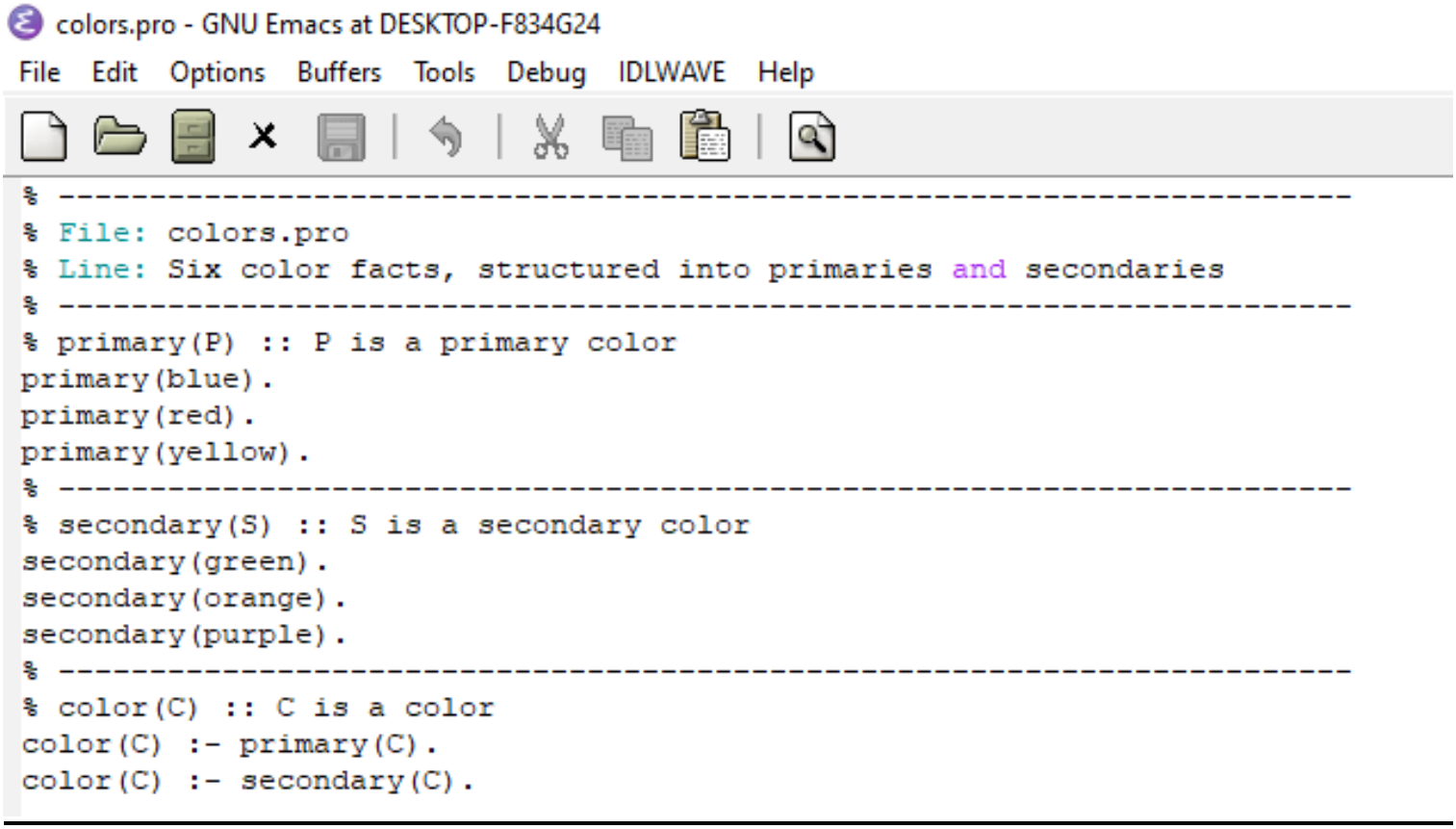
Task 1 – Colors KB:

Colrs KB Code



Colors KB Interaction

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?- primary(blue).
ERROR: Unknown procedure: primary/1 (DWIM could not correct goal)
?- consult('C:/Users/User/Documents/Prolog/prolog/colors.pro').
true.

?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(purple).
true.

