EXPERIMENT 11: Implement a python program for Map Coloring to implement CSP.

Aim: Implement an Algorithm in Python to solve the Map Coloring to implement CSP.

Code:

```
from constraint import Problem


def map_coloring():
    problem = Problem()


    # Define variables (regions) and their possible values (colors)
    problem.addVariables(["WA", "NT", "SA", "Q", "NSW", "V", "T"], ["red", "green", "blue"])


    # Add constraints (neighboring regions must have different colors)
    problem.addConstraint(lambda wa, nt: wa != nt, ("WA", "NT"))
    problem.addConstraint(lambda wa, sa: wa != sa, ("WA", "SA"))
    problem.addConstraint(lambda nt, sa, q: nt != sa and nt != q, ("NT", "SA", "Q"))
    problem.addConstraint(lambda nt, q, nsw: nt != q and nt != nsw, ("NT", "Q", "NSW"))
    problem.addConstraint(lambda sa, q, nsw, v: sa != q and sa != nsw and sa != v, ("SA", "Q", "NSW", "V"))
    problem.addConstraint(lambda q, nsw: q != nsw, ("Q", "NSW"))
    problem.addConstraint(lambda nsw, v: nsw != v, ("NSW", "V"))


    solutions = problem.getSolutions()
    return solutions


if __name__ == "__main__":
    solutions = map_coloring()
    print("Number of solutions:", len(solutions))
    for solution in solutions:
        print(solution)
```

Output:

```
Number of solutions: 4
{'WA': 'red', 'NT': 'green', 'SA': 'blue', 'Q': 'red', 'NSW': 'green', 'V': 'red', 'T': 'red'}
{'WA': 'red', 'NT': 'green', 'SA': 'blue', 'Q': 'red', 'NSW': 'green', 'V': 'red', 'T': 'green'}
{'WA': 'red', 'NT': 'green', 'SA': 'blue', 'Q': 'red', 'NSW': 'green', 'V': 'blue', 'T': 'red'}
{'WA': 'red', 'NT': 'green', 'SA': 'blue', 'Q': 'red', 'NSW': 'green', 'V': 'blue', 'T': 'green'}
```

Result: Code has been Implemented successfully.