Ex·No:1 N-QUEENS PROBLEM

DATE:

AIM:

To solve the N-Queen Problem where the goal is to place n-queens on a nxn chessboard such that no 2 queens attack each other.

ALGORITHM:

step 1: start
step 2: create a nxn chessboard with all cells set to 0, representing no queens placed.
Step 3: Ensure no queen in same row / upper diagonal / lower diagonal.
step 4: Try placing a queen in each row of current column if it is safe using safe().
step 5: Move to next column if works, else backtrack by removing queen.
step 6: If queens are placed in all columns, return success
Step 7: Display the board.
Step 8: Print 'Solution doesn't exist' if no solution exists.

3. No Dir

PROGRAM:

```
def isSafe(board, row, col, n):
    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,1,-1),
                   range(col,-1,-1))
        if board[i][j]==1:
            return false
    return true
def solveNQutil(board, col, n):
    if col>=n:
        retun true
    for i in range(n):
        if isSafe(board, i, col, n):
            board[i][col]==1
        if solveNQutil(board, col+1, n)==true:
            return True
        board[i][col]=0
    return False
def solveNQ(n):
    board=[[0]*n for _ in range(n)]
    if solveNQutil(board,0,n)==False:
        print("Solution doesn't exist")
        return False
    for i in board:
        print(i)
        return True;
    n = int(input("entern n value:")
    solve_NQ(n)
```

OUTPUT :

```
Enter n value:-5
    [1,0,0,0,0]
    [0,0,0,1,0]
    [0,1,0,0,0]
    [0,0,0,0,1]
    [0,0,1,0,0]
```