

1.1 Show a sequence (i.e., trace the sequence of interleavings of statements) such that the statement "x is 10" is printed.

```
P1: x = 10; (P1 initializes x to 10)
P2: x = 10; (P2 initializes x to 10)
P1: x = x - 1; (P1 x becomes 9)
P2: x = x - 1; (P2 x becomes 9)
P1: x = x + 1;
P2: x = x + 1;
P1: if (x != 10) printf("x is %d", x);
P2: if (x != 10) printf("x is %d", x);
P1: x = x - 1;
P1: x = x + 1;
P2: x = x - 1;
P2: x = x + 1;
P1: if (x != 10) printf("x is %d", x);
P2: if (x != 10) printf("x is %d", x);
P1: if (x != 10) printf("x is %d", x); (the printf statement is executed, output:
"x is 10")
```

1.2 Show a sequence such that the statement "x is 8" is printed.

```
P1: x = 10; (P1 initializes x to 10)
P2: x = 10; (P2 initializes x to 10)
P1: x = x - 1; (P1 x becomes 9)
P2: x = x - 1; (P2 x becomes 9)
P1: x = x + 1;
P2: x = x + 1;
P1: if (x != 10) printf("x is %d", x);
P2: if (x != 10) printf("x is %d", x);
P1: x = x - 1; (P1 x becomes 9)
P2: x = x - 1; (P2 x becomes 8)
P1: x = x + 1; (P1 x becomes 9)
P2: x = x + 1; (P2 x becomes 10)
P1: if (x != 10) printf("x is %d", x); (printf statement is executed, output: "x is
8")
```

2. What is the difference between binary and general semaphores?

Binary semaphores have two states, 0 and 1, and are used for mutual exclusion, while general semaphores have a non-negative integer value and are used for controlling access to a set of resources.

3. What is a monitor?

It is a synchronization construct that allows multiples process to safely access shared resources and also provides a higher-lvl abstraction compared to semaphores.

4. What operations can be performed on a semaphore?

Semaphores can perform wait (P or Down) operations that decrements the

value of the semaphore or they can perform signal (V or UP) operations that increments the value.