# Python program to implement 8 queen problem

# Function to check if it is safe to place

# the queen at board[row][col]

def isSafe(mat, row, col):

n = len(mat)

# Check this col on upper side

for i in range(row):

if mat[i][col]:

return False

# Check upper diagonal on left side

i, j = row - 1, col - 1

while i >= 0 and j >= 0:

if mat[i][j]:

return False

i -= 1

j -= 1

# Check lower diagonal on left side

i, j = row - 1, col + 1

while i >= 0 and j < n:

if mat[i][j]:

return False

i -= 1

j += 1

return True

def placeQueens(row, mat):

n = len(mat)

# base case: If all queens are placed

# then return true

if row == n:

return True

# Consider the row and try placing

# queen in all columns one by one

for i in range(n):

# Check if the queen can be placed

if isSafe(mat, row, i):

mat[row][i] = 1

if placeQueens(row + 1, mat):

return True

mat[row][i] = 0

return False

# Function to find the solution

# to the 8-Queens problem

def queens():

n=int(input("how many queens are there ?"))

# Initialize the board

mat = [[0] \* n for \_ in range(n)]

placeQueens(0, mat)

return mat

if \_\_name\_\_ == "\_\_main\_\_":

res = queens()

for v in res:

print(" ".join(map(str, v)))