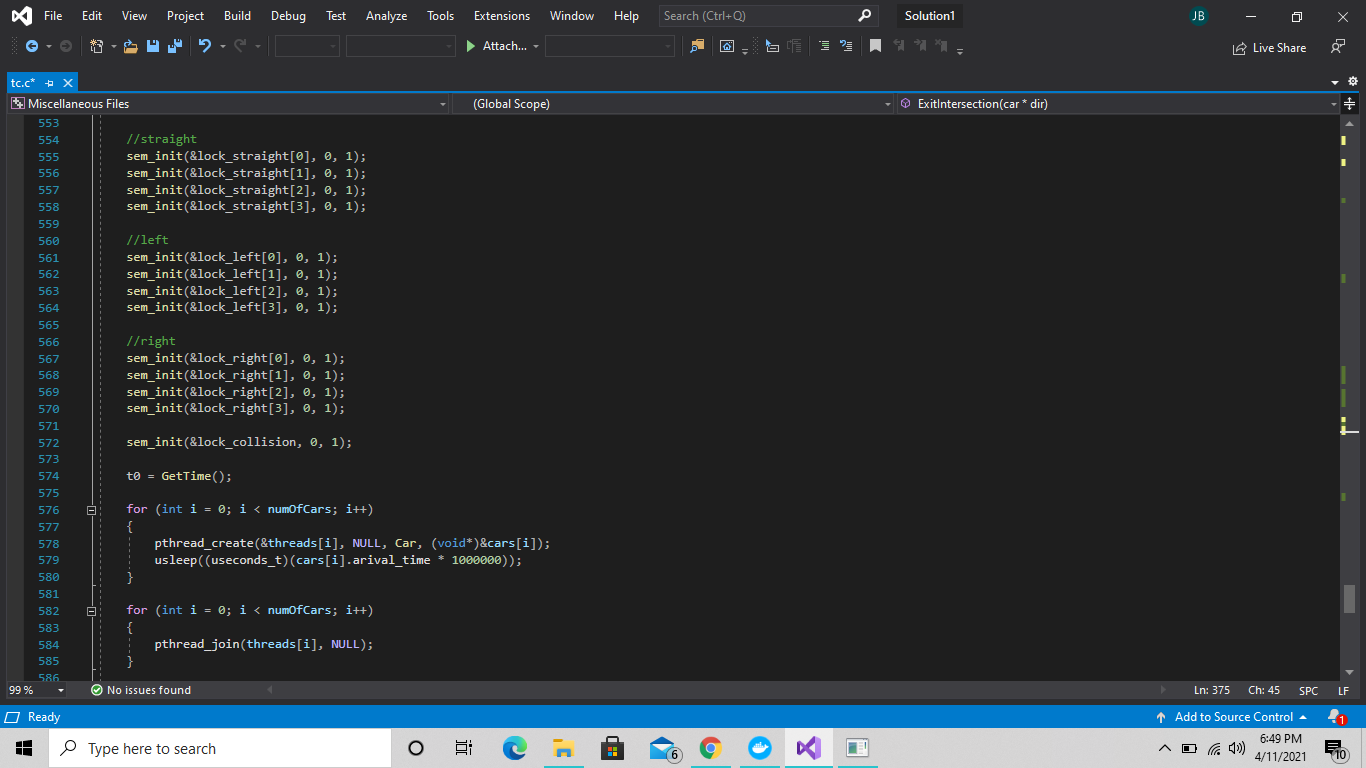
An example of how we are locking semaphores when cars arrive



Once a car begins to cross we can unlock the head of line lock, we then spin to simulate crossing



When we exit we unlock our semaphores



Example of us creating our threads, as well as initializing our semaphores.