

You are to submit a well commented listing of your Prolog program, along with the queries you posed, and the responses that you received to them.

Solutions are to be submitted in pairs.

Question 1. A crime wave has hit the city. Your mission is to enlist the help of the Prolog system to solve the crimes. Given below is a series of facts that you are to translate into Prolog, using predicates and constants as specified. The facts and rules you formulate, taken together, will form a Prolog program that you will then query to narrow down the list of crime suspects.

1. Using a predicate `possible_suspect`, and constants for the names of people, formulate the following facts:
 - Fred, Mary, Jane and George are all possible suspects.
2. Using a predicate `crime` and appropriate constants, record the following facts from the police log:
 - (a) Robbery1 was committed against John on Tuesday in the park.
 - (b) Assault1 was committed against Mary on Wednesday in the park.
 - (c) Robbery2 was committed against Jim on Wednesday in the pub.
 - (d) Assault2 was committed against Robin on Thursday in the park.
3. Using a predicate `was_at` record the following facts (twelve facts all together).
 - (a) Fred was at the park on Tuesday, and at the pub on Wednesday and Thursday.
 - (b) George was at the pub on Tuesday and Wednesday, and at home on Thursday.
 - (c) Jane was at home on Tuesday, and at the park on Wednesday and Thursday.
 - (d) Mary was at the pub on Tuesday, at the park on Wednesday, and at home on Thursday.
4. Using predicates `jealous_of` and `owes_money_to` enter the following facts:
 - (a) Fred is jealous of John.
 - (b) Jane is jealous of Mary.
 - (c) George owes money to Jim.
 - (d) Mary owes money to Robin.
5. Finally, you are to formulate the following rules in Prolog (using appropriate predicates):
 - (a) A suspect had a motive against a victim if either the suspect is jealous of the victim, or the suspect owes money to the victim.
 - (b) A person is a prime suspect for a crime if the person is a possible suspect and the person was at the time and place of the crime and the person had a motive against the victim of the crime.
6. Now that your program is complete, you are to formulate queries to the program corresponding to the following questions.
 - (a) Is Fred a prime suspect in Robbery1?
 - (b) Is George a prime suspect in Assault2?
 - (c) Who is (or are) the prime suspect(s) in Robbery1?
 - (d) Who is (or are) the prime suspect(s) in Robbery2?
 - (e) Who is (or are) the prime suspect(s) in Assault1?
 - (f) Who is (or are) the prime suspect(s) in Assault2?
 - (g) In what crimes is Fred a prime suspect?
 - (h) In what crimes is George a prime suspect?

Question 2.

1. Define a predicate `last/2` computing the last member of a nonempty list.
2. Define a predicate `reverse/2` such that `reverse(K,L)` holds if and only if list L is the reverse of list K.
For example, `?- reverse([a,b,c,a],L)` should produce the answer `L = [a,c,b,a]`.
3. Define a predicate `select/3` such that `select(X,K,L)` means that X is a member of list K and L is the result of removing one occurrence of X from K.
For example, `?- select(a,[a,b,c,a],L)` should produce the answers

```
L = [b, c, a];
L = [a, b, c];
No
```

Hint: For the previous 3 questions you may consider to use the predicate `append/3`.

4. Define a predicate `different/3` such that for a list L without repetitions `different(X,Y,L)` means that X and Y are different members of L.

Hint: Use the predicates `select/3` and `member/2`.

5. Define a predicate `samelenhth/2` such that `samelenhth(K,L)` means that K and L are lists of same length.
For example, the query `?- samelenhth([a,b,a],[d,e,f])` should produce the answer `Yes`, where `?- samelenhth([a,b,a],[d,e])` should produce the answer `No`.
What are the answers to `?- samelenhth([a,b,c],L)` and `?- samelenhth(L,[a,b,c])`? Does your program loop if you press “;”? (Remark: there is a solution which does not loop.)
6. Define a predicate `similar/2` such that `similar(K,L)` means that K and L are lists of same length containing the same elements (including their multiplicity).
For example, `?- similar([a,b,a],L)` (and also `?- similar(L,[a,b,a])`) should produce the answers (possibly in a different order)

```
L = [a, b, a] ;
L = [a, a, b] ;
L = [b, a, a] ;
L = [a, a, b] ;
L = [b, a, a] ;
L = [a, b, a] ;
No
```

The answer to `?- similar([a,b,b],[a,b,a])` should be `No`.

Test your predicates by appropriate queries.