1. (12 pts) Prolog

Given the following clauses, list all answers returned by the following queries.

```prolog
avenger(thor).
avenger(captainAmerica).
sibling(thor, loki).
asgardian(thor).
asgardian(X) :- sibling(Y, X), asgardian(Y).
train1(X, Y) :- avenger(X), !, avenger(Y).
train2(X, Y) :- avenger(X), X \= Y, avenger(Y).
```

a. (1 pt) ?- avenger(captainAmerica).

b. (1 pt) ?- asgardian(A).

c. (1 pt) ?- train1(A, B).

d. (1 pt) ?- train2(A, captainAmerica).

e. (2 pt) ?- train2(captainAmerica, A).

f. (2 pt) ?- foo([1, 2, 3], A).

g. (2 pts) ?- foo([2, 2, 4], A).

h. (2 pts) ?- foo([5, 2, 2, 3, 4, 4], A).
```
2. (8 pts) Multithreading

| class Buffer {                     | void produce(o) {               | Object consume( ) {                      |
| Buffer ( ) {                      |     synchronized (this) {       |     synchronized (this) {               |
|     Object buf = null;           |         1. if (!empty) wait( ); |         5. if (empty) wait( );           |
|     boolean empty = true;        |         2. empty = false;       |         6. empty = true;                |
| }                                 |         3. notifyAll( );         |         7. notifyAll( );                |
|                                   |         4. buf = o;              |         8. return buf;                  |
|                                   |     }                          |     }                                   |
|                                   | }                              | }                                       |

Consider the preceding multithreaded Java 1.4 code. Assume there are multiple producer and consumer threads being executed in the program, but only a single Buffer object. Questions about the “last statement executed” by two threads x & y refer to the most recently executed statement by those threads at some arbitrary time during the program execution. It does not mean the last statement executed by a thread before the thread exits. If a situation is possible, you need to give an example of how it is possible (e.g., thread x gets to statement a, then thread y gets to statement b). If a situation is not possible, you need to explain why.

a. (2 pts) Is it possible given two threads x and y for the last statement executed by thread x to be statement 2 and thread y to be statement 7 in the code above? Explain your answer.

b. (3 pts) Is it possible given two threads x and y for the last statement executed by thread x to be statement 3 and thread y to be statement 5 in the code above? Explain your answer.

c. (3 pts) Is it possible in the code above for two threads calling consume( ) to get the same value from the Buffer object? Explain your answer.