Data Management – exam of 15/06/2009

LAST NAME: _____

NAME: _____

STUDENT CODE: _____

I allow the publication of my grade on the Web site `http://www.dis.uniroma1.it/~lenzerini`, according to the Italian regulation (decreto legislativo 196/2003), which I hereby declare to know.
Faithfully,

_____

Problem 1 Consider the following schedule

$$S = r_2(T)\, r_1(P)\, w_1(S)\, w_1(P)\, c_1\, r_2(S)\, w_2(T)\, r_2(P)\, w_4(Q)\, w_3(T)\, r_2(Q)\, w_4(T)\, r_3(P)\, w_4(P) c_4\, c_3\, c_2.$$

1. Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then explain the answer.
2. Tell whether S is strict or not, and explain the answer.
3. Tell whether S is ACR (Avoid Cascading Rollback) or not, and explain the answer.
4. Tell whether S is conflict-serializable. If the answer is yes, then show a serial schedule that is conflict-equivalent to S. If the answer is no, then explain the answer.

Problem 2 A schedule with at most one "write" action is called a *1-write-only* schedule. Prove or disprove the following statement: every 1-write-only schedule is view serializable.

Problem 3 Let `Product(ProductCod,Description,ProductionYear,Cost)` be a relation whose key is `ProductCod`, and whose tuples require 1024 pages in secondary storage. The relation is used only by the following two operations:

- *search*: given the ProductCode of a product, return the cost of such product,
- *insertion*: insert a new product.

There are two options for representing the relation `Product`, namely by a heap file, or by a sorted file. For each of the following two scenarios, tell which of the two options you would choose, and explain your decision:

- **Scenario A**: 1 search every day, and 4 insertions every day.
- **Scenario B**: 3 searches every week, and 2 insertions every week.

Problem 4 Consider the relations R(A,B) and S(B,C), where B is a key for S, R is constituted by 3000 pages, each page of R contains 100 tuples, and S is constituted by 1000 pages.

What is the cost (in terms of number of pages that are either read or written) of the natural join between R and S if we use the naive nested loop algorithm with R as the outer relation? Explain in detail your answer.

Suppose that we have a hash index on S with search key B. What is the cost (in terms of number of pages that are either read or written) of the natural join between R and S if we use the index nested loop algorithm with R as the outer relation? Explain in detail your answer.