Title: Prolog Assignment #1: Various Computations

Abstract:

This assignment is merely the *prologue* of things yet to come…
ba-DumTSSSsss!!!

Task 1: //note: I did not demo 'halt.' because it automatically Xs out my editor window, meaning I can no longer screenshot it for the demo.

Demo:

```
Welcome to SWI-Prolog (threaded, 64
bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO
WARRANTY. This is free software.
Please run ?- license. for legal det
ails.

For online help and background, visi
t 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('colors.pro').
true.

?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(purple).
true.

?- secondary(yellow).
false.

?- color(blue).
true .

?- olor(purple).
Correct to: "color(purple)"? yes
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.

?-
```

Code:

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

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

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

Task 2: //note: spelling mistake: avocado not avocado

Demo:

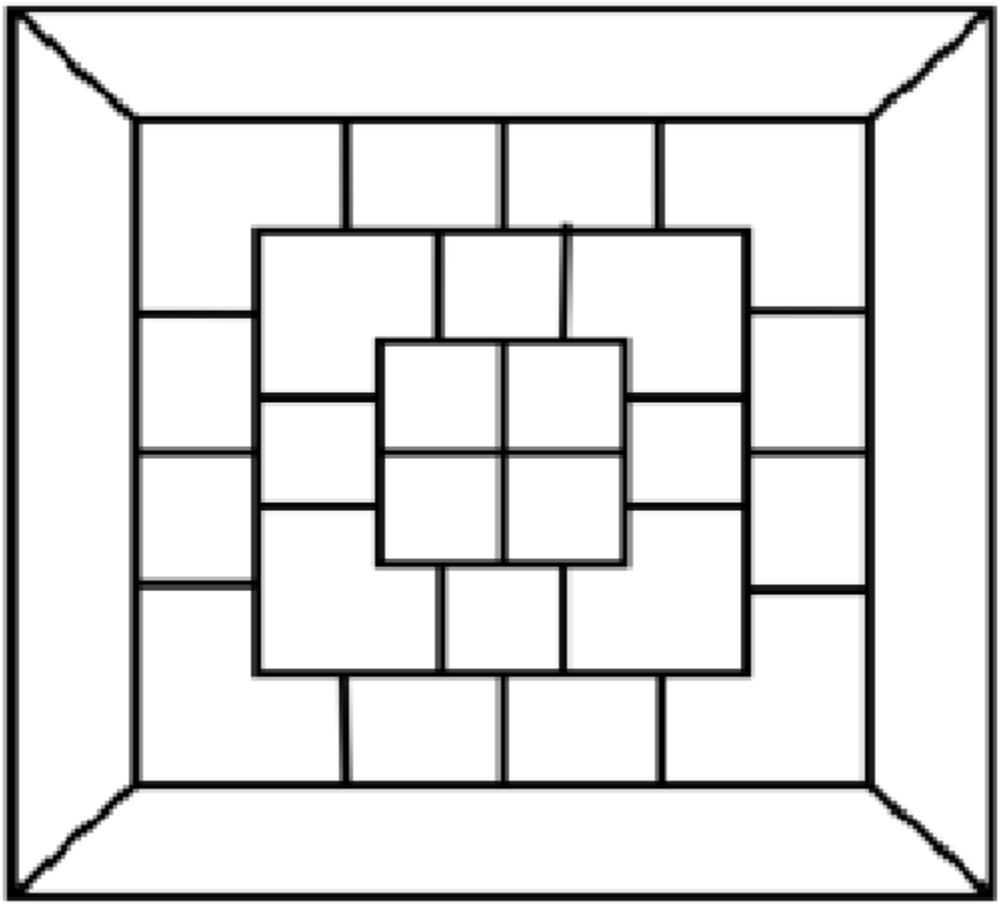
```
?- fruit(grapefruit).
ERROR: Unknown procedure: fruit/1 (D
WIN could not correct goal)
?- consult('foods.pro').
true.

?- fruit(grapefruit).
true.

?- fruit(date).
true.

?- fruit(carrot).
false.

?- vegetable(carrot).
true.

?- vegetable(broccoli).
true.

?- food(asparagus).
true.

?- food(avacado).
true
Unknown action: l (h for help)
Action?
Unknown action: (h for help)
Action?
Unknown action: (h for help)
Action? ,

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

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

?- food(F).
F = grapefruit ;
F = avacado ;
F = date ;
F = asparagus ;
F = broccoli ;
F = carrot.

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

true.

true.

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

true.

?- listing(food).
food(F) :-
    (   fruit(F)
    ;   vegetable(F)
    ).

true.

?-
```

