Problem 1. [30 pts]

1. In the following schedules, $R_i(A)$ stands for a Read$(A)$ operation by transaction $i$ and $W_i(A)$ stands for a Write$(A)$ operation by transaction $i$. For each of the following schedules show if it is conflict-serializable and give a conflict-equivalent serial schedule. Show all the conflict operations. Is any of the schedules view-serializable? If yes, give a view-equivalent serial schedule.

   (a) $R_1(A)R_2(B)W_3(A)R_2(A)R_1(B)$
   (b) $R_1(A)R_2(B)W_1(A)R_3(C)W_2(B)W_3(C)R_4(D)R_4(A)W_4(D)$
   (c) $R_3(E)R_1(D)W_2(C)W_3(A)R_1(E)W_4(B)R_1(B)W_3(E)R_4(A)W_4(C)$

2. For the following 2 schedules, insert the appropriate locks (shared and exclusive) into the schedule following the Strict 2PL protocol. Also explain what happens as the scheduler executes each schedule. Write the actual executed schedule.

   (a) $R_1(A)R_2(B)R_3(C)W_1(B)W_2(C)W_3(D)$
   (b) $R_1(A)R_2(B)R_3(C)R_1(B)R_2(C)R_3(A)W_1(A)W_2(B)W_3(C)$
Problem 2. [20 pts]

Consider the following hierarchy of lock-able objects:

- Database
  - Employee-Table(ssn, name, salary)
    * E1(132, Smith, 20K)
    * E2(456, Kelley, 40K)
    * E3(678, Johnson, 400K)
    * E4(792, Preeston, 40K)
    * E5(865, Johnson, 60K) ...
  - DPT-table(dnumber, dname, budget)
    * D1(1, Marketing, 1M)
    * D2(2, Engineering, 2M)
    * D3(3, R&D , 4M)
    * D4(4, HR , 1M) ...

For each of the following transactions list the objects and the locks that must be requested in the proper order, assuming that a multiple-granularity protocol is used:

1. T1: wants to read the salary for 'Smith'
2. T2: wants to change the name of 'Johnson' to 'Jonson'
3. T3: wants to give a 5% raise to every employee
4. T4: wants a list of all department names
Problem 3. [50 pts]

Consider the log below. The records are of the form: (t-id, object-id, old-value, new-value).

Assumptions: a. The PrevLSN has been omitted (it’s easy to figure it out yourself); b. for simplicity we assume that A, B, C and D each represents a page;

1. (T1, start)
2. (T1, A, 45, 10)
3. (T2, start)
4. (T2, B, 5, 10)
5. (T2, C, 35, 10)
6. (T1, D, 15, 5)
7. (T1, commit)
8. (T3, start)
9. (T3, A, 10, 15)
10. (T2, D, 5, 20)
11. (begin checkpoint, end checkpoint)
12. (T2, commit)
13. (T4, start)
14. (T4, D, 20, 30)
15. (T3, C, 10, 15)
16. (T3, commit)
17. (T4, commit)

A. (20 points) What are the values of pages A, B, C and D in the buffer pool after recovery? Please specify which transactions have been ReDo and which transactions have been UnDo, and you are not required to show the details of the intermediate step.

1. if the system crashes just before line 6 is written to the stable storage?
2. if the system crashes just before line 10 is written to the stable storage?
3. if the system crashes just before line 13 is written to the stable storage?
4. if the system crashes just before line 17 is written to the stable storage?

B. (30 points) Assume only the 3rd crash (as listed in Part A) has really happened and a recovery has then been performed, and the dirty pages caused by T1 have been flushed to disk before line 8, show the details of the Analysis phase (the content of two tables at both the beginning and the end of this phase), ReDo phase and UnDo phase (show the contents of ToUnDoList at each step and CLRs to be written to the LOG). You may assume that both the transaction table and dirty page table are empty at the beginning of line 1 of the log. Finally, show the content of log when the recovery has completed.