Prolog Programming Assignment #1: Various Computations

Learning Abstract

This assignment was aimed to introduce us to Prolog programming. Doing these assignments, I got accustomed to the nature of Prolog and the syntax used in the programs. I was able to get a grasp of the queries and various terms while I went through the lessons to do this assignment. Now I have a basic knowledge on Prolog and I understand its elegance.

Task 1 - Colors KB

Colors KB Code

```
%-------------------------------------------------------------------------------
% File: coolors.pro
% Line: Six color facts, structured into primaries and secondaries
%-------------------------------------------------------------------------------
% primary(P) :: P is a primary color

primary(blue).
primary(red).
primary(yellow).


%-------------------------------------------------------------------------------
%primary(S) :: S is a secondary color

secondary(green).
secondary(orange).
secondary(purple).

%-------------------------------------------------------------------------------
% color(C) :: C is a color

color(C) :- primary(C).
color(C) :- secondary(C).
```

Colors KB Interaction

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.1.8)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- color(blue).
ERROR: Unknown procedure: color/1 (DWIM could not correct goal)
?- consult('/Users/LENOVO/Prolog/colors.pro').
true.

?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(yellow).
false.

?- color(blue).
true .

?- color(purple).
true.

?- primary(P).
P = blue ;
P = red ;
P = yellow.

?- secondary(S).
S = green ;
S = orange ;
S = purple.

?- color(C).
C = blue ;
C = red ;
C = yellow ;
C = green ;
C = orange ;
C = purple.

?- listing(primary).
primary(blue).
primary(red).
primary(yellow).

true.


?- listing(secondary).
secondary(green).
secondary(orange).
secondary(purple).

true.

?- listing(color).
color(C) :-
    primary(C).
color(C) :-
    secondary(C).

true.
```

Task 2 - Food KB

Food KB Code

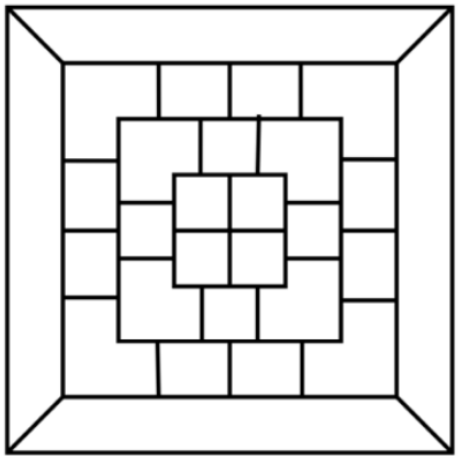
```
%-------------------------------------------------------------------------------
%File: foods.pro
%Line: Six food facts, structured into fruits and vegetables
%-------------------------------------------------------------------------------
% fruit(F) :: F is a fruit

fruit(grapefruit).
fruit(avocado).
fruit(date).

%-------------------------------------------------------------------------------

% vegetable(V) :: V is a vegetable

vegetable(asperagus).
vegetable(broccoli).
vegetable(carrot).

%-------------------------------------------------------------------------------
%food(FD) :: FD is a food

food(FD) :- fruit(FD).
food(FD) :- vegetable(FD).
```

Food KB Interaction

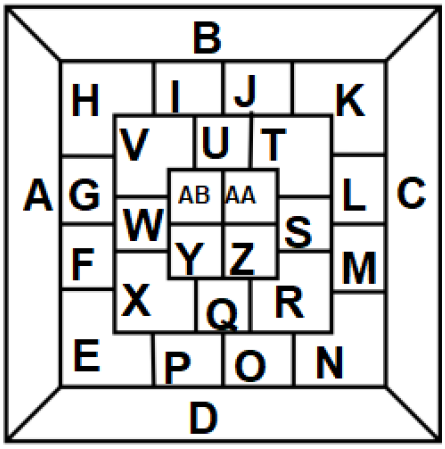
```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.1.8)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('/Users/LENOVO/Prolog/foods.pro').
true.

?- fruit(grapefruit).
true.

