

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **B.Tech - CSE** | Semester | : | **FALL 2020** |
| Course | : | **Database Management Systems (Embedded Lab)** | Code | : | **CSE2004** |
| Faculty | : | **Dr.Bhuvaneswari A** | Slot | : | **L31 + L32** |
| Ex. No. | : | **9** | Date | : | **6-Oct-2020** |

**KANDULA NAVEEN KUMAR**

**17MIS1056**

**PROLOG**

|  |
| --- |
| **PROLOG ONLINE TOOL :** <https://swish.swi-prolog.org/>  Sample Tutorial : <https://www.youtube.com/watch?v=g34jS24JWqw> |

Q1. Observe the following simple fact clauses

likes(mary,food).

likes(mary,wine).

likes(john,wine).

likes(john,mary).

How does Prolog respond to the following queries?

| ?- likes(mary,food).

| ?- likes(john,wine).

| ?- likes(john,food).

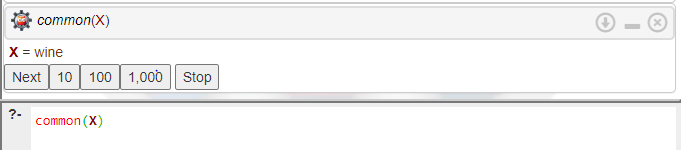
QUESTION:

How do you add the following facts?

1. John likes anything that Mary likes

ANS:

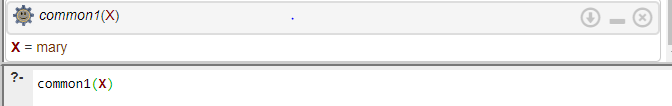




2. John likes anyone who likes wine

ANS:

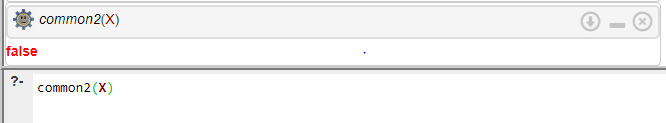




3. John likes anyone who likes themselves

ANS:





Q2. Observe the following simple fact clauses

male(albert). %a fact stating albert is a male

male(edward).

female(alice). %a fact stating alice is a female

female(victoria).

parent(albert,edward). %a fact: albert is parent of edward

parent(victoria,edward).

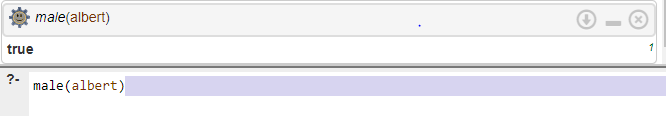
father(X,Y) :- parent(X,Y), male(X). %a rule: X is father of Y if X if a male parent of Y %body of above rule, can be on same line.

mother(X,Y) :- parent(X,Y), female(X)%a similar rule for X being mother of Y

How does Prolog respond to the following queries?

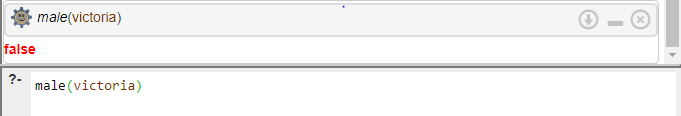
?-male(albert).

ANS:



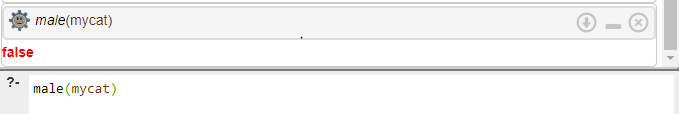
?-male(victoria).

ANS:



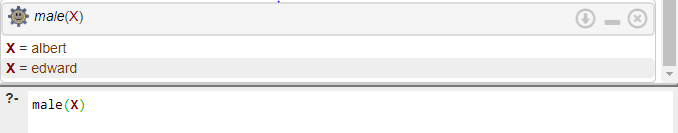
?-male(mycat).

ANS:



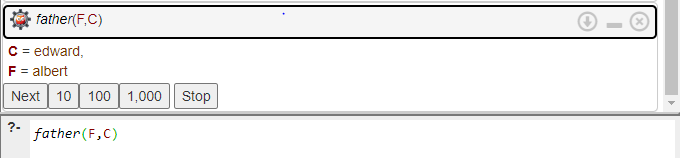
?-male(X).

ANS:



?-father(F,C).

ANS:



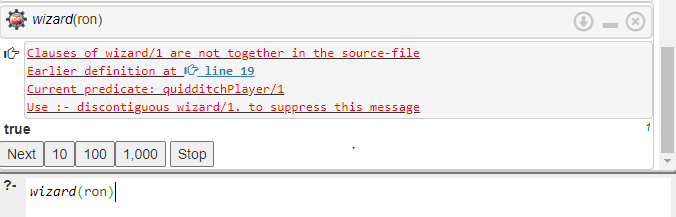
Q3. Suppose we are working with the following knowledge base:  
wizard(ron).  
hasWand(harry).  
quidditchPlayer(harry).  
wizard(X) :- hasBroom(X),hasWand(X).  
hasBroom(X) :- quidditchPlayer(X).

QUESTION:

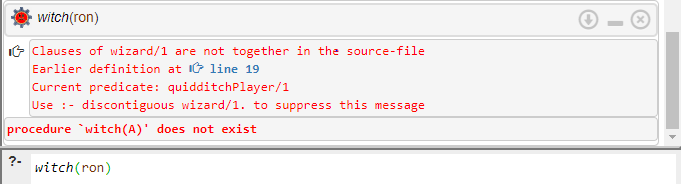
How does Prolog respond to the following queries?

wizard(ron).

