CS322 Fall 1997
Assignment 3

Due: 1:30pm, Friday 26 September 1997.

Question 1

[10 marks] Given the clauses:

```
rd(co(H,co(H,T)),T).
rd(co(A,T),co(A,R)) <-
   rd(T,R).
```

Consider the query:

```
ask rd(co(a,co(co(a,X),co(B,co(c,co(d,em))))),W).
```

1. [2 marks] How many answers are there to this query? What are they (give just the values for the variables *B* and *W* in the answer)?
2. [5 marks] Give an SLD derivation showing all substitutions for one of the answers in the above query.
3. [3 marks] Give a failing derivation for this query.

Question 2

In this question you are to write a CILog knowledge base for the design of custom video presentations.

You should assume that the video is annotated using the relation

```
segment(SegId, Duration, Covers)
```

where *SegId* is an identifier for the segment. (In a real application this will be enough information to extract the segment from the video disk). *Duration* is the time of the segment (in seconds). *Covers* is a list of topics that is covered by the video segment. An example of a video annotation is the database:

```
segment(seg0,10,[welcome]).
segment(seg1,30,[skiing,views]).
segment(seg2,50,[welcome,computational_intelligence,robots]).
segment(seg3,40,[graphics,dragons]).
segment(seg4,50,[skiing,robots]).
```

A presentation is a sequence of segments. You will represent a presentation by a list of segment identifiers.

1. [7 marks] Axiomatize a predicate

   ```
   presentation(MustCover, Maxtime, Segments).
   ```

   That is true if *Segments* is a presentation whose total running time is less than or equal to *Maxtime* seconds, such that all of the topics in the list *MustCover* are covered by a segment in the presentation. The aim of this predicate is to design presentations that cover a certain number of topics within a time limit.

   For example, given the query:

   ```
   cilog: ask presentation([welcome,skiing,robots], 90, Segs).
   ```

   should at least return the two answers (perhaps with the segments in the other order):

   ```
   Answer: presentation([welcome, skiing, robots], 90, [seg0, seg4]).
   Answer: presentation([welcome, skiing, robots], 90, [seg2, seg1]).
   ```

   Two procedures you may find useful are:

   ```
   % member(E,L) is true if E is in list L
   member(A,[A|R]).
   member(A,[H|L]) <-
      member(A,L).

   % notin(E,L) is true if E is not in list L
   notin(E,[]).
   notin(A,[B|L]) <-
      A \= B &
      notin(A,L).
   ```

   You must hand in a listing of your axiomatization. You need to specify the intended interpretation of all symbols used. You need to demonstrate that you have tested your axiomatization (including finding all answers to your query). You should explain briefly why each answer is an answer.

2. [3 marks] Note that what is required for part 1 is reasonably straightforward. However, this example domain will be used for future assignments, so it is worthwhile thinking about what you may want in such a presentation design program.

   Assuming you have a good user interface and a way to actually view the presentations, list *three* things that the above program doesn't do that you may want in such a presentation system.

   [There is no right answer for this part, you need to be creative to get full marks].