TUTORIAL FOUR

Process Synchronization- Part I

1.

a) Describe the three key requirements that must be satisfied by any solution to the critical section problem.

b) Consider the following user-level solution to the critical section problem, where **flag** and **turn** are shared variables initialized to **false** and **0**, respectively.

Process P0

```
while(1){
   flag=true;
   while(turn==1);
   critical-section
   turn=1;
   remainder-section
}
```

Process P1

```
while(1){
   turn=0;
   while(flag and turn==0);
   critical-section
   flag=false;
   remainder-section
}
```

Determine which of the three requirements in part (a) are not satisfied. Justify your answer.

2. Indicate whether the following statements are true or false. Justify your answers.

a) Race condition only occurs because a single high-level C instruction (e.g., counter++;) is translated into multiple low-level assembly instructions (e.g., register=counter; register=register+1; counter=register).
b) Mutual exclusion can be achieved by disabling interrupts during the critical section.
c) If a solution to a critical section problem satisfies progress, then it also satisfies bounded waiting.

3. Consider a computer that does not have a *TestAndSet* instruction, but has an instruction to ***swap*** the contents of a register and memory word in a single atomic command. Show how it can be used to implement the *entry section* and *exit section* which are before and after the critical section.