EXP NO: 1
DATE :

N-QUEENS PROBLEM

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

ALGORITHM:

```
Step1: Start
Step 2: Create a n×n chessboard with all cells set to 0,
representing no queens placed.
Step3 : Ensure no queen is in the same row,
upper diagonal, or lower for a given pos
Step4: Try placing a queen in each row of the
current col if it is safe using
Step5: Move to the next col if placing a queen
works, else backtrack by removing queen.
Step 6: If queen are placed in all columns return
success.
Step7: Display the board
Step8: If no sol. exists, print "solution does not
exist".
```

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  solve NQUtil (board , col, n):
    if  col >=n:
        return 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 does not exist")
        return False
    for i in board :
        print (i)
    return True
n = int (input ("enter n value :"))
solve NQ (n)
```

Output:-

```
Enter 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]
```