?- secondary(yellow).
false.

?- color(blue).
true
Unknown action: □ (h for help)
Action?
Unknown action: □ (h for help)
Action?
Unknown action: □ (h for help)
Action? ,

?- color(purple).
true.
```

```
?- primary(P).
P = blue ,

?- 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.

?- halt.
```

Task 2 – Food KB:

Food KB Code

foods.pro - GNU Emacs at DESKTOP-F834G24

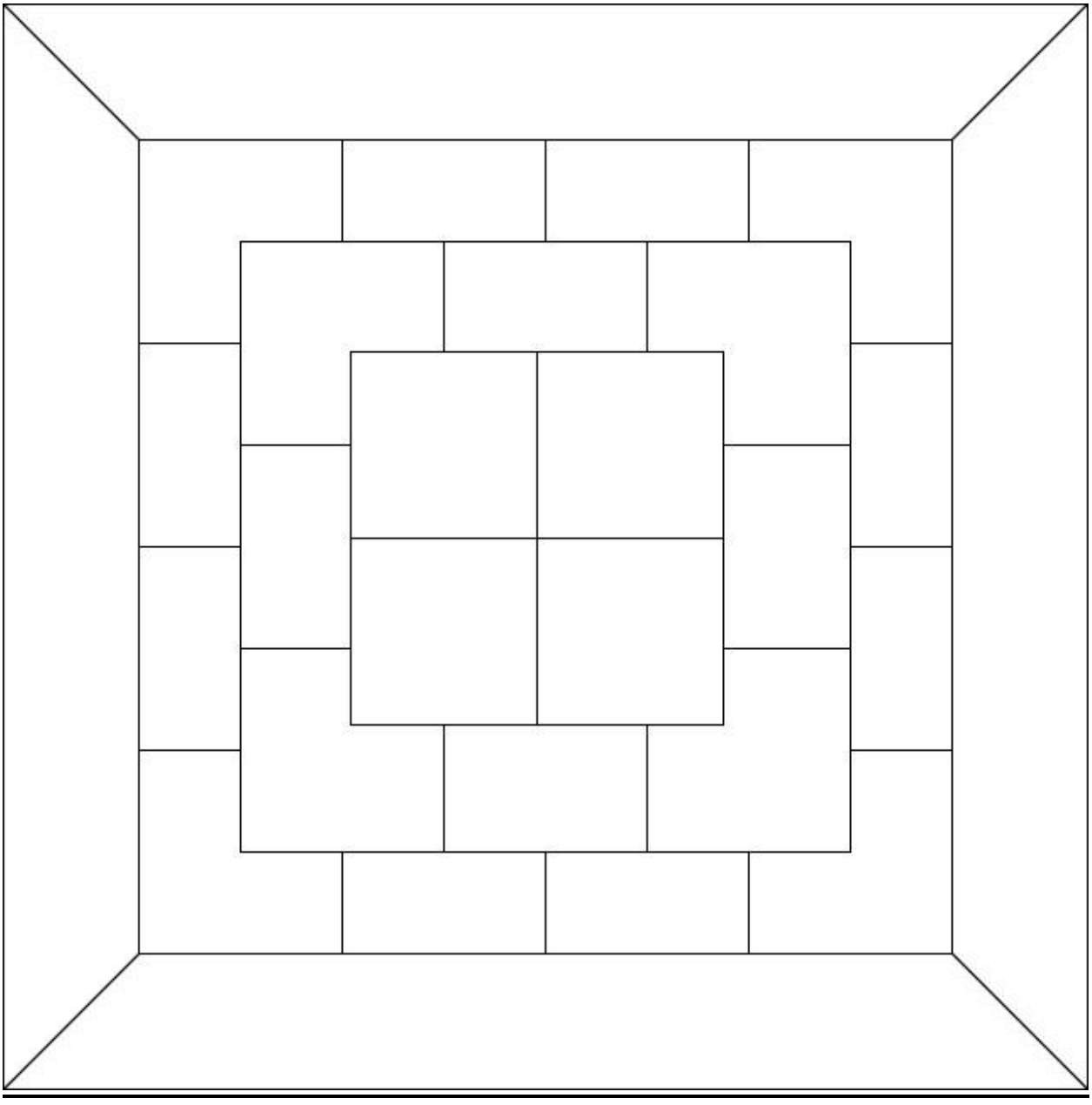
