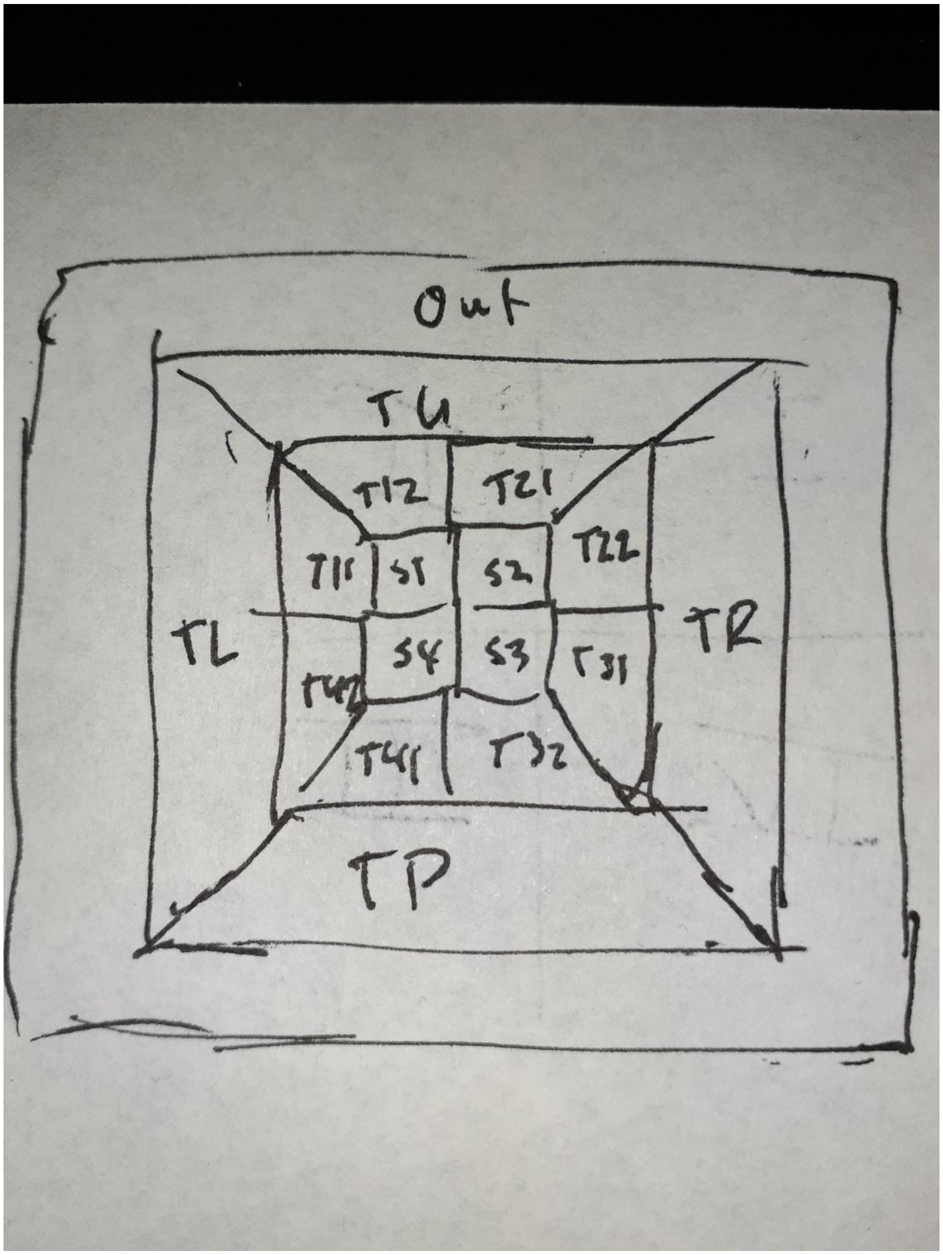
Prolog Programming Assignment # 1: Various Computations

Abstract: Knowledge representation, search and list processing problems to be solved in Prolog. This assignment has 4 tasks.

Interaction: Task 1 Map Coloring

1. 

2.

