

Programming Assignment- II

Constraint Satisfaction -Graph Coloring Problem

4365 Artificial Intelligence

Find an "admissible" coloring of a graph such that all adjacent nodes have different colors.

(It had been conjectured for a long time that such a coloring is possible with only 4 colors if the graph is planar. This conjecture was formulated precisely by F. Guthrie already in 1852 but proved by K. Appel and W. Haken only in 1976 using a computer program).

Write a Prolog program that finds all admissible colorings of a given graph using the colors red, blue, yellow, and green.

You should write a predicate called `coloring` which has two variables, first one is the assignment and the second one is the graph. The graph is a list with two items, the first item the list of vertices and the second item the list of list with adjacent vertices corresponding to the first list. The assignment will have `paint` for each vertex with appropriate color. The constraint here is that the adjacent vertices cannot have the same color. Your algorithm will return all possible combinations of four colors for the graph.

For example

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

```
M = [paint(a, green), paint(b, yellow)] ;  
false.
```

More examples are given at the end of the description.

Submission Instruction: -

For this assignment you will submit a single .pl file containing your Prolog program. Include a report file with input / output samples of your program.

Hint

Write `color/1` predicate indicating the 4 colors (you may name them red, yellow, green, blue or any 4 different colors).

Write the `different/2` predicate indicating how 2 colors are different.

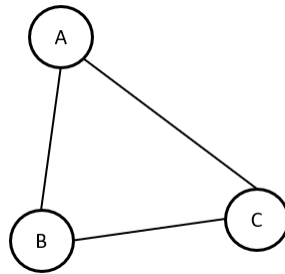
Write `coloring/2` predicate, any coloring must be complete and valid. So you will write 2 different predicate to ensure these 2 properties.

For example the predicate to ensure completeness will just go over the first list and add `paint(X, _)` for each element in the list `[X|T]` and ensure completeness of the remaining items `T` using recursion. There should be a base case stating that if both lists are empty in the graph `[[][]]` then empty assignment is an empty list as well `[]`.

The predicate to ensure validity of the assignment `M` should make sure if `X` is adjacent to `Y` then `X` and `Y` are given different colors in the assignment `M`. To ensure this you may have to look for assignment of `X` and `Y` separately in the assignment list `M` using a different predicate (`assignment(X, M, Xc)` should tell you what color `X` is assigned in the list of assignments `M`)

These assignment should have very few lines of code. If you have any questions, please ask during class.

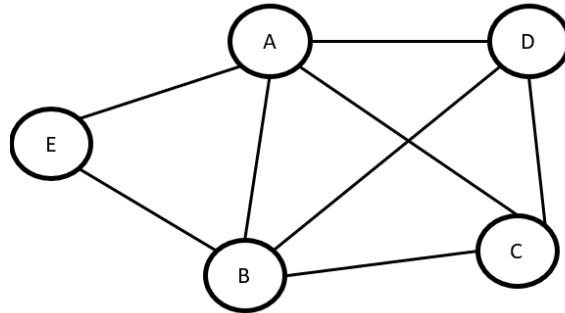
Example-1



```
?- coloring1(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.
```

Example-2

The following graph has multiple four-color assignments.



?- 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)] ;

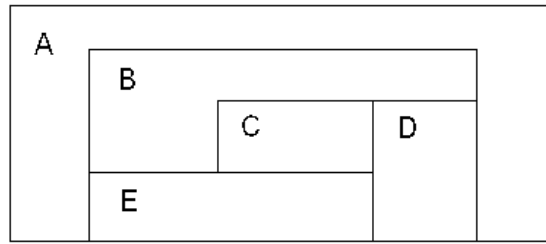
M = [paint(a, blue), paint(b, yellow), paint(c, green), paint(d, red), paint(e, green)] ;

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.

Example-3 Let's say the graphs looks like the following image.



The first list is the list of vertices $[a,b,c,d,e]$, second list is the list of adjacent vertices corresponding to first list $[[b,e,d],[a,c,d,e],[b,d,e],[a,b,c,e],[a,b,c,d]]$.