File Edit Options Buffers Tools Debug IDLWAVE Help

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

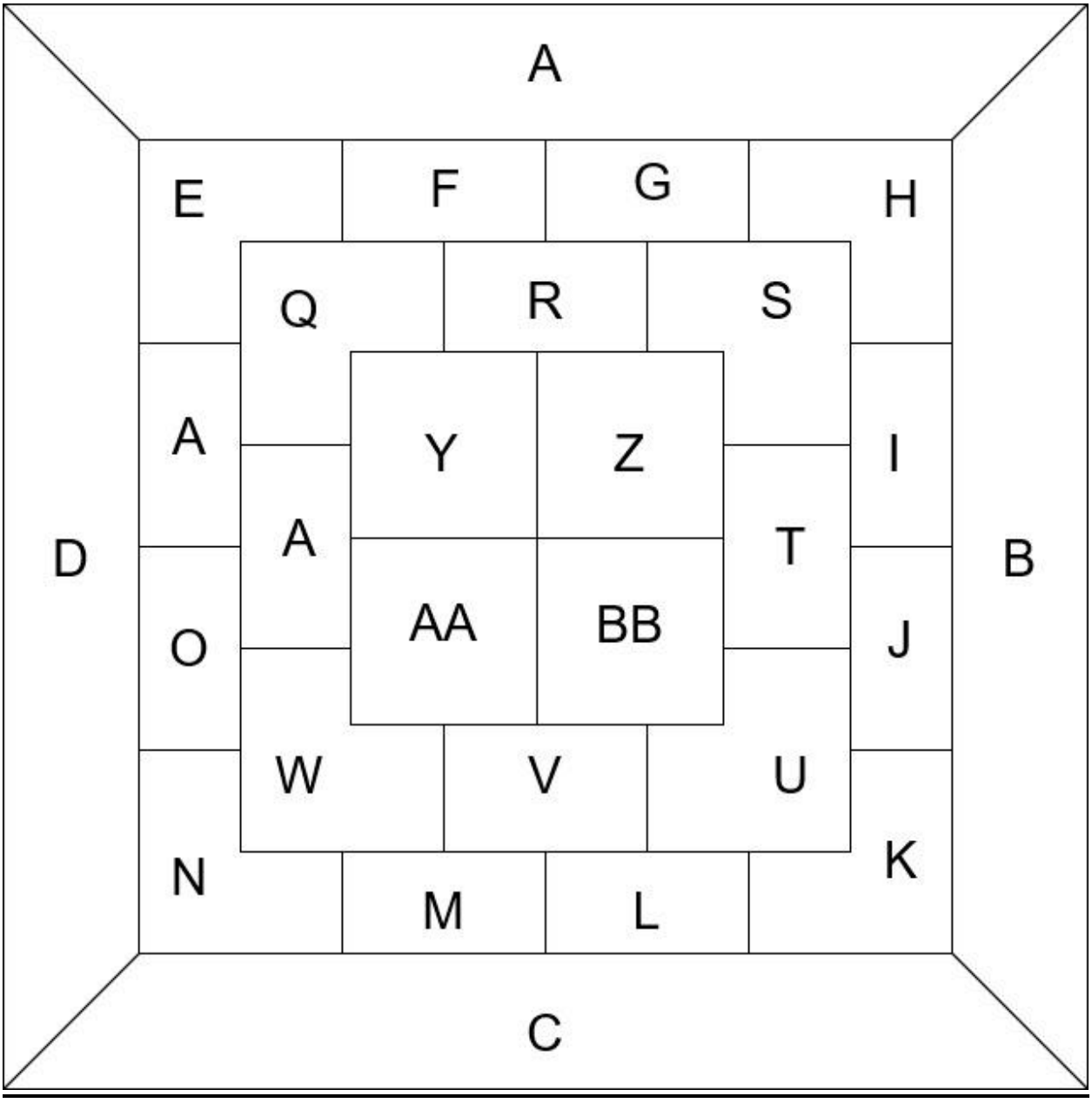
Food KB Interaction

Task 3 – Map Coloring

The Given Map