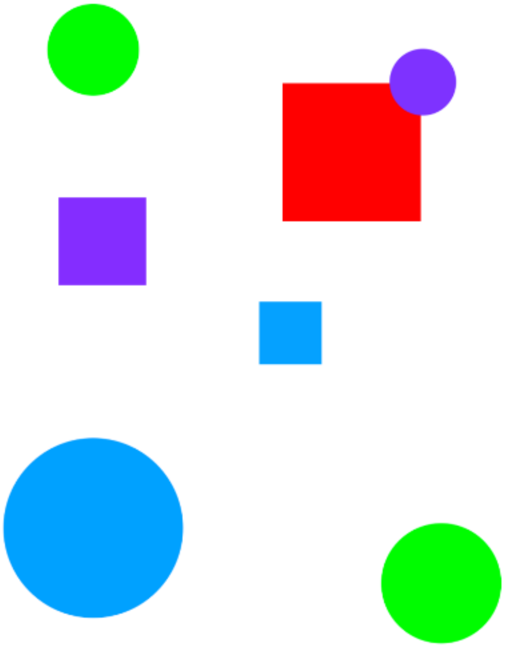
```
%------------------
  % File: map_coloring.pro
  % Line: Program to find a 4 color map rendering assignment 1
  %------------------

  %different(X,Y) :: X is not equal to Y

  different(red,blue).
  different(red,green).
  different(red,yellow).
  different(blue,red).
  different(blue,green).
  different(blue,yellow).
  different(green,blue).
  different(green,red).
  different(green,yellow).
  different(yellow,blue).
  different(yellow,green).
  different(yellow,red).



  %-----------------------
  %
coloring(s1,s2,s3,s4,t11,t12,t21,t22,t31,t32,t41,t42,tl,tu,td,tr,out)
:: The shapes are colored so that none of the shapes share
  % a same colored border.


coloring(S1,S2,S3,S4,T11,T12,T21,T22,T31,T32,T41,T42,TL,TU,TD,TR,OUT):
-
  different(S1, S2),
  %different(S1, S3),
  different(S1, S4),
  different(S1, T11),
  different(S1, T12),
  %different(S1, T21),
  %different(S1, T42),
  different(S2, S3),
  different(S2, S4),
  different(S2, T21),
  different(S2, T22),
  %different(S2, T12),
  %different(S2, T31),
  different(S3, S4),
  different(S3, T31),
  different(S3, T32),
  %different(S3, T22),
  %different(S3, T41),
  different(S4, T41),
  different(S4, T42),
  %different(S4, T11),
  %different(S4, T32),
  different(T11, T12),
  different(T11, T42),
  different(T11, TL),
  %different(T11, TU),
  %different(T12, TL),
  different(T12, TU),
  different(T12, T21),
  different(T21, TU),
  %different(T21, TR),
  different(T21, T22),
  %different(T22, TU),
  different(T22, TR),
  different(T22, T31),
  %different(T31, TD),
  different(T31, TR),
  different(T31, T32),
  different(T32, TD),
  %different(T32, TR),
  different(T32, T41),
  %different(T41, TL),
  different(T41, TD),
  different(T41, T42),
  different(T42, TL),
  %different(T42, TD),
  different(TL, TU),
  different(TL, TD),
  different(TL, OUT),
  different(TU, TR),
  different(TU, OUT),
  different(TR, TD),
  different(TR, OUT),
  different(TD, OUT).
```

3.

```
?- coloring(S1,S2,S3,S4,T11,T12,T21,T22,T31,T32,T41,T42,TL,TU,TD,TR,OUT).
S1 = S3, S3 = T21, T21 = T42, T42 = TR, TR = red,
S2 = T11, T11 = T31, T31 = T41, T41 = OUT, OUT = blue,
S4 = T12, T12 = T22, T22 = T32, T32 = TL, TL = green,
TU = TD, TD = yellow
```

?-

Interaction: Task 2 The Floating Shapes World



1.

2.

```
%------------------
%
% File: shapes_world_1.pro
% Line Loosely represented 2D shapes world.
%




%------------------
%
% Facts...
%
%------------------
```

```
%------------------
%
% square(N,side(L),color(C)) :: N is the name of the square with side L, 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, 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 name of all circles.

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

%------------------
%
% squares :: list the names of all squares.

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

%------------------
%
% shapes :: list the name of all 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 R * R * 3.14.

area(Name,A) :- square(Name,side(L),_), A is L * L.
```

```
3. ?- consult('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.

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

true.

?- shapes.
```

```
carla
cora
connie
claire
sera
sara
sarah
true.

?- blue(Shape).
Shape = sara

?- 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


?- halt.


Process prolog finished
```

Interaction: Task 3 Pokemon KB Interaction and Programming

Interaction: Task 4 Lisp Processing in Prolog

Interaction: Computing the area of the concentric disks image which is blue