1. What is the difference between binary and counting semaphores?

Binary only has 0 and 1 while counting can have more than 2 values

2. What are the operations that can be performed on a semaphore. Explain in brief about each operation and what it does in your own words.

initialize it and give it a count

sem post/signal
count up to max, sem wait
waits for count to be >0
before continuing and decrements
it.

5.6. Consider the following processes P1 and P2 that update the value of the shared variables, x and y, as follows:

```
Process P1 :                    Process P2 :
(  performs the operations:     (  performs the operations:
      x := x * y                      x ++
      y ++                            y := x * y
)                               )
LOAD R1, X                      LOAD R3, X
LOAD R2, Y                      INC R3
MUL R1, R2                      LOAD R4, Y
STORE X, R1                     MUL R4, R3
INC R2                          STORE X, R3
STORE Y, R2                     STORE Y, R4
```

Assume that the initial values of x and y are 2 and 3 respectively. P1 enters the system first and so it is required that the output is equivalent to a serial execution of P1 followed by P2. The scheduler in the uniprocessor system implements a pseudo-parallel execution of these two concurrent processes by interleaving their instructions without restricting the order of the interleaving.

a. If the processes P1 and P2 had executed serially, what would the values of x and y have been after the execution of both processes? 7, 28

b. Write an interleaved concurrent schedule that gives the same output as a serial schedule.

c. Write an interleaved concurrent schedule that gives an output that is different from that of a serial schedule.

```
P1: LOAD R1, X
P1: LOAD R2, Y
P1: MUL R1, R2
P1: STORE X, R1
P2: LOAD R3, X
P2: INC R3
P2: LOAD R4, Y
P1: INC R2
P1: STORE Y, R2
P2: MUL R4, R3
P2: STORE X, R3
P2: STORE Y, R4
```

```
P1: LOAD R1, X
P2: LOAD R3, X
P2: INC R3
P1: LOAD R2, Y
P2: LOAD R4, Y
P1: MUL R1, R2
P1: STORE X, R1
P2: MUL R4, R3
P2: STORE X, R3
P1: INC R2
P1: STORE Y, R2
P2: STORE Y, R4
```

3.

4. The following three functions are run on a shared processor by three processes. They can coordinate their execution via shared semaphores that respond to the standard signal(`sem_signal()`) and wait(`sem_wait()`) procedures. In order to produce the output HELLO, add respective `sem_signal()/sem_post()` and `sem_wait()` comands in the code. Create your own semaphores as needed.

- Is printing HELLO possible
- Number of semaphores - 2
- Names of semaphores - a, b
- Initial values of sempahores - 0

Function#1	Function#2	Function #3
`print("H")`	`print("L")` sem wait (a)	`print("O")` sem wait (b)
`print("E")`	`print("L")`	
sem post (a)	sem post (b)	

終わる