?- fruit(avocado).
true.

?- fruit(broccoli).
false.

?- vegetable(asperagus).
true.

?- vegetable(broccoli).
true.

?- vegetablr(date).
Correct to: "vegetable(date)"? yes
false.

?- food(date).
true .

?- food(carrot).
true.

?- fruit(F).
F = grapefruit ;
F = avocado ;
F = date.

?- vegetable(V).
V = asperagus ;
V = broccoli ;
V = carrot.

?- food(FD).
FD = grapefruit ;
FD = avocado ;
FD = date ;
FD = asperagus ;
FD = broccoli ;
FD = carrot.

?- listing(fruit).
fruit(grapefruit).
fruit(avocado).
fruit(date).

true.

?- listing(vegetable).
vegetable(asperagus).
vegetable(broccoli).
vegetable(carrot).

true.

?- listing(food).
food(FD) :-
    fruit(FD).
food(FD) :-
    vegetable(FD).

true.

?-
```

Task 3 - Map Coloring

The Given Map



The Labeled Map



Code for Coloring the Map

```
%-------------------------------------------------------------------------------
% File: map_coloring.pro
% Line: Program to find a 4 color map
% More: The colors will be red, blue, green, orange.
% More: The standard abbreviations are used to stand for the countries.
%-------------------------------------------------------------------------------
% different(X,Y) :: X is not equal to Y

different(red,blue).
different(red,green).
different(red,orange).
different(green,blue).
different(green,orange).
different(green,red).
different(blue,green).
different(blue,orange).
different(blue,red).
different(orange,blue).
different(orange,green).
different(orange,red).
%-------------------------------------------------------------------------------
% coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB)::
% The shapes that form the square are represented by numbers from 1 to 28 so that no
% shapes sharing borders are the same color.

coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB) :-
different(A,B),
different(A,H),
different(A,G),
different(A,F),
different(A,E),
different(A,D),
different(B,H),
different(B,I),
different(B,J),
different(B,K),
different(B,C),
different(C,K),
different(C,L),
different(C,M),
different(C,N),
different(C,D),
different(D,E),
different(D,P),
different(D,O),
different(D,N),
different(E,F),
```

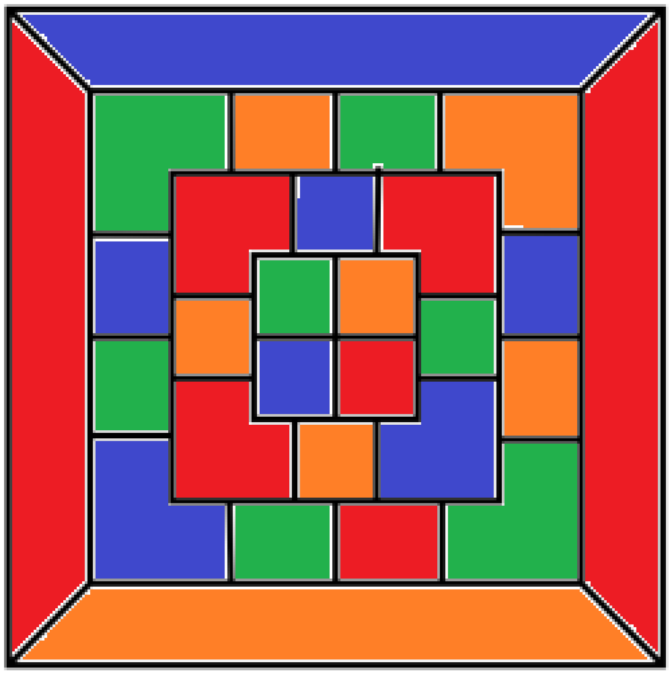
```
different(E,X),
different(E,P),
different(F,G),
different(F,X),
different(F,W),
different(G,H),
different(G,V),
different(G,W),
different(H,I),
different(H,V),
different(I,J),
different(I,U),
different(I,V),
different(J,K),
different(J,T),
different(J,U),
different(K,L),
different(K,T),
different(L,M),
different(L,S),
different(L,T),
different(M,N),
different(M,R),
different(M,S),
different(N,O),
different(N,R),
different(O,P),
different(O,Q),
different(O,R),
different(P,Q),
different(P,X),
different(Q,R),
different(Q,X),
different(Q,Y),
different(Q,Z),
different(R,S),
different(R,Z),
different(S,T),
different(S,Z),
different(S,AA),
different(T,U),
different(T,AA),
different(U,V),
different(U,AA),
different(U,AB),
different(V,W),
different(V,AB),
different(W,X),
different(W,Y),
```