Code:

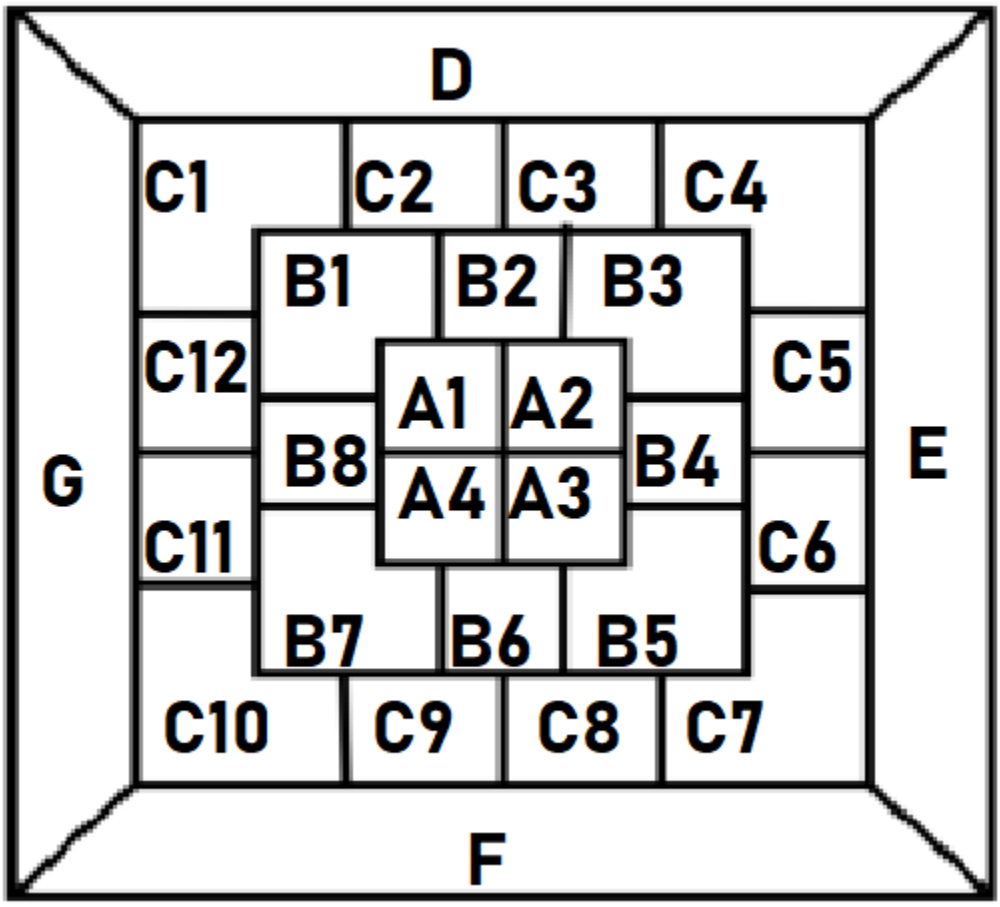
```
fruit(grapefruit).
fruit(avacado).
fruit(date).

vegetable(asparagus).
vegetable(broccoli).
vegetable(carrot).

food(F) :- fruit(F); vegetable(F).
```

Task 3:

Given Map:



Labeled Map:



Code:

```
d(red,blue).
d(red,green).
d(red,orange).
d(green,blue).
d(green,orange).
d(green,red).
d(blue,green).
d(blue,orange).
d(blue,red).
d(orange,blue).
d(orange,green).
d(orange,red).
%d(A,B) :- d(B,A).
```

color(A1,A2,A3,A4,B1,B2,B3,B4,B5,B6,B7,B8,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,
D,E,F,G):-