The Labeled Map



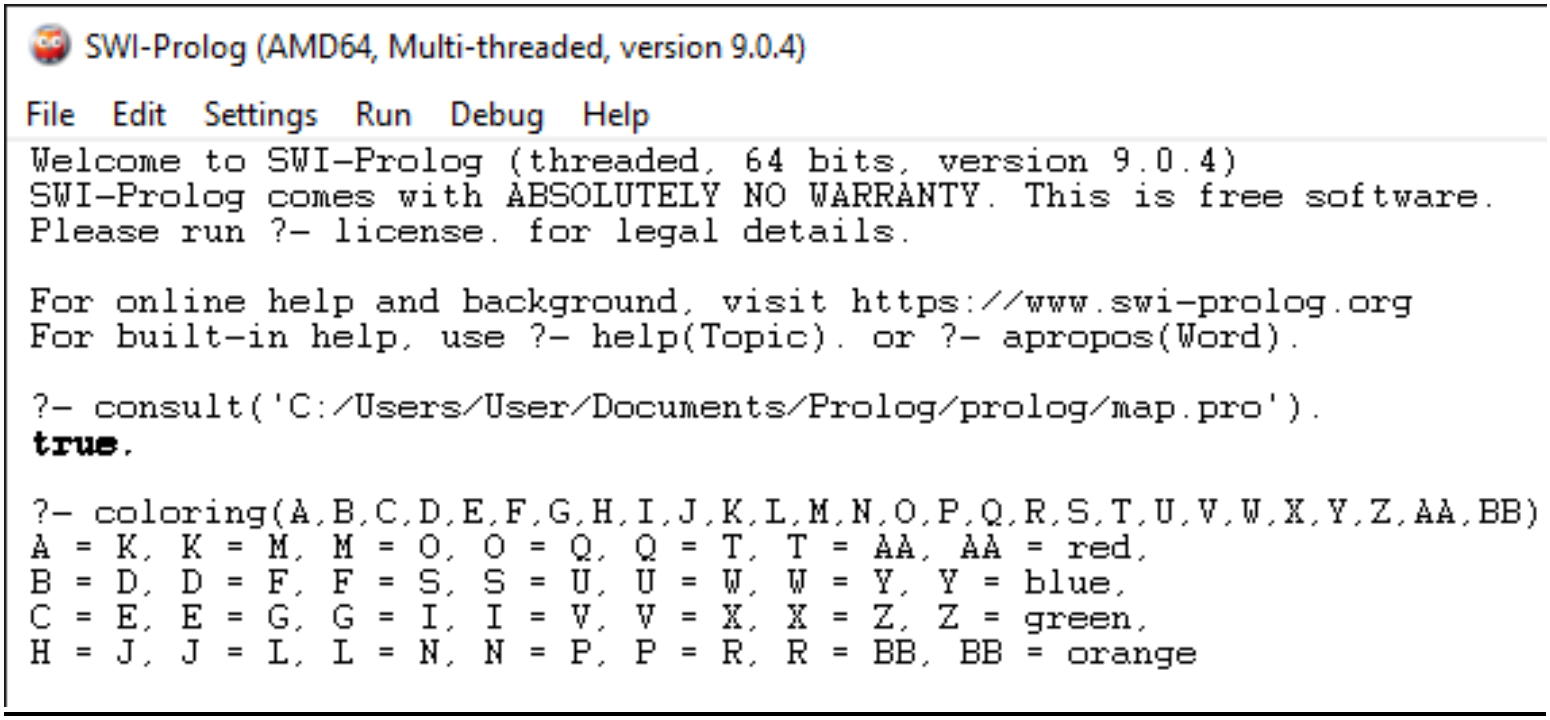
Code

map.pro - GNU Emacs at DESKTOP-F834G24

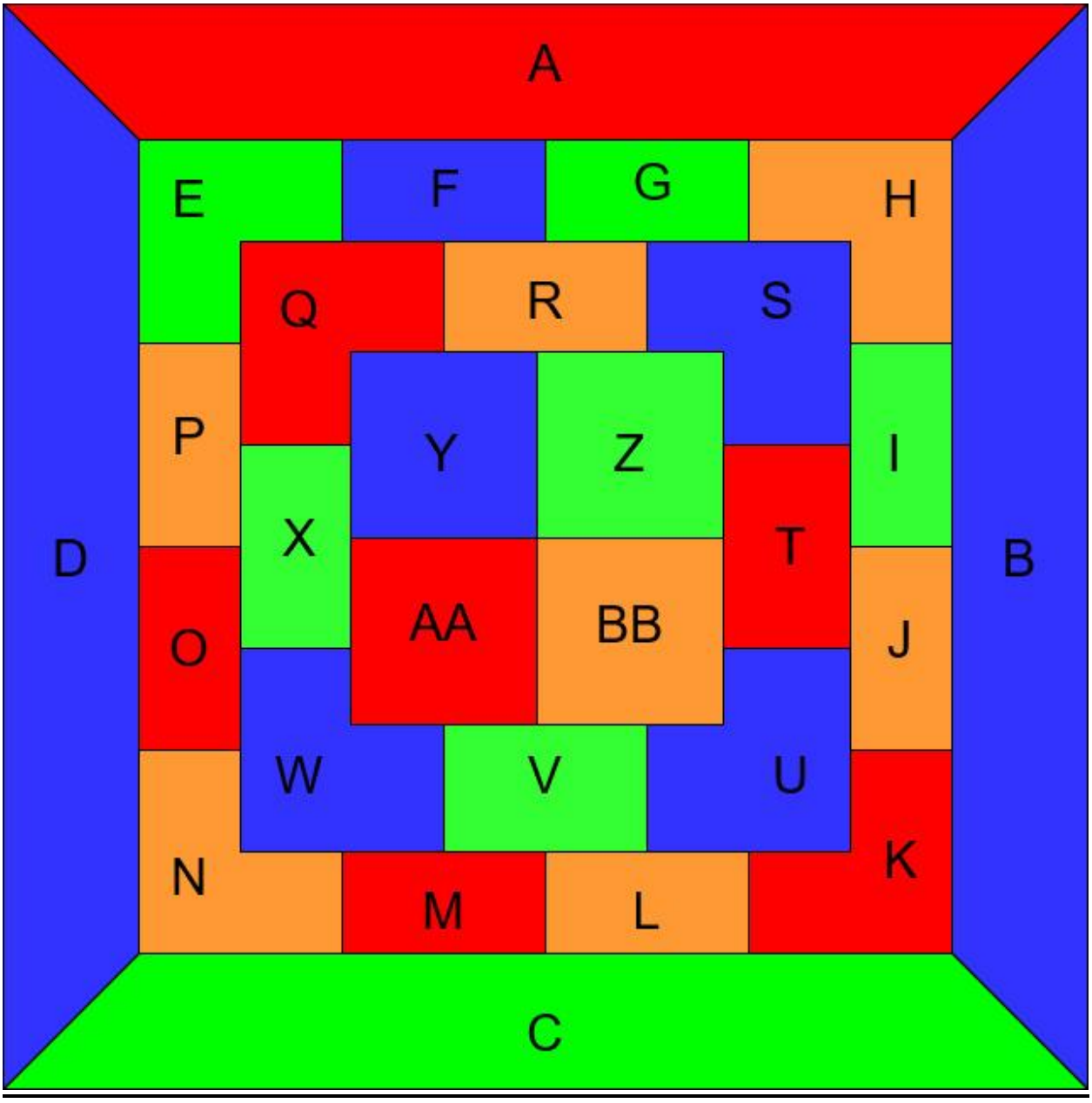
File Edit Options Buffers Tools Debug IDLWAVE Help

