|  |  |
| --- | --- |
| EX.NO:1 | **8 QUEENS PROBLEM** |
| DATE: |

**Program:**

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

for i in range(row):

if board[i][col]=='Q':

return False

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

if board[i][j]=='Q':

return False

for i,j in zip(range(row,-1,-1),range(col,len(board))):

if board[i][j]=='Q':

return False

return True

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

n=len(board)

if row==n:

return True

for col in range(n):

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

board[row][col]='Q'

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

turn Tr

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

return False

def solve\_queen():

board=[['\_'for \_ in range(8)] for j in range(8)]

if solve\_queen\_util(board,0)==False:

printf("Solution doesn't Exist")

return False

print\_board(board)

return Trdef print\_board(board):

for i in range(8):

for j in range(8):

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

print()

solve\_queen()

**Output:**



**Result:**

The Program 8 Queens program is successfully compiled and implemented.