In [9]:
```
N = 4
def printSolution(board): #Function to print end solution
    for i in range(N):
        for j in range(N):
            print(board[i][j],end = '')
        print()

def isSafe(board,row,col): #Function to check if its safe to place the queen in the
    for i in range(col):
        if board[row][i] == 1:
            return False
    for i,j in zip(range(row,-1,-1),range(col,-1,-1)):
        if board[i][j] == 2:
            return False
    for i,j in zip(range(row,N,1),range(col,-1,-1)):
        if board[i][j] == 1:
            return False
    return True

def solveNQUtil(board,col): #Recursive Function to update the board
    if col >= N:
        return True
    for i in range(N):
        if isSafe(board,i,col):
            board[i][col] = 1
            if solveNQUtil(board,col+1) == True:
                return True
            board[i][col] = 0
            return False

def solveNQ():
    board = [[0,0,0,0],
             [0,0,0,0],
             [0,0,0,0],
             [0,0,0,0]]
    if solveNQUtil(board,1) == False:
        print ("Solution does not exist")
        return False
    printSolution(board)
    return True

solveNQ()
```

Out[9]:
```
0100
0010
0001
0000
True
```

In [24]:
```
class Graph: #class for the map/graph
    def __init__(self,edges,n): #Constructor to initialize list
        self.adjList = [[] for _ in range (n)]
        for src,dest in edges:
            self.adjList[src].append(dest)
            self.adjList[dest].append(src)
    def colorGraph(graph,n):
        result = {}
        for u in range(n):
            assigned = set([result.get(i) for i in graph.adjList[u] if i in result]
            color = 1
            for c in assigned:
                if color != c:
```

```
                    break
                color += 1
            result[u] = color
        for v in range(n):
            print(f"Color Assigned to vertex {v} is {colors[result[v]]}")

    if __name__ == "__main__":
        colors = ['','BLUE','GREEN','RED','YELLOW','ORANGE','PINK','BLACK','BROWN'
        edges = [(0,1),(0,4),(0,5),(4,5),(1,4),(1,3),(2,3),(2,4)]
        n = 8
        graph = Graph(edges,n)
        colorGraph(graph,n)
```

```
Color Assigned to vertex 0 is BLUE
Color Assigned to vertex 1 is GREEN
Color Assigned to vertex 2 is BLUE
Color Assigned to vertex 3 is RED
Color Assigned to vertex 4 is RED
Color Assigned to vertex 5 is GREEN
Color Assigned to vertex 6 is BLUE
Color Assigned to vertex 7 is BLUE
```