Abstract:

This assignment introduced me to the basics of Prolog, and through completing tasks such as replicating demos, generating food facts, and coloring a map with non-adjacent colors, I gained a solid understanding of the language's fundamentals.

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).
```

Demo:

```
?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(purple).
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

Code:

```
%--------------------------------------
%File: foods.pro
%Line: Six food facts
%--------------------------------------
%fruit(F) :: F is a fruit food
fruit(grapefruit).
fruit(avocado).
fruit(date).
%--------------------------------------
%vegetable(V) :: V is a vegetable food
vegetable(asperagus).
vegetable(broccoli).
vegetable(carrot).
%--------------------------------------
%food(P) :: P is a food
food(P) :- fruit(P).
food(P) :- vegetable(P).
```

Demo:

```
?- fruit(grapefruit).
true.

?- fruit(avocado).
true.

?- fruit(date)
|    .
true.

?- vegetable(asperagus).
true.

?- vegetable(broccoli).
true.

?- vegetable(carrot).
true.

?- fruit(carrot).
false.

?- vegetable(date).
false.

?- food(carrot).
true.

?- food(avocado).
true
```

Task 3:

Code:

```
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, A2, F, G, H, I, J, K, L, M, N, O, P,
 Q, R, S, T, U, V, W, X, Y, Z, A1) :-
different(A,B),
different(A,C),
different(A,D),
different(A,E),
different(A,A2),
different(A,F),
different(B,A),
different(B,A1),
different(B,C),
different(B,G),
different(B,L),
different(B,R),
different(A1,B),
different(A1,R),
different(A1,Y),
different(A1,Z),
different(A1,W),
different(A1,F),
different(F,A),
different(F,A2),
different(F,K),
different(F,Q),
different(F,W),
different(F,A1),
different(C,A),
different(C,D),
different(C,H),
```

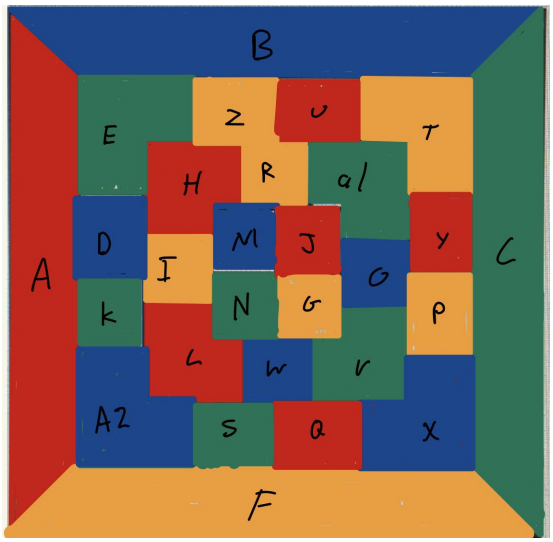
```
different(C,G),
different(C,B),
 different(D,A),
different(D,E),
different(D,I),
different(D,H),
different(D,C),
different(E,A),
different(E,A2),
different(E,J),
different(E,I),
different(E,D),
different(A2,A),
different(A2,F),
different(A2,K),
different(A2,J),
different(A2,E),
different(K,F),
different(K,Q),
different(K,P),
different(K,J),
different(K,A2),
different(Q,K),
different(Q,F),
different(Q,W),
different(Q,V),
different(Q,P),
different(W,Q),
different(W,F),
different(W,A1),
different(W,Z),
different(W,V),
different(Z,V),
different(Z,W),
different(Z,A1),
different(Z,Y),
different(Z,X),
different(Y,X),
different(Y,Z),
different(Y,A1),
different(Y,R),
different(Y,S),
different(R,L),
different(R,S),
```

```
different(R,Y),
different(R,A1),
different(R,B),
different(L,G),
different(L,M),
different(L,S),
different(L,R),
different(L,B),
different(G,C),
different(G,H),
different(G,M),
different(G,L),
different(G,B),
different(H,C),
different(H,D),
different(H,I),
different(H,N),
different(H,M),
different(H,G),
different(I,D),
different(I,E),
different(I,J),
different(I,O),
different(I,N),
different(I,H),
different(J,E),
different(J,A2),
different(J,K),
different(J,P),
different(J,O),
different(J,I),
different(P,J),
different(P,K),
different(P,Q),
different(P,V),
different(P,U),
different(P,O),
different(V,P),
different(V,Q),
different(V,W),
 different(V,Z),
 different(V,X),
different(V,U),
different(X,U),
```

```
different(X,V),
different(X,Z),
different(X,Y),
different(X,S),
different(X,T),
different(S,M),
different(S,T),
different(S,X),
different(S,Y),
different(S,R),
different(S,L),
different(M,H),
different(M,N),
different(M,T),
different(M,S),
different(M,L),
different(M,G),
different(N,H),
different(N,I),
different(N,O),
different(N,U),
different(N,T),
different(N,M),
different(O,I),
different(O,J),
different(O,P),
different(O,U),
different(O,T),
different(O,N),
different(U,O),
different(U,P),
different(U,V),
different(U,X),
different(U,T),
different(U,N),
different(T,N),
different(T,O),
different(T,U),
 different(T,X),
different(T,S),
different(T,M).
```

Demo:

```
?- coloring(A, B, C, D, E, A2, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y,
Z, A1).
A = H, H = J, J = L, L = Q, Q = U, U = Y, Y = red,
B = D, D = A2, A2 = M, M = O, O = W, W = X, X = blue,
C = E, E = K, K = N, N = S, S = V, V = A1, A1 = green,
F = G, G = I, I = P, P = R, R = T, T = Z, Z = orange
```



Task 4: The floating shapes world

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.
```

Demo:

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

true.

?- squares.
sara
sarah
true.

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

true.

?- circles.
carla
cora
connie
claire
true.

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

true.

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

?- shapes.
carla
cora
connie
claire
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
sara
false.


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

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