Prolog Programming Assignment #1: Various Computations

Task 1 – Colors KB

Code:

```
% ----------------------------------------------------------------------
% File: colors.pro
% Line: Six color facts, structured into primaries and secondaries
% ----------------------------------------------------------------------
% primary(P) :: P is a primary color
primary(blue).
primary(red).
primary(yellow).
% ----------------------------------------------------------------------
% secondary(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:

```
1 ?- primary(blue).
ERROR: Unknown procedure: primary/1 (DWIM could not correct goal)
2 ?- consult("colors.pro").
3 ?- pwd.
```

% c:/users/ghent/

true.

4 ?- consult("colors.pro").

true.

5 ?- consult('colors.pro').

true.

6 ?- primary(blue).

true.

7 ?- primary(red).

true.

8 ?- primary(green).

false.

9 ?- secondary(green).

true.

10 ?- secondary(purple).

true.

11 ?- secondary(yellow).

false.

12 ?- color(blue).

true .

13 ?- color(purple).

true.

14 ?- primary(P).

P = blue ;

P = red ;

P = yellow.

15 ?- secondary(S).

S = green ;

S = orange ;

S = purple.

16 ?- color(C).

C = blue ;

C = red ;

C = yellow ;

C = green ;

C = orange ;

C = purple.

17 ?- listing(primary).

primary(blue).

primary(red).

primary(yellow).

true.

18 ?- listing(secondary).

secondary(green).

secondary(orange).

secondary(purple).

true.

19 ?- listing(color).

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

20 ?- halt
```

Task 2 – Food KB

Code:

```
% ----------------------------------------------------------------------
% File: foods.pro
% Line: Six foods, structured into fruits and vegtables
% ----------------------------------------------------------------------
% fruit(F) :: F is a fruit
fruit(grapefruit).
fruit(avacado).
fruit(date).
% ----------------------------------------------------------------------
% vegetable(V) :: V is a vegetable
vegetable(asparagus).
vegetable(broccoli).
vegetable(carrot).
% ----------------------------------------------------------------------
% Food(F) :: F is a food
food(F):- fruit(F);vegetable(F).
```

Food KB Interaction:

1 ?- fruit(grapefruit).

ERROR: Unknown procedure: fruit/1 (DWIM could not correct goal)

2 ?- consult('foods.pro').

true.

3 ?- fruit(grapefruit).

true

4 ?- fruit(date).

true.

5 ?- fruit(carrot).

false.

6 ?- vegetable(carrot).

true.

7 ?- vegetable(broccoli).

true.

8 ?- food(asparagus).

true.

9 ?- food(avacado).

true ;

false.

10 ?- fruit(F).

F = grapefruit ;

F = avacado ;

F = date.

11 ?- vegetable(V).

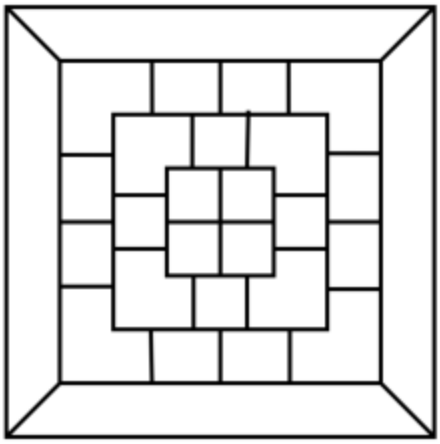
V = asparagus ;

V = broccoli ;

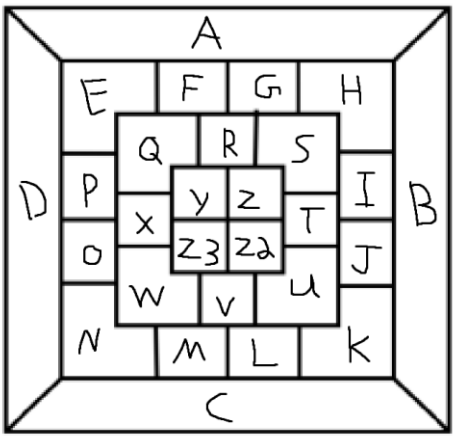
```
V = carrot.
12 ?- food(F).
F = grapefruit ;
F = avacado ;
F = date ;
F = asparagus ;
F = broccoli ;
F = carrot.
13 ?- listing(fruit).
fruit(grapefruit).
fruit(avacado).
fruit(date).
true.
14 ?- listing(vegetable).
vegetable(asparagus).
vegetable(broccoli).
vegetable(carrot).
true.
15 ?- listing(food).
food(F) :-
    (   fruit(F)
    ;   vegetable(F)
    ).
true.
16 ?- halt.
```

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 rendering for the given map.

% More: The colors used will be red, blue, green orange.