The above graph is a planar graph and has multiple assignments using 4 colors, some of the solutions are as follows.

```
coloring(M, [[a,b,c,d,e], [[b,e,d], [a,c,d,e], [b,d,e], [a,b,c,e], [a,b,c,d]]).
```

```
M = [paint(a, red), paint(b, blue), paint(c, red), paint(d, green),  
paint(e, yellow)] ;
```

```
M = [paint(a, red), paint(b, blue), paint(c, red), paint(d, yellow),  
paint(e, green)] ;
```

```
M = [paint(a, red), paint(b, yellow), paint(c, red), paint(d, green),  
paint(e, blue)] ;
```

```
M = [paint(a, red), paint(b, yellow), paint(c, red), paint(d, blue),  
paint(e, green)] ;
```

```
M = [paint(a, red), paint(b, green), paint(c, red), paint(d, yellow),  
paint(e, blue)] ;
```

```
M = [paint(a, red), paint(b, green), paint(c, red), paint(d, blue),  
paint(e, yellow)] ;
```

```
M = [paint(a, blue), paint(b, red), paint(c, blue), paint(d, green),  
paint(e, yellow)] ;
```

```
M = [paint(a, blue), paint(b, red), paint(c, blue), paint(d, yellow),  
paint(e, green)] ;
```

```
M = [paint(a, blue), paint(b, yellow), paint(c, blue), paint(d,  
green), paint(e, red)] ;
```

```
M = [paint(a, blue), paint(b, yellow), paint(c, blue), paint(d, red),  
paint(e, green)] ;
```

```
M = [paint(a, blue), paint(b, green), paint(c, blue), paint(d,  
yellow), paint(e, red)] ;
```

```
M = [paint(a, blue), paint(b, green), paint(c, blue), paint(d, red),  
paint(e, yellow)] ;
```

.....

Example-4

```
coloring1(M, [[a,b,c,d,e,f],[[b,c,d,e],[a,c,d,f],[a,b,d,e,f],[a,b,c,e,f],  
[a,c,d],[b,c,d]]]).
```

```
M = [paint(a, red), paint(b, blue), paint(c, yellow), paint(d, green),  
paint(e, blue), paint(f, red)] ;
```

```
M = [paint(a, red), paint(b, blue), paint(c, green), paint(d, yellow),  
paint(e, blue), paint(f, red)] ;
```

```
M = [paint(a, red), paint(b, yellow), paint(c, blue), paint(d, green),  
paint(e, yellow), paint(f, red)] ;
```

```
M = [paint(a, red), paint(b, yellow), paint(c, green), paint(d, blue),  
paint(e, yellow), paint(f, red)] ;
```

```
M = [paint(a, red), paint(b, green), paint(c, blue), paint(d, yellow),  
paint(e, green), paint(f, red)] ;
```

```
M = [paint(a, red), paint(b, green), paint(c, yellow), paint(d, blue),  
paint(e, green), paint(f, red)] ;
```

```
M = [paint(a, blue), paint(b, red), paint(c, yellow), paint(d, green),  
paint(e, red), paint(f, blue)] ;
```

```
M = [paint(a, blue), paint(b, red), paint(c, green), paint(d, yellow),  
paint(e, red), paint(f, blue)] ;
```

```
M = [paint(a, blue), paint(b, yellow), paint(c, red), paint(d, green),  
paint(e, yellow), paint(f, blue)] ;
```

```
M = [paint(a, blue), paint(b, yellow), paint(c, green), paint(d, red),  
paint(e, yellow), paint(f, blue)] ;
```

```
M = [paint(a, blue), paint(b, green), paint(c, red), paint(d, yellow),  
paint(e, green), paint(f, blue)] ;
```

```
M = [paint(a, blue), paint(b, green), paint(c, yellow), paint(d, red),  
paint(e, green), paint(f, blue)] ;
```

```
M = [paint(a, yellow), paint(b, red), paint(c, blue), paint(d, green),  
paint(e, red), paint(f, yellow)] ;
```

```
M = [paint(a, yellow), paint(b, red), paint(c, green), paint(d, blue),  
paint(e, red), paint(f, yellow)] ;
```

```
M = [paint(a, yellow), paint(b, blue), paint(c, red), paint(d, green),  
paint(e, blue), paint(f, yellow)] ;
```

```
M = [paint(a, yellow), paint(b, blue), paint(c, green), paint(d, red),  
paint(e, blue), paint(f, yellow)] ;
```

```
M = [paint(a, yellow), paint(b, green), paint(c, red), paint(d, blue),  
paint(e, green), paint(f, yellow)] ;
```

```
M = [paint(a, yellow), paint(b, green), paint(c, blue), paint(d, red),  
paint(e, green), paint(f, yellow)] ;
```

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