**NAME :** DUDHE ANIL AJINATH

**ROLL NO :** TEAD21269

**SUBJECT :** AI

**CLASS :** TE

**BRANCH :** AI&DS

**EXPERIMENT NO :**

**TITLE :**

**Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph colouring problem.**

**CODE:**

def reset():

global board, counter

board = [[0] \* n for i in range(n)]

counter = 0

def display(board):

global counter

counter += 1

for i in range(0, n):

for j in range(0, n):

print(board[i][j], end=" ")

print()

print()

def check(board, row, column):

for i in range(0, column):

if board[row][i] == 1:

return False

for i, j in zip(range(row, -1, -1), range(column, -1, -1)):

if board[i][j] == 1:

return False

for i, j in zip(range(row, n), range(column, -1, -1)):

if board[i][j] == 1:

return False

return True

def marker(board, column):

possibility = False

if column == n:

display(board)

return True

for i in range(0, n):

if check(board, i, column):

board[i][column] = 1

possibility = marker(board, column + 1)

board[i][column] = 0

return possibility

n = int(input("Enter the size of the board: "))

counter = 0

reset()

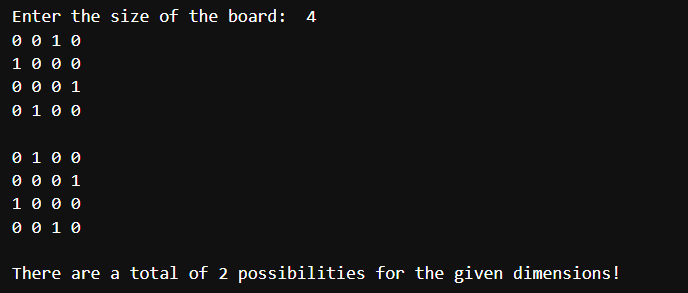
if not marker(board, 0) and counter == 0:

print("No feasible solution exists for the given dimensions")

else:

print("There are a total of " + str(counter) + " possibilities for the given dimensions!")

**OUTPUT:**

****