%A SPOKES
d(A1,A2),
d(A2,A3),
d(A3,A4),
d(A4,A1),

%A RIM
d(A1,B8),
d(A1,B1),
d(A1,B2),
d(A2,B2),
d(A2,B3),
d(A2,B4),
d(A3,B4),
d(A3,B5),
d(A3,B6),
d(A4,B6),
d(A4,B7),
d(A4,B8),

%B SPOKES
d(B1,B2),
d(B2,B3),
d(B3,B4),
d(B4,B5),
d(B5,B6),
d(B6,B7),
d(B7,B8),
d(B8,B1),

%B RIM
d(B1,C12),
d(B1,C1),
d(B1,C2),
d(B2,C2),
d(B2,C3),
d(B3,C3),

d(B3,C4),
d(B3,C5),
d(B4,C5),
d(B4,C6),
d(B5,C6),
d(B5,C7),
d(B5,C8),
d(B6,C8),
d(B6,C9),
d(B7,C9),
d(B7,C10),
d(B7,C11),
d(B8,C11),
d(B8,C12),

%C SPOKES
d(C1,C2),
d(C2,C3),
d(C3,C4),
d(C4,C5),
d(C5,C6),
d(C6,C7),
d(C7,C8),
d(C8,C9),
d(C10,C11),
d(C11,C12),
d(C12,C1),

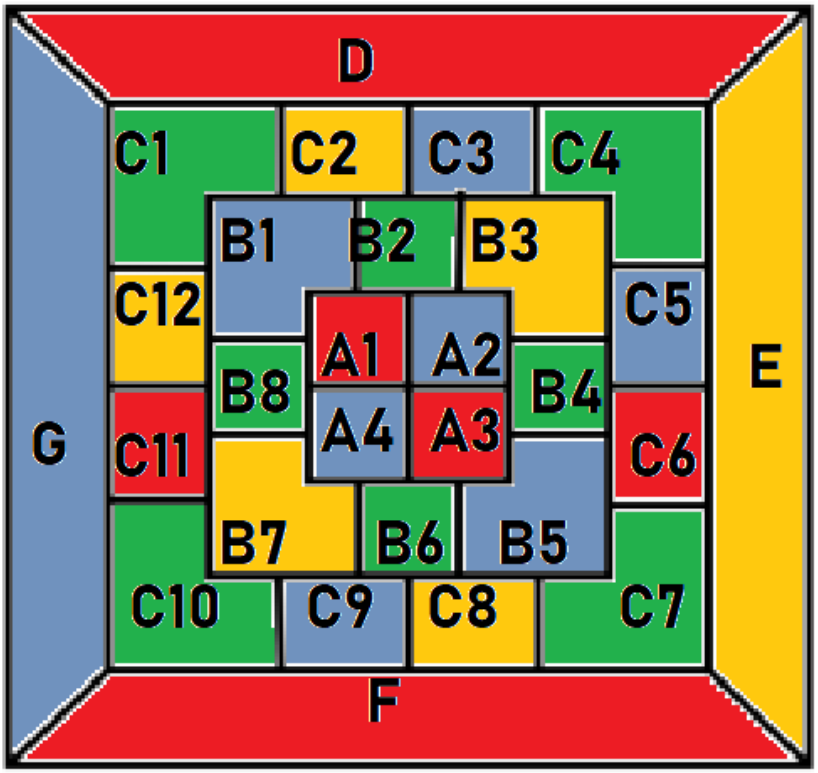
%C RIM
d(D,C1),
d(D,C2),
d(D,C3),
d(D,C4),
d(E,C4),
d(E,C5),
d(E,C6),
d(E,C7),
d(F,C7),
d(F,C8),
d(F,C9),
d(F,C10),

```
d(G,C10),
d(G,C11),
d(G,C12),
d(G,C1),

%DEFG SPOKES
d(D,E),
d(E,F),
d(F,G),
d(G,D).
```

Demo:

