17.15 Consider the following two transactions:

T_{13}: read(A);
read(B);
if $A = 0$ **then** $B := B + 1$;
write(B).

T_{14}: read(B);
read(A);
if $B = 0$ **then** $A := A + 1$;
write(A).

Let the consistency requirement be $A = 0 \vee B = 0$, with $A = B = 0$ as the initial values.

a. Show that every serial execution involving these two transactions preserves the consistency of the database.

b. Show a concurrent execution of T_{13} and T_{14} that produces a nonserializable schedule.

c. Is there a concurrent execution of T_{13} and T_{14} that produces a serializable schedule?

a. （1）串行执行：先 T13，后 T14

初始值	A=0，B=0
T13	A=0，B=1
T14	A=0，B=1

此时满足一致性需求，A=0∪B=0

（2）串行执行：先 T14，后 T13

初始值	A=0，B=0
T13	A=1，B=0
T14	A=1，B=0

此时满足一致性需求，A=0∪B=0

b.

T13	T14
read(A)	
read(B)	
	read(B)
	read(A)
if A=0 then B:=B+1	
write(B)	
	if B=0 then A:=A+1
	write(A)

由上表可知，最终 A=1，B=1，不满足一致性需求 A=0∪B=0。

但串行化调度都满足，所以其执行产生不可串行化调度。

c. 不存在，因为串行化调度满足 A=0∪B=0。而在并发执行过程中，若 T13 先执行 read(A)，则必有 B=1，而 read(B)是 T14 的第 1 条语句，所以为实现并发，read(B)会出现在 write(B)之前，所以读到的 B 必定为 0，所以 T14 会使 A=1。

若 T14 先执行 read(B)，则同理见上。

所以，无论以什么顺序并发执行，都不能满足一致性条件 A=0∪B=0。

18.2 Consider the following two transactions:

T_{34}: read(A);
read(B);
if $A = 0$ **then** $B := B + 1$;
write(B).

T_{35}: read(B);
read(A);
if $B = 0$ **then** $A := A + 1$;
write(A).

Add lock and unlock instructions to transactions T_{31} and T_{32} so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?

对于 T34：

```
lock-S(A)
read(A)
lock-X(B)
read(B)
if A = 0
then B := B + 1
write(B)
unlock(A)
unlock(B)
```

对于 T35：

```
lock-S(B)
read(B)
lock-X(A)
read(A)
if B = 0
then A := A + 1
write(A)
unlock(B)
unlock(A)
```

所以可列出下表：

T34	T35
lock-S(A)	
	lock-S(B)
read(A)	
	read(B)
lock-X(B)	
	lock-X(A)

由于 X 锁和 S 锁不兼容，所以 T34 等待 T35 释放 S(B)，T35 则等待 T34 释放 S(A)，导致死锁。