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

% 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,Z2,Z3) :-
different(A,B),
different(A,D),
different(A,H),
different(A,G),
different(A,F),
different(A,E),
different(B,H),
different(B,I),
different(B,J),
different(B,K),
different(C,B),
different(C,D),
different(C,K),
different(C,L),
```

different(C,M),

different(C,N),

different(D,E),

different(D,P),

different(D,O),

different(D,N),

different(E,F),

different(E,Q),

different(E,P),

different(F,R),

different(F,Q),

different(F,G),

different(G,S),

different(G,R),

different(H,I),

different(H,G),

different(H,S),

different(I,S),

different(I,T),

different(I,J),

different(J,T),

different(J,U),

different(K,J),

different(K,U),

different(K,L),

different(L,V),

different(L,U),

different(M,W),

different(M,V),

different(M,L),

different(N,O),

different(N,W),

different(N,M),

different(O,X),

different(O,W),

different(P,Q),

different(P,X),

different(P,O),

different(T,S),

different(T,Z),

different(T,Z2),

different(T,U),

different(R,S),

different(R,Z),

different(R,Y),

different(R,Q),

different(X,Q),

different(X,Y),

different(X,Z3),

different(X,W),

different(V,W),

different(V,Z3),

different(V,Z2),

different(V,U),

different(U,Z2),

different(W,Z3),

different(Y,Q),

different(Z,Y),

different(Z,Z2),

different(Z,Z3),

different(Z,S),

different(Z3,Y),

different(Z3,Z2),

different(Z2,Y).

Map Coloring Interaction

1 ?- consult("mapcolors.pro").

true.

2 ?- 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,Z2,Z3).

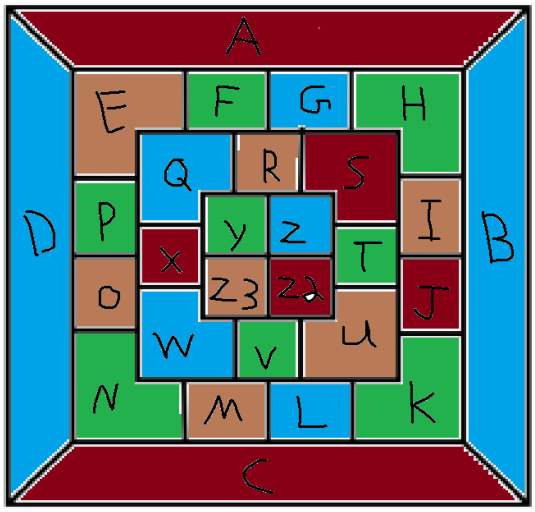
A = C, C = O, O = Q, Q = S, S = U, U = Z3, Z3 = red,

B = D, D = G, G = L, L = T, T = W, W = Y, Y = blue,

E = I, I = K, K = M, M = R, R = X, X = Z2, Z2 = orange,

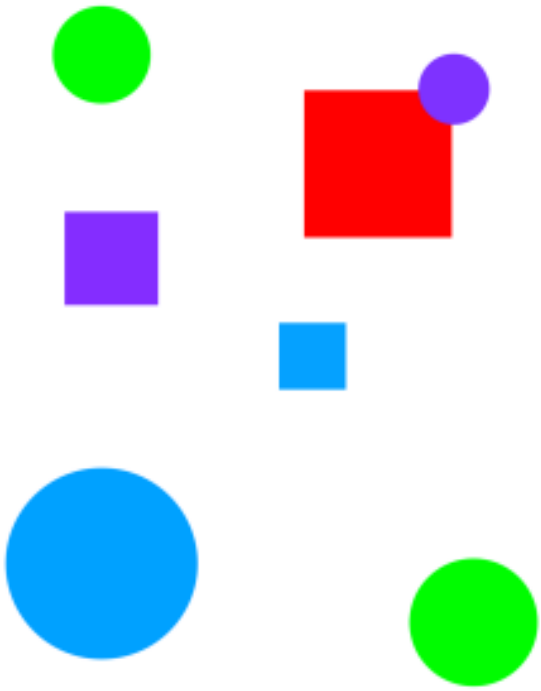
F = H, H = J, J = N, N = P, P = V, V = Z, Z = green .

The Colored Map



Task 4 – Floating Shapes World Image

Floating Shapes World Image



Floating Shapes World KB Code

```
% ----------------------------------------------------------------------
% --- File: shapes_world_1.pro
% --- Line: Loosely represented 2-D shapes world (simple take on SHRDLU)
% ----------------------------------------------------------------------
% ----------------------------------------------------------------------
% --- Facts ...
% ----------------------------------------------------------------------
% ----------------------------------------------------------------------
% --- square(N,side(L),color(C)) :: N is the name of a square with side L
% --- and color C
square(sera,side(7),color(purple)).
square(sara,side(5),color(blue)).
square(sarah,side(11),color(red)).
% ----------------------------------------------------------------------
% --- circle(N,radius(R),color(C)) :: N is the name of a circle with
% --- radius R and color C
circle(carla,radius(4),color(green)).
circle(cora,radius(7),color(blue)).
circle(connie,radius(3),color(purple)).
circle(claire,radius(5),color(green)).
% ----------------------------------------------------------------------
% Rules ...
% ----------------------------------------------------------------------
% ----------------------------------------------------------------------
```

```
% --- circles :: list the names of all of the circles
circles :- circle(Name,_,_), write(Name),nl,fail.
circles.
% ---------------------------------------------------------------------
% --- squares :: list the names of all of the squares
squares :- square(Name,_,_), write(Name),nl,fail.
squares.
% ---------------------------------------------------------------------
% --- squares :: list the names of all of the shapes
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

```
1 ?- consult("shapes_world_1.pro").
true.
2 ?- listing(squares).
squares :-
    square(Name, _, _),
    write(Name),
    nl,
    fail.
squares.
true.
3 ?- squares.
sera
sara
sarah
true.
4 ?- listing(circles).
circles :-
    circle(Name, _, _),
    write(Name),
    nl,
    fail.
circles.
true.
5 ?- circles.
carla
cora
```

```
connie
claire
true.
6 ?- listing(shapes).
shapes :-
    circles,
    squares.
true.
7 ?- shapes.
carla
cora
connie
claire
sera
sara
sarah
true.
8 ?- blue(Shape).
Shape = sara
Unknown action: l (h for help)
Action? .
9 ?- large(Name),write(Name),nl,fail.
cora
sarah
false.
10 ?- small(Name),write(Name),nl,fail.
carla
```

connie

claire

sera

sara

false.

11 ?- area(cora,A).

A = 153.86 .

12 ?- area(carla, A).

A = 50.24 .