Connor Palmira

The results of my program are similar, but not quite the same as the example results due to the nature of randomness. My program does follow the same format as the example results, so if the example and my program had used the same random values then the output would have been the same.

For Project #5 B, the purpose of using semaphores was so that only some of the threads would continue at a given time. The project calls for either the North and South directions to be continuing or the East and West directions, but never all of them at the same time. Therefore, the purpose of the semaphores was to make sure that the threads for North and South would not continue at the same time as the threads for East and West, and vice versa.

The role that semaphores had for Project #5 B was to control the “lights” of the virtual intersection. It simulated a real intersection (without turning cars) and would have two opposing directions (i.e. North and South) turn green, then yellow, and then red before switching to the other opposing directions. It also controlled the threads, “vehicles” in this scenario, through semaphores to make sure that vehicles going North could only go when the light was green.

If the light was green and there were no vehicles approaching, then the program would be continuing until either there were vehicles approaching from that direction or the light would turn to red for vehicles of the other directions to go. For example, if the North and South lights were green but all of the waiting vehicles were waiting for the East and West directions, the program would just keep going, letting vehicles that might show up from the North and South through, but eventually turning yellow then red for the other directions.

Similarly, if the light was red and there were many cars waiting for the light to turn red, the threads would be stuck in an infinite loop waiting for the light to turn green. Once the light would be green, they would pass through the intersection.