1. Consider three threads (`T1`, `T2`, and `T3`) that are created from the following code.

```
semaphore s1 = 1, s2 = 1, s3 = 1;

 void T1()                 void T2()                 void T3()
 {                         {                         {
      …                         …                         …
      semWait(s3);              semWait(s2);              semWait(s1);
      …                         …                         …
      semWait(s1);              semWait(s3);              semWait(s2);
      …                         …                         …
      semSignal(s3);            semSignal(s2);            semSignal(s1);
      …                         …                         …
      semSignal(s1);            semSignal(s3);            semSignal(s2);
 }                         }                         }

void main()
{
      parbegin (T1, T2, T3);
}
```

a) Show a case in which deadlock will occur.
b) Draw a resource allocation graph that shows the deadlock state.
c) Will deadlock occur if the order of resource requests of `T1` is changed as follows?

```
void T1()
{
     …
     semWait(s1);
     …
     semWait(s3);
     …
     semSignal(s3);
     …
     semSignal(s1);
}
```

2. Suppose that there is a resource deadlock in a system. Is it possible for a process which is NOT in the circular chain in the corresponding resource allocation graph to be deadlocked?

3. Consider a system with a total of 150 units of memory, allocated to three processes as shown:

Process	Claim	Allocation
1	70	45
2	60	40
3	60	15

Apply the ***banker's algorithm*** to determine whether it would be safe to grant each of the following requests. If yes, indicate a sequence of terminations that could be guaranteed possible.
a) A fourth process arrives, with a maximum memory need of 60 and an initial need of 25 units.
b) A fourth process arrives, with a maximum memory need of 60 and an initial need of 35 units.

Self-test

1. A condition of policy that must be present for a deadlock to be possible is _____.
A. Mutual exclusion
B. Hold and wait
C. No preemption
D. All of the above

2. A closed chain of processes exists, such that each process holds at least one resource needed by the next process in the chain is the condition of _____ .
A. no preemption
B. mutual exclusion
C. circular wait
D. hold and wait

3. The _____ condition can be prevented by requiring that a process request all of its required resources at one time and blocking the process until all requests can be granted simultaneously.
A. mutual exclusion
B. hold and wait
C. circular wait
D. no preemption

4. The _____ condition can be prevented by defining a linear ordering of resource types.
A. hold and wait
B. no preemption
C. mutual exclusion
D. circular wait

5. In the banker's algorithm, a safe state is defined as one in which _____.
A. At least one potential process sequence does not result in a deadlock
B. All potential process sequences do not result in a deadlock:
C. Several potential process sequences do not result in a deadlock:
D. None of the above

6. A strategy for dealing with deadlocks that allows the presence of deadlock is called _____.
A. Deadlock Prevention
B. Deadlock Avoidance
C. Deadlock Detection
D. None of the above