```
% ------------------------------------------------------------------------
% File: map_coloring.pro
% Line: Program to find a 4 color map rendering for prolog assinment map.
% More: The colors used will be red, blue, green orange.
% More: Letters of the alphabet are used to stand for sections of the map.
% ------------------------------------------------------------------------
% 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,BB) :: PL_Assignment_1 Map
% Map sections are labled with a unique letter and colored
% so that none of the sections sharing a border 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,BB) :-

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

Map Coloring Interaction

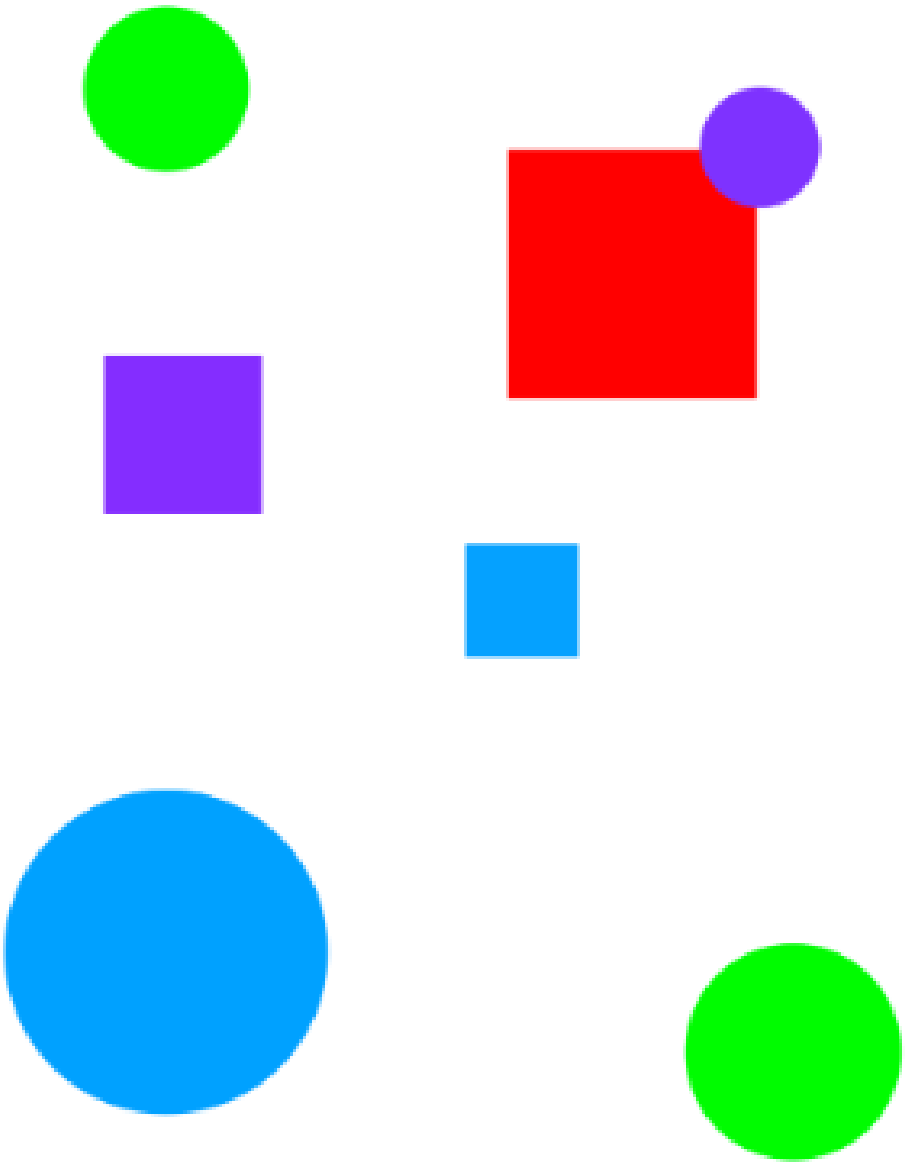

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- consult('C:/Users/User/Documents/Prolog/prolog/map.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,BB)
A = K, K = M, M = O, O = Q, Q = T, T = AA, AA = red,
B = D, D = F, F = S, S = U, U = W, W = Y, Y = blue,
C = E, E = G, G = I, I = V, V = X, X = Z, Z = green,
H = J, J = L, L = N, N = P, P = R, R = BB, BB = orange

The Colored Map



Task 4 – Floating Shapes World KB:

