1. (12 pts) Prolog

You are writing a program to analyze 2-card starting hands in Texas Hold’em, a poker game. Given the following clauses, list all answers returned by the following queries.

\[
\begin{align*}
\text{pair}(\text{pocketRockets}). \\
\text{pair}(\text{kingKong}). \\
\text{beats}(\text{pocketRockets}, \text{kingKong}). \\
\text{beats}(\text{bigSlick}, \text{bigChick}). \\
\text{beats}(X,Y) :- \text{pair}(X), \neg \text{pair}(Y). \\
\text{crushes}(X,Y) :- \text{pair}(X), \neg \text{pair}(Y).
\end{align*}
\]

\[
\begin{align*}
\text{foo}([X|Y], X). \\
\text{foo}([X,Y|T], R) :- \text{foo}(T, R).
\end{align*}
\]

a. (1 pts) ?- beats(pocketRockets, bigChick).
   true.

b. (1 pts) ?- beats(X, bigChick).
   X=bigSlick;
   X=pocketRockets;
   X=kingKong.

c. (1 pts) ?- beats(pocketRockets, X).
   Y=kingKong.

d. (2 pts) ?- beats(X, Y).
   X=pocketRockets,
   Y=kingKong;
   X=bigSlick,
   Y=bigChick.

e. (2 pts) ?- crushes(X, bigChick).
   X=pocketRockets.

f. (1 pts) ?- foo([1], A).
   A=1.

g. (2 pts) ?- foo([1,2], A).
   A=1.

h. (2 pts) ?- foo([1,2,3,4,5], A)
   A=1;
   A=3;
   A=5.
2. (8 pts) Multithreading

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 a thread refer to the most recently executed statement by that thread 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.

<table>
<thead>
<tr>
<th>class Buffer { Buffer ( ) { Object buf = null; } }</th>
<th>void produce(o) { synchronized (this) { 1. while (buf != null) wait( ); 2. buf = o; 3. notifyAll( ); } }</th>
<th>Object consume( ) { synchronized (this) { 4. if (buf == null) wait( ); 5. notifyAll( ); 6. Object tmp = buf; 7. buf = null; 8. return tmp; } }</th>
</tr>
</thead>
</table>

a. (2 pts) Is it possible given 3 threads x, y, and z for the last statement executed by thread x to be statement 2, thread y to be statement 1, and thread z to be statement 5 in the code above? Explain your answer.

No, since thread x and z are both within the synchronized region, and they cannot both acquire the lock for the buffer at the same time (since neither thread is at a call to wait).

b. (3 pts) Is it possible in the code above for two threads x & y calling consume( ) to have consume( ) return null for thread y? Assume produce(o) is never called with o=null. Explain your answer.

Yes, since wait is not called within a while loop thread y may continue past wait even though buf is null.

c. (3 pts) Is it possible in the code above for two threads x & y calling produce(o) to have y overwrite the value x assigns to buf? Explain your answer.

No, since wait is called within a while loop thread y may not continue past the wait until buf is null.