EXP 2: STATE SPACE SEARCH ALGORITHMS

2. 2 N QUEENS PROBLEM

```
1. Start
2.  Taking number of queens as input from user and create a chessboard. The board is initialized with zeros, representing empty cells.
        NxN matrix with all elements set to 0
3. Checking vertically and horizontally, this function checks if placing a queen at (i, j) would be attacked by any existing queens. It checks rows, columns, and diagonals.
4. Define k ranging from 0to N
        check if board[i][k]==1 or board[k][j]==1:
        Then return True
5. checking diagonally
        Define k ranging from 0to N
        Define I ranging from 0to N
        Check if  (k+l==i+j) or (k-l==i-j):
        Check if board[k][l]==1:
        Then return True
        Else return False
6. Placing Queens on board, the recursive function tries to place queens one by one in different cells, ensuring no two queens attack each other. It uses backtracking to find a solution.
        Define I ranging from 0to N
        Define J ranging from 0to N
        Check if (not(attack(i,j))) and (board[i][j]!=1):
        board[i][j] = 1
        if N_queens(n-1)==True:
        Then return True
        board[i][j] = 0
        Then return False
7. Display Queens on Board
8. Stop
```

PROGRAM

```
# Taking number of queens as input from user

print ("Enter the number of queens")

N = int(input())

# here we create a chessboard

# NxN matrix with all elements set to 0

board = [[0]*N for _ in range(N)]
```

```
def attack(i, j):
    #checking vertically and horizontally
    for k in range(0,N):
        if board[i][k]==1 or board[k][j]==1:
            return True
    #checking diagonally
    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)
```

RESULT

The above program has been successfully executed and output obtained is verified.

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]
```