1. Binary semaphores can only be 0 or 1, to block or allow a thread/process from accessing a resource. A counting semaphore can be any non-negative integer and is used when there are multiple resources available. Further to show how many of the resources are available so that multiple processes can potentially access the resource at the same.

2. Operations are either signal or wait. A wait operation attempts to decrement the value to block the process. A signal operation increments the value potentially unblock the waiting process.

   a. x = 7, y = 28

   b.

   ```
   load r1, x
   load r2, y
   mult r1, r2
   store x, r1
   inc r1
   store x, r1
   load r4, y
   inc r4
   store y, r4
   load r3, x
   mult r4, r3
   store y, r4
   ```

   c.

   ```
   load r1, x
   load r2, y
   mult r1, r2
   store x, r1
   inc r1
   load r4, y
   inc r4
   load r3, x
   mult r4, r3
   store y, r4
   ```

3. Yes it is possible to print Hello.
   Names: s1, s2
   Initial value: 0,0

```
Function #1         Function #2         Function #3


print(“H”)          sem_wait(s1)        sem_wait(s2)
print("E”)          print(“L”)          print(“O”)
sem_signal(s1)      print(“L”)
                    sem_signal(s2)
```