Q1. What is mutex and semaphore?

⇒ • Mutex :-

- Mutex is used to ensure only one thread at a time can access the resource protected by mutex.
- Mutex is good only for managing mutual exclusion to some shared resource.
- Mutex is easy and efficient for implementation.
- Mutex can be in one of two states :- locked / unlocked.
- Mutex is represented by one bit. zero (0) means unlock and any other value represents locked.
- A segment of code in which a thread may be accessing some shared variable is called a critical region.
- Mutex variable is just like binary semaphore. But both are not same.

• Semaphore :-

- Semaphore is a non-negative integer variable that is used as a Flag used to solve critical section problem.
- Semaphore is an operating system abstract data type. It takes only integer value.
- Dijkstra introduced to operations to operate on semaphore
- Process calls P operation when it wants to enter its critical section & calls V operation when it wants to exit its critical operation.
- Wait Operation (P) on semaphore decrease its value by

```
waits : while S<0
            do loops ;
        S: = S-1 ;
```

- Signal Operation (V) increments its value signal :

$S := S+1;$

- Properties of semaphore.
1. Machine independant.
2. Simple to implement.
3. Correctness is easy to determine.

Q2. Explain types of Semaphore?

⇒ 1. Binary Semaphore :-

- Binary semaphore is also known as mutex locks. It deals with the critical section for multiple processes.
- Binary semaphore value is only between 0 and 1.
- Used to control access to a single resource.
- at value = 1, resource is available
  value = 0, resource is being used.

2. Counting Semaphore :-

- Used with that resource which has Finite number of instances.
- Non-binary semaphore are often referred to as either a counting semaphore or a genered semaphore.
- It can take any non-integer value.
- Used to control access to multiple instances of a resource.
- The value represents the number of available resources.
- Process waiting for semaphore is stored in the queue for binary and counting semaphore.

Q3. Explain deadlock concepts.

⇒ - A lack of process synchronization can result in two extreme conditions are deadlock or starvation.
- Deadlock is the problem of multiprogrammed system.
- Deadlock can be defined as the permanant blocking of a set of processes that either complete for system.
- In a system deadlock, one or more processes are deadlocked.
- Deadlock Example :-

1. Computer system is collection of limited / Finite number of resources. These resources are distributed among number of computing processes.
2. Resources are of two types.
   i] Reusable resources
   ii] Consumable resources.

- Necessary Condition for Deadlock

1. Mutual exclusion :- A resource may be aquired exclusively by only process at a time.
2. Hold & Wait :- Process currently holding resources that were granted earlier.
3. Non-preemption :- once process has obtained a resource the system cannot remove it from control unit.
4. Circular Wait :- A circular chain of hold & wait condition exists in a system.

- Deadlock Prevention :-

To prevent a deadlock, the OS must eliminate one of the four necessary conditions given above.

- Deadlock Avoidance :-

The system must be able to decide whether granting a resource is safer or not anly make the allocation when it is safe.