# -\*- coding: utf-8 -\*-

import copy;

\_\_author\_\_ = 'jinyao.xian';

total = 0;

class Position:

def \_\_init\_\_(self, x, y):

self.x = x;

self.y = y;

def eight\_queen(num):

blank\_chess\_board = [[0 for \_ in range(num)] for \_ in range(num)];

def queen(chess\_board, nextx, chess\_num=0):

if chess\_num == num: # 可使用nextx作为递归最终条件 nextx == num

print\_board(chess\_board);

else:

for y in range(len(chess\_board[nextx])): # 纵向遍历

if can\_use\_this\_position(Position(nextx, y), chess\_board):

new\_chess\_board = copy.deepcopy(chess\_board);

put\_chess(Position(nextx, y), new\_chess\_board);

queen(new\_chess\_board, nextx + 1, chess\_num+1); # 横向遍历

queen(blank\_chess\_board, 0);

def print\_board(chess\_board):

global total;

total += 1;

for x in chess\_board:

print(x);

print("")

def can\_use\_this\_position(position, chess\_board):

if 1 in chess\_board[position.x]:

return False;

for x in range(len(chess\_board)):

if 1 == chess\_board[x][position.y]:

return False;

tmp = abs(x - position.x);

if position.y - tmp >= 0 and chess\_board[x][position.y - tmp] == 1:

return False;

elif position.y + tmp < len(chess\_board[x]) and chess\_board[x][position.y + tmp] == 1:

return False;

return True;

def put\_chess(position, chess\_board):

chess\_board[position.x][position.y] = 1;

return chess\_board;

if \_\_name\_\_ == '\_\_main\_\_':

eight\_queen(8);

print(total);