

ECES 338 Assignment #3

Due: February 24, 2000

Spring'2000
Ozsoyoglu, G.

(100 points)

In this assignment, you will work semaphores and the conditional critical region statements. Please start the assignment right away. Expect to spend more time for question #3. For your algorithm, you can use any pseudo-code (or C or C++) syntax. It is OK to discuss the algorithms with your friends. However, when you start writing your algorithm, you should do it alone.

(30 points) **1)** Cars coming from the north and south must pass a bridge across a river. Unfortunately, there is only one lane on the bridge. So, at any moment, the bridge can be crossed only by *one or more* cars coming from the same direction (but not from opposite directions). For liveness, the direction of traffic across the bridge changes each time 10 cars have crossed from one direction while one or more cars are waiting to cross it from the opposite direction. Give an algorithm for cars (processes) to cross the bridge *using* semaphores.

(30 points) **2)** Solve question 1 using conditional critical region statements (instead of semaphores).

(40 points) **3)** A multiple semaphore allows the wait and signal primitives to operate on several semaphores simultaneously. It is useful for acquiring and releasing several resources in one atomic operation. Thus, the semantics of **atomic** wait and signal primitives (for two semaphores S and R) can be defined as follows:

```
wait ( $S,R$ ) :   while (  $S \leq 0$  or  $R \leq 0$  ) do no-op;  
                  $S := S - 1$ ;  
                  $R := R - 1$ ;  
  
signal ( $S,R$ ) :  $S := S + 1$ ;  
                  $R := R + 1$ ;
```

Please note that we also allow regular wait (R), wait (S), signal (R) and signal (S) primitives on the two semaphores R and S --except that we now define the **atomic** primitives wait (S,R) and signal (S,R) with the above-listed semantics.

Show how a multiple semaphore can be implemented using regular semaphores and shared variables.