Floating Shapes World Image



Floating Shapes World KB Code

shapes_world_1.pro - GNU Emacs at DESKTOP-F834G24

File Edit Options Buffers Tools Debug IDLWAVE Help

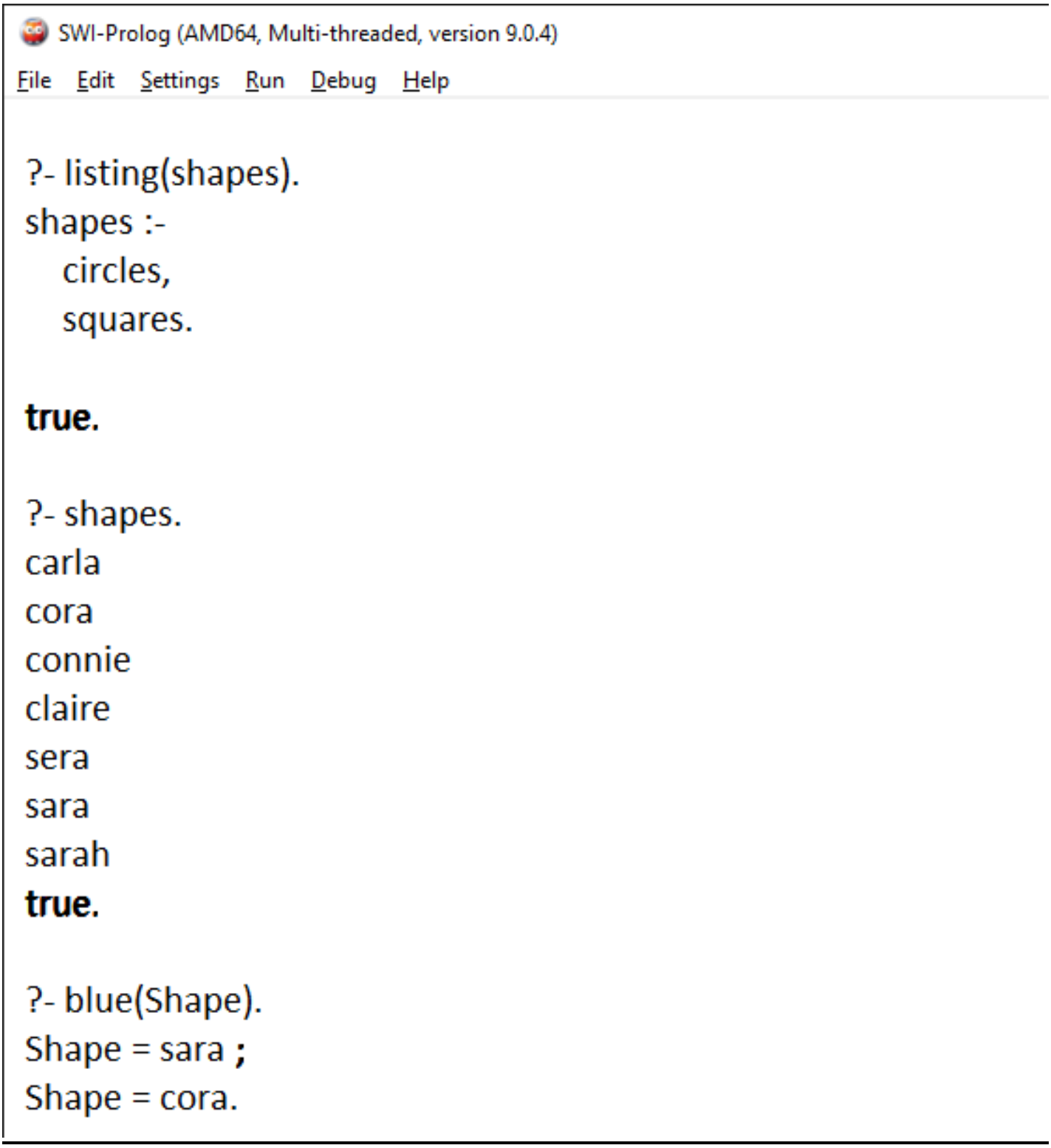
```
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% --- File: shapes_world_1.pro
% --- Line: Loosely represented 2-D shapes world (simple take on SHRDLU)
% ------------------------------------------------------------------------
% ------------------------------------------------------------------------
% --- 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 Interactions

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

File Edit Settings Run Debug Help

```
?- consult('C:/Users/User/Documents/Prolog/prolog/shapes_world_1.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.
```


SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
?- listing(shapes).
shapes :-
circles,
squares.
true.
?- shapes.
carla
cora
connie
claire
sera
sara
sarah
true.
?- blue(Shape).
Shape = sara ;
Shape = cora.

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

File Edit Settings Run Debug Help

```
?- 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 ;
false.

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

?-
```