Since we have a 9x9 square board, we can divide it into 9 3x3 sub-boards.Initially all the squares are colored blue.
For part a) we want all sub-squares to be of the same color but neighbouring (top , down, left, right) sub-squares to be of different color than the current sub-square. There are only 2 possible solutions to the same:-
make the corner sub-squares and the centre sub-square red and the others blue
make the corner sub-squares and the centre sub-square blue and the others red

R	R	R	B	B	B	R	R	R
R	RED	R	B	BLUE	B	R	RED	R
R	R	R	B	B	B	R	R	R
B	B	B	R	R	R	B	B	B
B	BLUE	B	R	RED	R	B	BLUE	B
B	B	B	R	R	R	B	B	B
R	R	R	B	B	B	R	R	R
R	RED	R	B	BLUE	B	R	RED	R
R	R	R	B	B	B	R	R	R

making it look more compact,w e get (here each cell is a 3x3 sub-square)

RED	BLUE	RED
BLUE	RED	BLUE
RED	BLUE	RED

The solution space is of size 2, **however since any square can be any colour, the state space rises to 281 (as each of the 81 squares has 2 choices of color, either red or blue, hence 2x2x2x2....81 times)**
formulating the problem as follows:-
Find a solution to the grid coloring problem of a grid of size 9x9 where each sub-grid of size 3x3 is of different color to its neighbours. Initially the grid is blue and only 2 colors are allowed per square - red or blue.
For part b) the state space does not change as we still have the choice to color each square with both color, albeit we can color only once.
Hence this condition makes the breadth first solution much faster at converging at the right solution than the iterative depth first search as the depth first search relies on backtracking, that is, change your combination if it is not the right one. However, since here we color each square only once, hence we will not be able to backtrack (as in backtrack, when we get stuck goin down the current path, we backtrack to previous most correct partial solution, tweak, and start again).
reformulating the problem as follows:-
Find a solution to the grid coloring problem of a grid of size 9x9 where each sub-grid of size 3x3 is of different color to its neighbours. Initially the grid is blue and only 2 colors are allowed per square - red or blue. Each square may have its color changed at max once.
For part c), we are not dealing with colors at the square level but at the 3x3 sub-square level. Hence this gives us a state space of only **29 = 512 valid states**
reformulating the problem as follows:-
Find a solution to the grid coloring problem of a grid of size 3x3where each square is of

different color to its neighbours. Initially the grid is blue and only 2 colors are allowed per square - red or blue. Each square may have its color changed at max once.

To de-abstract/generalize part a) from part c), we proceed as follows:-

->color a 3x3 grid with either red or blue such that no two adjacent neighbours have the same color

->break each square into a further 3x3 grid

->each square can have its color changed any number of times