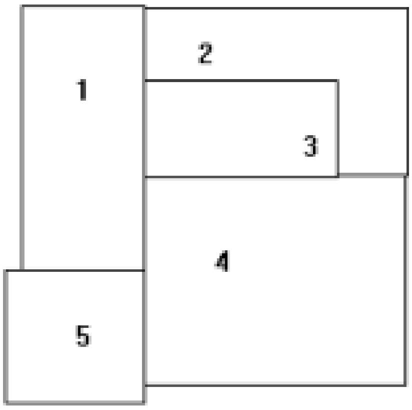
Experiment No. 10

<u>Aim</u>: To implement graph coloring problem in Prolog.

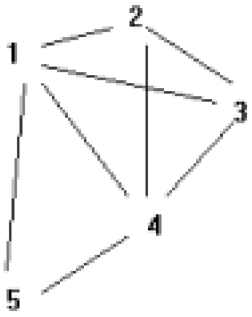
<u>Requirements</u>: Compatible version of SWI-Prolog.

<u>Theory</u>:

A famous problem in mathematics concerns coloring adjacent planar regions. Like cartographic maps, it is required that, whatever colors are actually used, no two adjacent regions may not have the same color. Two regions are considered adjacent provided they share some boundary line segment. Consider the following map.



We have given numerical names to the regions. To represent which regions are adjacent, consider also the following graph.



Here we have erased the original boundaries and have instead drawn an arc between the names of two regions, provided they were adjacent in the original drawing. In fact, the adjacency graph will convey all of the original adjacency information.

Implementation:

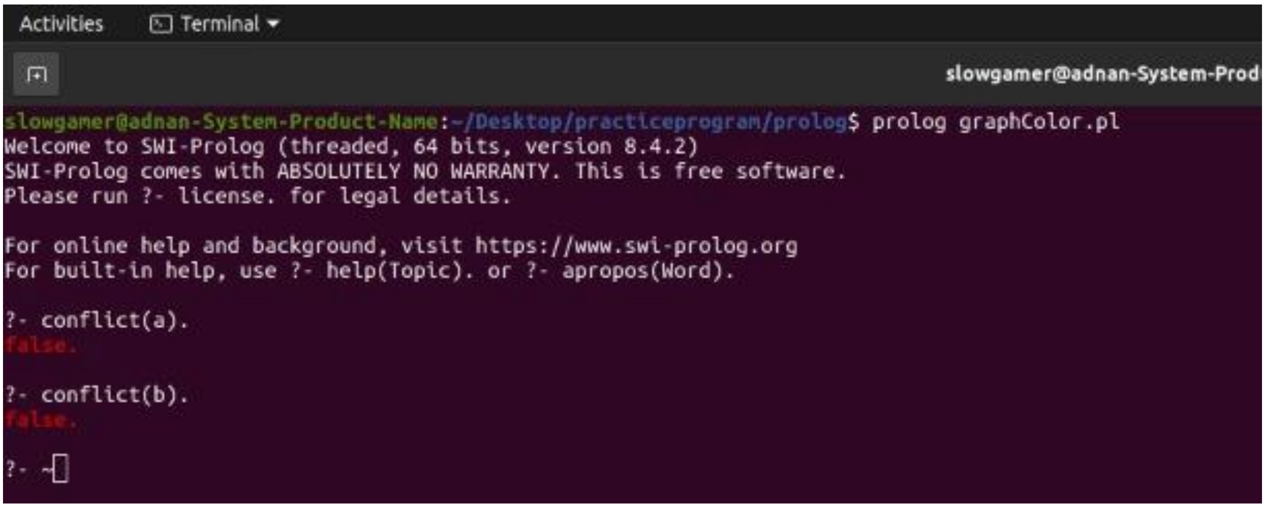
```
color(1,red,a).
color(2,blue,a).
color(3,green,a).
color(4,yellow,a).
color(5,blue,a).
color(1,red,b).
color(2,blue,b).
color(3,green,b).
color(4,yellow,b).
color(5,blue,b).

adj(1,2).
adj(1,3).
adj(1,4).
adj(1,5).
adj(2,3).
adj(2,4).
adj(3,5).
adj(4,5).

adjacent(X,Y) :- adj(X,Y);adj(Y,X).

conflict(Z):- adjacent(X,Y),color(X,K,Z),color(Y,K,Z), write(X),write('->'),write(Y),writeln(' conflict').
```

<u>Output</u>:



```
slowgamer@adnan-System-Product-Name:~/Desktop/practiceprogram/prolog$ prolog graphColor.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- conflict(a).
false.

?- conflict(b).
false.

?- ~
```

<u>Conclusion</u>: We have successfully implemented graph coloring problem in Prolog.