16.2

|  |  |
| --- | --- |
| **Transaction T** | **Transaction U** |
| read(j) | read(k) |
| y = read (i)  write(j, 44)  write(i, 33) |  |
|  | write(i, 55)  y = read(j)  write(k, 66) |

|  |  |
| --- | --- |
| **Transaction T** | **Transaction U** |
|  | read(k)  write(i, 55)  y = read(j)  write(k, 66) |
| read(j)  y = read (i)  write(j, 44)  write(i, 33) |  |

|  |  |
| --- | --- |
| **Transaction T** | **Transaction U** |
| read(j)  y = read (i)  write(j, 44)  write(i, 33) |  |
|  | read(k)  write(i, 55)  y = read(j)  write(k, 66) |

16.3

Start:

T:x = read(j); y = read (i); write(j, 44); write(i, 33); U:x = read(k); write(i, 55); y = read (j); write(k, 66).

i

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction T** | | **Transaction U** | |
| Read(j) |  | Read(k) |  |
| Read(i)  Write(j, 44)  Write(i,33) | Lock j  Lock i |  |  |
|  |  | write(i, 55)  y = read(j)  write (k, 66) | Wait for for T to release I, j |

ii

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction T** | | **Transaction U** | |
| Read(j) |  | Read(k) |  |
| Read(i)  Write(j, 44)  Write(i,33) | Lock j |  |  |
|  |  | write(i, 55)  y = read(j)  write (k, 66) | Wait for release on j |

iii

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction T** | | **Transaction U** | |
| Read(j) |  | Read(k) |  |
| Read(i)  Write(j, 44)  Write(i,33) | Lock j  Lock i |  | Wait for for T to release i |
|  |  | write(i, 55)  y = read(j)  write (k, 66) | Lock i |
|  |  | commit |  |
|  | Abort |  |  |

16.8

Since one of the requirements of serial equivalence is to have to have the transaction complete all its operations on conflicting objects before releasing the lock, otherwise it would not be as if it were run in serial.

T: x = read(i); write(j, 44);

U:write(i, 55); write(j, 66);

|  |  |
| --- | --- |
| **Transaction T** | **Transaction U** |
|  | Write(i,55) |
| X = Read(i) |  |
| write(j, 44) |  |
|  | Write(j, 66) |
|  |  |