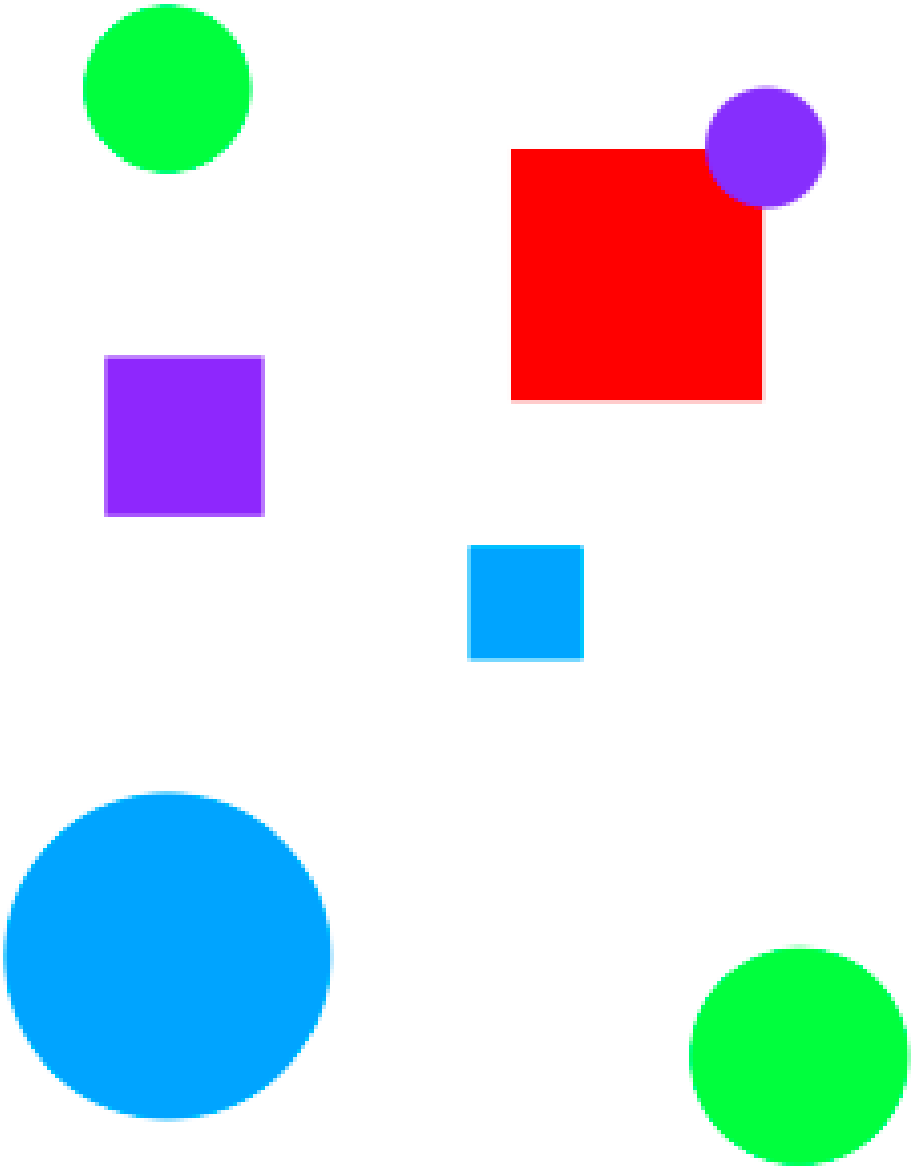
```
?- consult('map.pro').
true.

?- color(A1,A2,A3,A4,B1,B2,B3,B4,B5,B6,B7,B8,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,D,E,F,G).
% c:/Users/tgerm/Documents/Prolog/map.pro compiled 0.00 sec, 5 clauses
A1 = A3, A3 = C6, C6 = C11, C11 = D, D = F, F = red,
A2 = A4, A4 = B1, B1 = B5, B5 = C3, C3 = C5, C5 = C9, C9 = G, G = blue,
B2 = B4, B4 = B6, B6 = B8, B8 = C1, C1 = C4, C4 = C7, C7 = C10, C10 = green,
B3 = B7, B7 = C2, C2 = C8, C8 = C12, C12 = E, E = orange
```

Colored Map:



Task 4:

Le Imagé:



Demo:

```
?- consult('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 ,

?-
```

Code:

```
square(sera,side(7),color(purple)).
square(sara,side(5),color(blue)).
square(sarah,side(11),color(red)).

circle(carla,radius(4),color(green)).
circle(cora,radius(7),color(blue)).
circle(connie,radius(3),color(purple)).
circle(claire,radius(5),color(green)).

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

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

shapes :- circles,squares.

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

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

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