CS 311 - Programming Language Concepts
Programming Assignment #4

Objective:

- To program using the logic programming paradigm.

Assignment:

Download and install SWI-Prolog from **http://www.swi-prolog.org/**

OR, you can use the online Prolog compiler at **https://swish.swi-prolog.org**

Research Prolog features and programming style, then answer the 5 questions below.

Brief Introduction to Prolog

Note: You will find more programming related information in the help file that comes up directly after you install SWI-Prolog.

Relations in Prolog

- Prolog programs specify *relationships* among objects and properties of objects.
- When we say, “John owns the book,” we are declaring the ownership relationship between two objects: John and the book.
- When we ask, “Does John own the book?” we are trying to find out about a relationship.
- Relationships can also rules such as:

  Two people are sisters **if** they are both female **and** they have the same parents.

- A rule allows us to find out about a relationship even if the relationship isn’t explicitly stated as a fact.

A little more on being sisters

- As usual in programming, you need to be a bit careful how you phrase things:
- The following would be better:

  A and B are sisters **if**

  A and B are both female **and**
  they have the same father **and**
  they have the same mother **and**
  A is not the same as B

Programming in Prolog

- Declare facts describing explicit relationships between objects and properties objects might have (e.g. Mary likes pizza, grass has color green)
- Define rules defining implicit relationships between objects (e.g. the sister rule above) and/or rules defining implicit object properties (e.g. X is a parent if there is a Y such that Y is a child of X).

One then uses the system by:

- Asking questions above relationships between objects, and/or about object properties (e.g. does Mary like pizza? is Joe a parent?)

Facts

- Properties of objects *or* relationships between objects;
- "Dr Hecker lectures in course 3120," is written in Prolog as:
- lectures(hecker, 3120).
- *Notice that:*
  - names of properties/relationships begin with lower case letters.
  - the relationship name appears as the first term
  - objects appear as comma-separated arguments within parentheses.
  - A period "." must end a fact.
  - objects also begin with lower case letters. They also can begin with digits (like 3120), and can be strings of characters enclosed in quotes (as in reads(fred, "War and Peace")).
- lectures(hecker, 3120). is also called a *predicate*

Facts about a hypothetical computer science department:

```
% lectures(X, Y): person X lectures in course Y
    lectures(hecker, 3120).
    lectures(codd, 3311).
    lectures(backus, 3021).
    lectures(ritchie, 3201).
    lectures(minsky, 3414).
    lectures(codd, 3314).

% studies(X, Y): person X studies in course Y
    studies(fred, 3020).
    studies(jack, 3311).
    studies(jill, 3314).
    studies(jill, 3414).
    studies(henry, 3414).
    studies(henry, 3314).

%year(X, Y): person X is in year Y
    year(fred, 1).
    year(jack, 2).
    year(jill, 2).
    year(henry, 4).
```

Together, these facts form Prolog's *database*.

Queries

- Once we have a database of facts (and, soon, rules) we can ask questions about the stored information.
- Suppose we want to know if Hecker lectures in course 3120. Load the above database into prolog and then ask:

?- *lectures(hecker, 3120).* true. ?- *<control-D>* %	"?-" is Prolog's prompt output from Prolog hold down control & press D to leave Prolog

- *Notice that:*
    - In SWI Prolog, queries are terminated by a full stop.
    - To answer this query, Prolog consults its database to see if this is a known fact.
    - In example dialogues with Prolog, the text in *green italics* is what the user types.

Your Assignment:

1. Given the relations

father(X,Y)	X is the father of Y
mother(X, Y)	X is the mother of Y
female(X)	X is female
male(X)	X is male

Define prolog relations for the following:

a. sibling
b. sister
c. grandson
d. descendant

Provide some facts for the father, mother, male, and female predicates and then test the entire thing using Prolog.

2. Write a Prolog relation *remove(E,L,R)* that is true if *R* is the list which results from removing one instance of *E* from list *L*. The relation is false if *E* isn't a member of *L*.

What are all of the answers to the following queries?

ask remove(a,[b,a,d,a],R).

ask remove(E,[b,a,d,a],R).

ask remove(E,L,[b,a,d]).

ask remove(p(X),[a,p(a),p(p(a)),p(p(p(a)))],R).

3. Write a Prolog relation *subsequence(L1,L2)* that is true if list *L1* contains a subset of the elements of *L2* in the same order.

How many different proofs are there for each of the following queries?

ask subsequence([a,d],[b,a,d,a]).

ask subsequence([b,a],[b,a,d,a]).

ask subsequence([X,Y],[b,a,d,a]).

ask subsequence(S,[b,a,d,a]).

Explain why there are that many.

4. Write a Prolog relation that returns a list containing the union of the elements of two given lists.

5. Write another relation (anything you want) that does something not performed above. Explain what your relation does.