

N_QUEEN PROBLEM

NAME: J.L.PUGAZH MUKILAN

REG NO: 22BCE9292

SLOT :L7+L8

1)Solve N queen problem

```
Click here to ask Blackbox to help you code faster
import numpy as np
import sys
global N
N=10

def issafe(matrix,row,col):
    # for the left diagonal
    for i,j in zip(range(row, -1,-1), range(col, -1,-1)):
        if matrix[i][j]==1:
            return False

    #for the right diagonal
    for i, j in zip(range(row, N,1), range(col, N,1)):
        if matrix[i][j]==1 :
            return False

    #checking in the particular row
    for i in range(col):
        if matrix[row][i]==1:
            return False
    return True

def solveNqueen(matrix,column_to_solve):
    if column_to_solve>=N:
        return True

    for i in range(N):
        if issafe(matrix,i,column_to_solve):
            matrix[i][column_to_solve]=1

            if solveNqueen(matrix,column_to_solve+1)==True:
                print("wait for some time")
                return True

            matrix[i][column_to_solve]=0

    return False
```

```

def printmatrix(matrix):
    for i in range(N):
        print("\n")
        for j in range(N):
            if matrix[i][j]==1:
                print(" Q ",end=" ")
            else:
                print(" . ",end=" ")

def start():
    print("started")
    matrix=np.zeros((N,N))
    if solveNqueen(matrix,0):
        printmatrix(matrix)
        return True
    print(matrix)
start()

```

OUTPUT:

10 QUEEN

20 QUEEN

The image displays two 10x10 matrices, each representing a solution to the 10-Queens problem. In these matrices, 'Q' denotes the position of a queen, and '.' denotes an empty cell. The left matrix shows a valid solution where no two queens share the same row, column, or diagonal. The right matrix shows another valid solution, also with no two queens sharing the same row, column, or diagonal.