Solutions:

The functions must be **atomic** (indivisible).

2. Two threads, **T1()** and **T2()** are accessing the same critical sections (CS). They share three semaphores S1, S2, and S3. Consider the following piece of code for T1 and T2:

T1:		T2:
down (S1); down (S3); down (S2) CS(); up (S2); up (S3); up (S1);		down (S3); down (S2); CS(); up (S2); up (S3);

(i) Does the code achieve the purpose of mutual exclusion? Why?
(ii) Is deadlock possible for T1 and T2? Why? If your answer is yes, modify the code so that the two threads are deadlock free. If your answer is no, explain or prove your claim.

Solutions:

- (i) Yes, it is because a thread can enter the critical section only if it has acquired the two or three semaphores ("keys"). If one thread holds a semaphore, the other cannot have it. So only one thread can enter the critical section.
- (ii) No, deadlock is not possible. This is because both threads request the semaphores in the same order (S3 then S2). Thus there can never be a circular wait and consequently no deadlock is possible.