```
different(W,AB),
different(X,Y),
different(Y,Z),
different(Y,AA),
different(Y,AB),
different(Z,AA),
different(Z,AB),
different(AA,AB).
```

Map Coloring Interaction

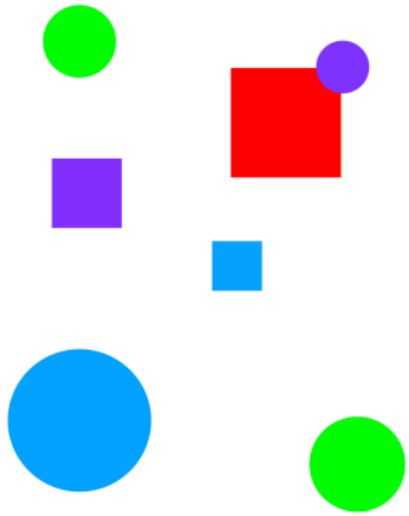
```
?- consult('/Users/LENOVO/Prolog/map_coloring.pro').
true.

?- coloring(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB).
A = C, C = O, O = T, T = V, V = X, X = Z, Z = red,
B = E, E = G, G = L, L = R, R = U, U = Y, Y = blue,
D = I, I = K, K = M, M = Q, Q = W, W = AA, AA = orange,
F = H, H = J, J = N, N = P, P = S, S = AB, AB = green ,

?-
```

The Colored Map



Task 4 - Floating Shapes World KB

Floating Shapes World Image



Floating Shapes World KB Code

```
squares :- square(Name,_,_), write(Name),nl,fail.
squares.

%-----------------------------------------------------------------------------
shapes :- circles,squares.

%-----------------------------------------------------------------------------
%blue(Name) :: Name is a blue shape

blue(Name) :- square(Name,_,color(blue)).
blue(Name) :- circle(Name,_,color(blue)).

%-----------------------------------------------------------------------------
% large(Name) :: Name is a large shape

large(Name) :- area(Name,A), A >= 100.

%-----------------------------------------------------------------------------
% small(Name) :: Name is a small shape

small(Name) :- area(Name,A), A < 100.

%-----------------------------------------------------------------------------
% area(Name,A) :: A is the area of the shape with name Name

area(Name,A) :- circle(Name,radius(R),_), A is 3.14 * R * R.
area(Name,A) :- square(Name,side(S),_), A is S * S.
```

Floating Shapes World KB Interaction

```
?- consult('/Users/LENOVO/Prolog/shapes_world.pro').
true.

?- listing(squares).
squares :-
    square(Name, _, _),
    write(Name),
    nl,
    fail.
squares.

true.

?- squares.
sera
sara
sarah
true.

?- listing(circles).
circles :-
    circle(Name, _, _),
    write(Name),
    nl,
    fail.
circles.

true.

?- circles.
carla
cora
connie
claire
true.

?- listing(shapes).
shapes :-
    circles,
    squares.

true.

?- shapes.
carla
cora
connie
claire
sera
sara
sarah
true.

?- blue(Shape).
Shape = sara ;
Shape = cora.

?- large(Name),write(Name),nl,fail.
cora
sarah
false.

?- small(Name),write(Name),nl,fail.
carla
connie
claire
sera
sara
false.

?- area(cora,A).
A = 153.86 .

?- area(carla,A).
A = 50.24 .

?-
```