A. Functional dependencies/normalization

1. Given the schema R(A,B,C,D,E), and functional dependencies A->D, B->C, CD->E, A->BC, E->B.
a) Is the schema in BCNF? If not, list an FD that violates BCNF.
b) Is the schema in 3NF? If not, list an FD that violates 3NF.

2. Problem 7.11 . Given functional dependencies A->BC, CD->E, B->D, E->A, and the relation (A,B,C,D,E), show that the decomposition (A,B,C), (A,D,E) is not dependency preserving.

3. Decompose the schema from problem 1 into BCNF and 3NF.

4. Problem 7.23. Show that the following decomposition of the schema from problem 2 is not a lossless decomposition: R1=(A,B,C), R2=(C,D,E). Hint: give an example relation (set of tuples in schema R=(A,B,C,D,E)) such that joining the the projection of R on R1 and R2 fails to generate the original set of tuples in R.

B. Storage/Files/Indexing

1. Assume you have two disk systems, one organized as a RAID level 1 (mirroring), and a second one organized as a RAID level 5 (block interleaved distributed parity).
   a. Which of these is more efficient if multiple users try to access the data?
   b. How many disks would you have to access when writing a block of data to each of these systems?

2. Assume you have a memory buffer with 3 slots, and that your database requests the following blocks from disk, in the given order:
A B A C A B AC B D C

   a. Which blocks will be in memory at the end of this sequence if using the MRU strategy?
   b. Which blocks will be in memory at the end of this sequence if using the LRU strategy?

3. Assume you are given the following B+ tree where each internal node must contain between 2 and 3 records.

```
      7|15|
    /  \    \
   /     \    \
1|2|6   7|9|  15|16|
```
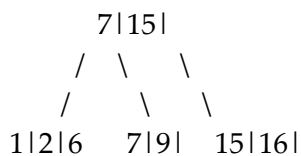
a. What will the tree look like after inserting value 3?
b. What will the tree look like after deleting value 15?
c. How many blocks will you need to bring into memory for each of the operations in points a and b?

C. Query processing

1. Explain the difference between a relational algebra expression and a query plan.

2. Assume you have a B+ tree index with a fan-out of 10 built for attribute A of relation r, and that this relation contains 100,000 records. Also assume that it takes $t_s = 10$ ms to seek a block on the disk, and $t_t = 1$ms to transfer a block into memory. Further assume a disk block can hold 20 records.

a. How long would the operation $\sigma_{A=100}(r)$ take?
b. How long would the operation $\sigma_{A>100 \wedge A<200}(r)$ take, assuming that the values for attribute A range between 0 and 1000?
c. How long does the operation $\sigma_{C=10}(\sigma_{A>100 \wedge A<200}(r))$ take, where C is another attribute of relation r, and assuming that the intermediate results of the nested selection are written back to disk?
d. In answering c, what assumptions did you make about the outer selection?

3. Using the parameters from problem 2, describe the time it would take to perform the three queries listed above in a:
a. ordered file (using binary search)
b. unordered file.