

19.21) No, since the system crash after normal read\_item transaction, which will cause a difference neither in the main memory nor in the Database.

19.22) if the system crashes before [write\_item,T2,D,25,26], then also this will not cause a different behavior in the recovery process, as the transaction T2 is still not committed, so the recovery system was going to undo all the uncommitted items.

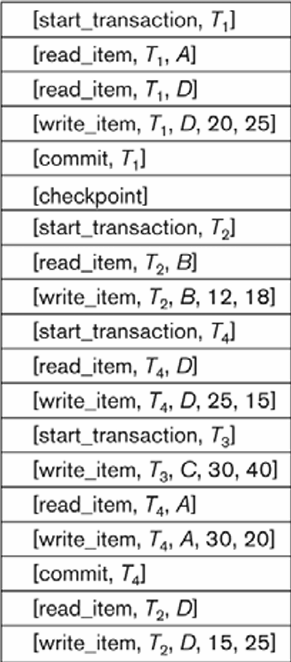
A screenshot of a computer

Description automatically generated

Note:

* Immediate update protocols are -> Undo and Undo/Redo.
* Deferred update protocols are -> Redo and Undo/Redo
* So here in this question we will use the Undo first then try to apply Undo / Redo.

1. **Undo solution**



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| Stop here. |
| Undo |
| Undo |
| T2 not committed -> B = 12 |
| Ignore – committed |
| Ignore – committed |
| Ignore – committed |
| Undo |
| T3 not committed -> C = 30 |
| Ignore – committed |
| Ignore – committed |
| T4 is done, so we will just ignore |
| T2 not committed -> undo read. |
| Crash --- |

1. A screenshot of a computer

   Description automatically generated**Undo/Redo**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| Stop here. |
| Undo |
| Undo |
| T2 not committed -> B = 12 (Undo) |
| Redo (6) in that order 6,5,4,3,2,1. |
| Redo (5) |
| Redo (4) -> D = 15 |
| Undo |
| T3 not committed -> C = 30 (Undo) |
| Redo (3) |
| Redo (2) -> A = 20 |
| T4 is done, so we need to Redo (1) |
| T2 not committed -> undo read. |
| Crash --- |

A screenshot of a computer

Description automatically generated**19.24) Using deferred update (Redo)**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| Stop here. |
| Ignore |
| Ignore |
| Ignore |
| Redo (6) in that order 6,5,4,3,2,1. |
| Redo (5) |
| Redo (4) -> D = 15 |
| ignore |
| ignore |
| Redo (3) |
| Redo (2) -> A = 20 |
| T4 is done, so we need to Redo (1) |
| Ignore |
| Crash --- |