

Q1) Consider the following solution to the synchronization problem for process P_i :

Shared data:

int turn = i; *//can take either i or j*

bool flag[2] = {false,false};

```
do{
  A1    flag[i] = true;
  A2    turn = (i+1) % 2;
  A3    while(flag[(i+1)%2] && (turn == (i+1) %2));
  A4    


Critical section


  A5    flag[i] = false;
  A6    

Remainder section


}while(true);
```

Which of the following are satisfied?

- a) Only progress
-  b) Both mutual exclusion and bounded waiting
- c) Only mutual exclusion
- d) All are satisfied
- e) None is satisfied

Answer: b) Both mutual exclusion and bounded waiting

Q2) Consider the concurrently running processes A and B given below.

Shared data: semaphore $s1=0$, $s2=0$;

Process A:

```
do{
    ...
    wait(s1);
    instruction a;

    instruction b;


    instruction c;
    signal(s2);
}while(true);
```

Process B:

```
do{
    ...
    instruction d;
    signal(s1);

    wait(s2);
    instruction e;
}while(true);
```

Which of the following order of executions is possible?

- a) abcde
- b) daebc
- c) deabc
-  d) dabce
- e) dceba

Answer: d) dabce

Q3) Suppose that there are two processes A and B which share the semaphore variables R and S. These semaphore variables are initialized as $R=0$ and $S=1$. Each section is composed of a set of instructions.


Process A

```
Wait (R)
Wait (S)
Section_A1
Signal (S)
Section_A2
Signal (R)
```

Process B

```
Section_B1
Signal (R)
Section_B2
Wait (S)
Section_B3
Signal (S)
```

Then, which of the following is true?

- A. Execution in Section_A1 cannot start until Section_B1 is complete
- B. Execution in Section_A1 and Section_B2 simultaneously is not possible
- C. Execution in Section_A1 and Section_B3 simultaneously is possible
- D. Execution in Section_A2 and Section_B3 simultaneously is not possible
-  E. both C and D

Answer : E) both c and d