Exp No: 01

Date: 10/08/24

N - Queens

Aim:

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

Algorithm:

1) Start
2) Create a nxn chessboard with all cells set to 0, representing no games queens placed.
3) Ensure no queen is in the same row, upper diagonal, or lower diagonal for a given position.
4) Try placing a queen in each row of the current column, if it is safe using issafe ().
5) Move to the next column if placing a queen works, else backtrack by removing queen.
6) If queen are placed in all columns return success.
7) Display the board.
8) If no solution 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,n,-1), range (col,-1,-1)):
        if board [i][j]==1 :
            return False

   return True

def solveNQUtil (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 p in board:
        print(p)
    return True

n = int(input("Enter n value:"))
solveNQ(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]
```

Result:

Thus, the program of N-Queens problem was successfully executed and the output was verified.