4 queen,
8 queen

Apurva Gupta

September 2022

1 4 QUEEN

```
global N
N = 4

def printSolution(board):
    for i in range(N):
        for j in range(N):
            print (board[i][j],end=' ')
        print()



def isSafe(board, row, col):


    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] == 1:
            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):
```

```
    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, 0) == False:
        print ("Solution does not exist")
        return False

    printSolution(board)
    return True


solveNQ()
```

1.1 OUTPUT

```
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

```
    True
```

2 8 QUEEN

```
    print ("Enter the number of queens")
N = int(input())

board = [[0]*N for _ in range(N)]
def attack(i, j):

    for k in range(0,N):
        if board[i][k]==1 or board[k][j]==1:
            return True

    for k in range(0,N):
        for l in range(0,N):
            if (k+l==i+j) or (k-l==i-j):
                if board[k][l]==1:
                    return True
    return False
def N_queens(n):
    if n==0:
        return True
    for i in range(0,N):
        for j in range(0,N):
            if (not(attack(i,j))) and (board[i][j]!=1):
                board[i][j] = 1
                if N_queens(n-1)==True:
                    return True
                board[i][j] = 0
    return False
N_queens(N)
for i in board:
    print (i)
```

2.1 OUTPUT

```
Enter the number of queens
8
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
```

```
[0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0, 0]
```