TUTORIAL FIVE

Process Synchronisation- Part II

1. Describe how a counting semaphore can be used to control a pool of identical resources.

2. Show that, if the *wait*(*S*) (see the implementation below, where *S* is a semaphore) operation is not executed atomically, then mutual exclusion may be violated.

```
wait(S){
  while(S<=0) ;
    S--;
}
```

3. Describe how *wait*(*S*) and *signal*(*S*) of a semaphore can be implemented using a *TestAndSet* instruction, given the below semaphore structure definition. *Hints: the semaphore value and the process queue L are shared variables among different processes when those processes access the semaphore.*

```
typedef struct {
    int value;
    struct process *L;
} semaphore;
```