ANS:

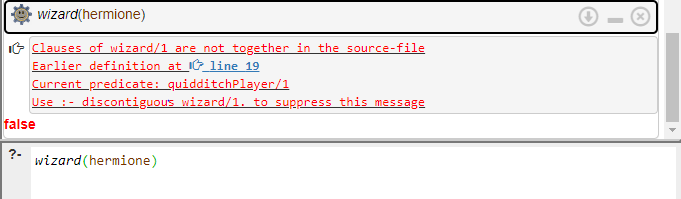
  
witch(ron).

ANS:

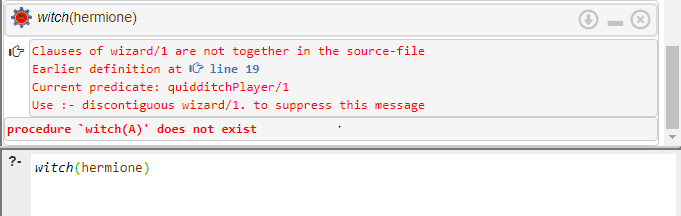


wizard(hermione).

ANS:

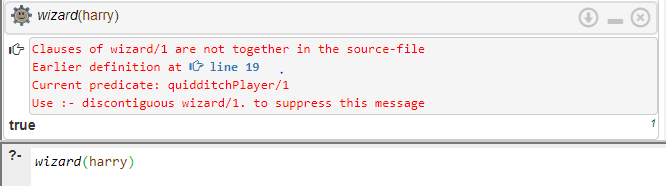
  
witch(hermione).

ANS:

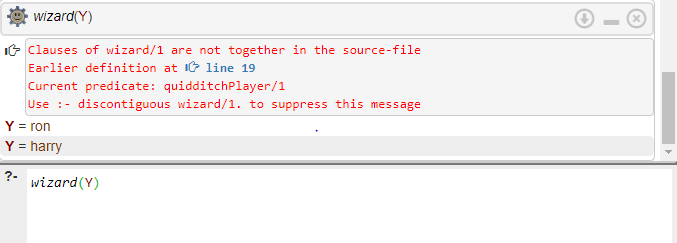


wizard(harry).

ANS:

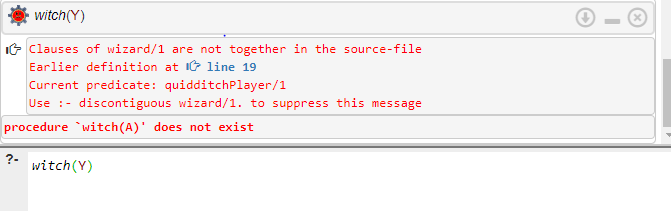


wizard(Y).  
ANS:



witch(Y).

ANS:



Q4. Observe the following simple clauses in family genealogy

James I

|

|

+----------------+-----------------+

| |

Charles I Elizabeth

| |

| |

+----------+------------+ |

| | | |

Catherine Charles II James II Sophia

|

|

|

George I

Here are the facts

-------------------------------

male(james1).

male(charles1).

male(charles2).

male(james2).

male(george1).

female(catherine).

female(elizabeth).

female(sophia).

parent(charles1, james1).

parent(elizabeth, james1).

parent(charles2, charles1).

parent(catherine, charles1).

parent(james2, charles1).

parent(sophia, elizabeth).

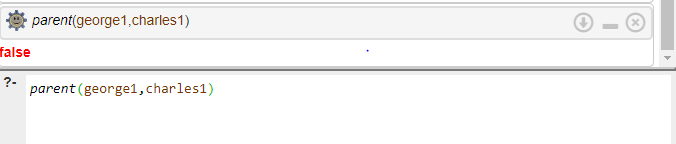
parent(george1, sophia).

QUESTION:

Formulate the following PROLOG queries:

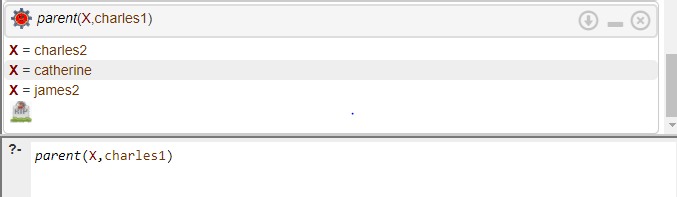
1. Was George I the parent of Charles I?

ANS:



1. Who was Charles I's parent?

ANS:



1. Who were the children of Charles I?

ANS:



|  |
| --- |
| Q5.  Recursion: Towers of Hanoi  The 3-disk setup is like this:  | | |  xxx | |  xxxxx | |  xxxxxxx | |  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    Here's a sample:  % move(N,X,Y,Z) - move N disks from peg X to peg Y, with peg Z being the  % auxilliary peg  %  % Strategy:  % Base Case: One disc - To transfer a stack consisting of 1 disc from  % peg X to peg Y, simply move that disc from X to Y  % Recursive Case: To transfer n discs from X to Y, do the following:  %Transfer the first n-1 discs to some other peg X  %Move the last disc on X to Y  % Transfer the n-1 discs from X to peg Y  move(1,X,Y,\_) :-  write('Move top disk from '),  write(X),  write(' to '),  write(Y),  nl.  move(N,X,Y,Z) :-  N>1,  M is N-1,  move(M,X,Z,Y),  move(1,X,Y,\_),  move(M,Z,Y,X).  - note the use of "anonymous" variables \_  Here is what happens when Prolog solves the case N=3.  ?- move(3,left,right,center).  Move top disk from left to right  Move top disk from left to center  Move top disk from right to center  Move top disk from left to right  Move top disk from center to left  Move top disk from center to right  Move top disk from left to right    yes |

ANS:

