global N

N=int(input("Enter number of queens :"))

Board=[['\_'for x in range(N)] for y in range(N)]

print(Board)

def printBoard(Board):

for i in Board:

for j in i:

print(j,end='')

print(" ")

def isSafe(Board,row,col):

for i in range(col):

if(Board[row][i]=='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,N,1),range(col,-1,-1)):

if(Board[i][j]=='Q'):

return False

return True

def SolveQueen(Board,col):

if(col>=N):

return True

for i in range(N):

if(isSafe(Board,i,col)==True):

Board[i][col]='Q'

printBoard(Board)

print(' ')

if(SolveQueen(Board,col+1)==True):

return True

Board[i][col]='\_'

print('\nBacktraking here..')

return False

if SolveQueen(Board,0)==False:

print('\nSolution not exist')

else:

print('\nFinal solution')

printBoard(Board)

output

Enter number of queens :4

[['\_', '\_', '\_', '\_'], ['\_', '\_', '\_', '\_'], ['\_', '\_', '\_', '\_'], ['\_', '\_', '\_', '\_']]

Q\_\_\_

\_\_\_\_

\_\_\_\_

\_\_\_\_

Q\_\_\_

\_\_\_\_

\_Q\_\_

\_\_\_\_

Backtraking here..

Q\_\_\_

\_\_\_\_

\_\_\_\_

\_Q\_\_

Q\_\_\_

\_\_Q\_

\_\_\_\_

\_Q\_\_

Backtraking here..

Backtraking here..

Backtraking here..

\_\_\_\_

Q\_\_\_

\_\_\_\_

\_\_\_\_

\_\_\_\_

Q\_\_\_

\_\_\_\_

\_Q\_\_

\_\_Q\_

Q\_\_\_

\_\_\_\_

\_Q\_\_

\_\_Q\_

Q\_\_\_

\_\_\_Q

\_Q\_\_

Final solution

\_\_Q\_

Q\_\_\_

\_\_\_Q

\_Q\_\_