Sample Runs:

```
?- consult('D:\\source\\ai_portfolio\\programming-assignment-2\\coloring.pl').
true.

?- coloring(M, [[a,b,c], [[b,c], [a,c], [a,b]]]).
M = [paint(a, red), paint(b, blue), paint(c, yellow)] ;
M = [paint(a, red), paint(b, blue), paint(c, green)] ;
M = [paint(a, red), paint(b, yellow), paint(c, blue)] ;
M = [paint(a, red), paint(b, yellow), paint(c, green)] ;
M = [paint(a, red), paint(b, green), paint(c, blue)] ;
M = [paint(a, red), paint(b, green), paint(c, yellow)] ;
M = [paint(a, blue), paint(b, red), paint(c, yellow)] ;
M = [paint(a, blue), paint(b, red), paint(c, green)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, red)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, green)] ;
M = [paint(a, blue), paint(b, green), paint(c, red)] ;
M = [paint(a, blue), paint(b, green), paint(c, yellow)] ;
M = [paint(a, yellow), paint(b, red), paint(c, blue)] ;
M = [paint(a, yellow), paint(b, red), paint(c, green)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, red)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, green)] ;
M = [paint(a, yellow), paint(b, green), paint(c, red)] ;
M = [paint(a, yellow), paint(b, green), paint(c, blue)] ;
M = [paint(a, green), paint(b, red), paint(c, blue)] ;
M = [paint(a, green), paint(b, red), paint(c, yellow)] ;
M = [paint(a, green), paint(b, blue), paint(c, red)] ;
M = [paint(a, green), paint(b, blue), paint(c, yellow)] ;
M = [paint(a, green), paint(b, yellow), paint(c, red)] ;
M = [paint(a, green), paint(b, yellow), paint(c, blue)] ;
false.

?-
```

```
?- consult('D:\\source\\ai_portfolio\\programming-assignment-2\\coloring.pl').
true.

?- coloring(M,[[a,b,c,d,e],[[b,c,d,e],[a,c,d,e],[a,b,d],[a,b,c],[a,b]]]).
M = [paint(a, red), paint(b, blue), paint(c, yellow), paint(d, green), paint(e, yellow)] ;
M = [paint(a, red), paint(b, blue), paint(c, yellow), paint(d, green), paint(e, green)] ;
M = [paint(a, red), paint(b, blue), paint(c, green), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, red), paint(b, blue), paint(c, green), paint(d, yellow), paint(e, green)] ;
M = [paint(a, red), paint(b, yellow), paint(c, blue), paint(d, green), paint(e, blue)] ;
M = [paint(a, red), paint(b, yellow), paint(c, blue), paint(d, green), paint(e, green)] ;
M = [paint(a, red), paint(b, yellow), paint(c, green), paint(d, blue), paint(e, blue)] ;
M = [paint(a, red), paint(b, yellow), paint(c, green), paint(d, blue), paint(e, green)] ;
M = [paint(a, red), paint(b, green), paint(c, blue), paint(d, yellow), paint(e, blue)] ;
M = [paint(a, red), paint(b, green), paint(c, blue), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, red), paint(b, green), paint(c, yellow), paint(d, blue), paint(e, blue)] ;
M = [paint(a, red), paint(b, green), paint(c, yellow), paint(d, blue), paint(e, yellow)] ;
M = [paint(a, blue), paint(b, red), paint(c, yellow), paint(d, green), paint(e, yellow)] ;
M = [paint(a, blue), paint(b, red), paint(c, yellow), paint(d, green), paint(e, green)] ;
M = [paint(a, blue), paint(b, red), paint(c, green), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, blue), paint(b, red), paint(c, green), paint(d, yellow), paint(e, green)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, red), paint(d, green), paint(e, red)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, red), paint(d, green), paint(e, green)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, green), paint(d, red), paint(e, red)] ;
M = [paint(a, blue), paint(b, yellow), paint(c, green), paint(d, red), paint(e, green)] ;
M = [paint(a, blue), paint(b, green), paint(c, red), paint(d, yellow), paint(e, red)] ;
M = [paint(a, blue), paint(b, green), paint(c, red), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, blue), paint(b, green), paint(c, yellow), paint(d, red), paint(e, red)] ;
M = [paint(a, blue), paint(b, green), paint(c, yellow), paint(d, red), paint(e, yellow)] ;
M = [paint(a, yellow), paint(b, red), paint(c, blue), paint(d, green), paint(e, blue)] ;
M = [paint(a, yellow), paint(b, red), paint(c, blue), paint(d, green), paint(e, green)] ;
M = [paint(a, yellow), paint(b, red), paint(c, green), paint(d, blue), paint(e, blue)] ;
M = [paint(a, yellow), paint(b, red), paint(c, green), paint(d, blue), paint(e, green)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, red), paint(d, green), paint(e, red)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, red), paint(d, green), paint(e, green)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, green), paint(d, red), paint(e, red)] ;
M = [paint(a, yellow), paint(b, blue), paint(c, green), paint(d, red), paint(e, green)] ;
M = [paint(a, yellow), paint(b, green), paint(c, red), paint(d, blue), paint(e, red)] ;
M = [paint(a, yellow), paint(b, green), paint(c, red), paint(d, blue), paint(e, blue)] ;
M = [paint(a, yellow), paint(b, green), paint(c, blue), paint(d, red), paint(e, red)] ;
M = [paint(a, yellow), paint(b, green), paint(c, blue), paint(d, red), paint(e, blue)] ;
M = [paint(a, green), paint(b, red), paint(c, blue), paint(d, yellow), paint(e, blue)] ;
M = [paint(a, green), paint(b, red), paint(c, blue), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, green), paint(b, red), paint(c, yellow), paint(d, blue), paint(e, blue)] ;
M = [paint(a, green), paint(b, red), paint(c, yellow), paint(d, blue), paint(e, yellow)] ;
M = [paint(a, green), paint(b, blue), paint(c, red), paint(d, yellow), paint(e, red)] ;
M = [paint(a, green), paint(b, blue), paint(c, red), paint(d, yellow), paint(e, yellow)] ;
M = [paint(a, green), paint(b, blue), paint(c, yellow), paint(d, red), paint(e, red)] ;
M = [paint(a, green), paint(b, blue), paint(c, yellow), paint(d, red), paint(e, yellow)] ;
M = [paint(a, green), paint(b, yellow), paint(c, red), paint(d, blue), paint(e, red)] ;
M = [paint(a, green), paint(b, yellow), paint(c, red), paint(d, blue), paint(e, blue)] ;
M = [paint(a, green), paint(b, yellow), paint(c, blue), paint(d, red), paint(e, red)] ;
M = [paint(a, green), paint(b, yellow), paint(c, blue), paint(d, red), paint(e, blue)] ;
false.
```

Analysis on Results

I found it very interesting that this constraint satisfaction problem could be solved using the logical programming paradigm by simply specifying the constraints. I had only worked with procedural and object-oriented programming languages before this, so the power of Prolog definitely blew my mind a little bit.