Artificial Intelligence Lab Assignment 4

def displayBoard(board):

for i in range(len(board)):

for j in range(len(board)):

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

print("")

def isSafe(possible, j, k, n):

# Checking if entire column (k) is safe

for r in range(0, j):

if possible[r][k] == 'Q': # Changing Row

return False

# Checking if left diagonal is safe (left to right)

sub = min(j, k)

left\_j = j - sub

left\_k = k - sub

while left\_j < j and left\_k < k:

if possible[left\_j][left\_k] == 'Q':

return False

else:

left\_j += 1

left\_k += 1

# Checking if right diagonal is safe (right to left) (chk only in one direction)

right\_j = j

right\_k = k

while right\_j >= 0 and right\_k < n:

if possible[right\_j][right\_k] == 'Q':

return False

else:

right\_j -= 1

right\_k += 1

return True

def solve\_n\_queens\_util(board, row, n):

if row == n:

return True

for i in range(n):

if isSafe(board, row, i, n):

board[row][i] = 'Q'

if solve\_n\_queens\_util(board, row + 1, n):

return True

board[row][i] = '\_'

return False

def solve\_n\_queens(n):

board = [['\_'] \* n for \_ in range(n)]

if not solve\_n\_queens\_util(board, 0, n):

print("NO SOLUTION FOUND !!!")

else:

print("\n\n POSSIBLE SOLUTION IS ")

displayBoard(board)

n = int(input("ENTER SIZE OF CHESS BOARD = "))

solve\_n\_queens(n)

PS C:\Users\hp\Desktop\AI> python assi4.py

ENTER SIZE OF CHESS BOARD = 4

POSSIBLE SOLUTION IS

\_ Q \_ \_

\_ \_ \_ Q

Q \_ \_ \_

\_ \_ Q \_

PS C:\Users\hp\Desktop\AI> python assi4.py

ENTER SIZE OF CHESS BOARD = 8

POSSIBLE SOLUTION IS

Q \_ \_ \_ \_ \_ \_ \_

\_ \_ \_ \_ Q \_ \_ \_

\_ \_ \_ \_ \_ \_ \_ Q

\_ \_ \_ \_ \_ Q \_ \_

\_ \_ Q \_ \_ \_ \_ \_

\_ \_ \_ \_ \_ \_ Q \_

\_ Q \_ \_ \_ \_ \_ \_

\_ \_ \_ Q \_ \_ \_ \_

PS C:\Users\hp\Desktop\AI>