**Objective: Exploring a project in prolog.**

**Project Title:** Detective Murder solving Problem

**Description:**

You are a detective trying to solve a murder case

There are three suspects - Art, Burt, and Carl

They are also the only three witnesses

Here's their statements:

**Art:** Burt was the victim's friend, but the victim and carl were deadly enemies.

**Burt:** I was out of town when it happened, and on top of that I didn't even know the guy.

**Carl:** I'm innocent. I don't know who did it. I saw Art and Burt driving around town then.

Determine who is lying.

**M** is the **murderer**

**a, b, and c** are **Art**, **Burt**, and **Carl**

**W** is the current list of witnesses

**Code:**

testimony(a, friend(b)).

testimony(a, enemy(c)).

testimony(b, out\_of\_town(b)).

testimony(b, stranger(b)).

% Ignore innocent and I don't know who did it as both

% consistent with any theory

testimony(c, in\_town(c)). % Carl was in town if he saw Art & Burt

testimony(c, in\_town(a)).

testimony(c, in\_town(b)).

%

% inconsistent meaning

inconsistent(friend(X), enemy(X)).

inconsistent(friend(X), stranger(X)).

inconsistent(enemy(X), stranger(X)).

inconsistent(out\_of\_town(X), in\_town(X)).

murderer(M) :-

member(M, [a, b, c]), % pick somebody

select(M, [a, b, c], Witnesses), % ignore their testimony by

% removing them from witness list

consistent(Witnesses). % if their story is consistent, M is a suspect

%

% A set of witnesses are consistent if among them there is no

% inconsistent testimony

%

consistent(W) :-

\+ inconsistent\_testimony(W).

% A set of witnesses have inconsistent testimony

% if two pieces of testimony belonging to two different

% witnesses are inconsistent

%

inconsistent\_testimony(W) :-

member(X, W), % X and Y are witnesses

member(Y, W),

X \= Y, % and are different

testimony(X, XT), % a piece of X's testimony

testimony(Y, YT), % a piece of Y's testimony

inconsistent(XT, YT). % that are inconsistent

**Output:**

