**✅ Python Program: 8-Queen Problem Using Backtracking**

**PROGRAM:**

N = 8 # Number of queens

def print\_solution(board):

print("🟰 Solution:")

for row in board:

line = ""

for col in row:

line += "👑 " if col else "⬜ "

print(line)

print("\n")

def is\_safe(board, row, col):

# Check same column

for i in range(row):

if board[i][col]:

return False

# Check upper-left diagonal

i, j = row - 1, col - 1

while i >= 0 and j >= 0:

if board[i][j]:

return False

i -= 1

j -= 1

# Check upper-right diagonal

i, j = row - 1, col + 1

while i >= 0 and j < N:

if board[i][j]:

return False

i -= 1

j += 1

return True

def solve\_n\_queen(board, row):

if row == N:

print\_solution(board)

return True # Change to return False if you want \*\*all\*\* solutions

for col in range(N):

if is\_safe(board, row, col):

board[row][col] = 1

if solve\_n\_queen(board, row + 1):

return True

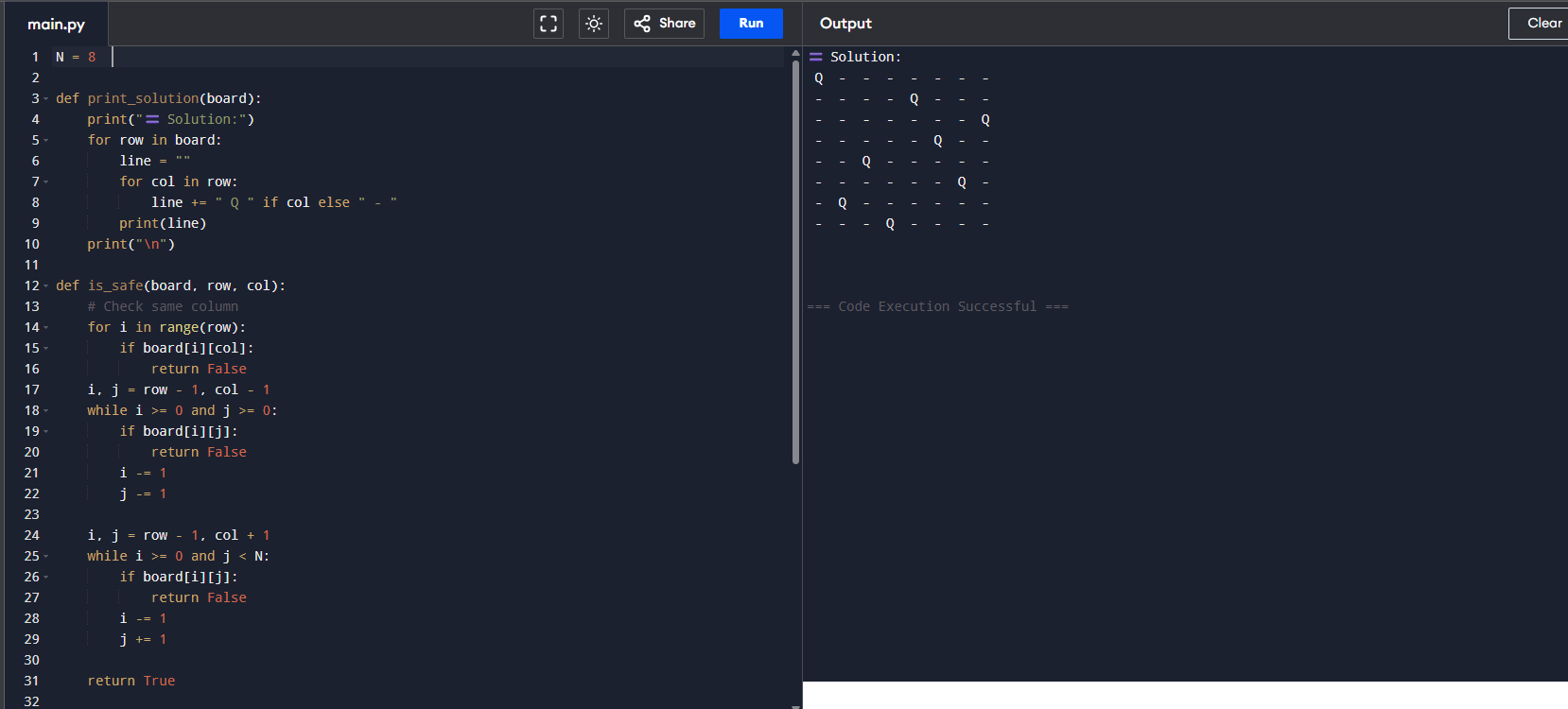
board[row][col] = 0 # Backtrack

return False

board = [[0 for \_ in range(N)] for \_ in range(N)]

if not solve\_n\_queen(board, 0):

print("❌ No solution found.")

****