# Number of queens

n=4

# Matrix

a=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]

# Dictionary for backtrack

b={}

# Checking if column is safe

def isColumnSafe(r,c):

while(r>=0):

if(a[r][c] == 1):

return 0

r = r-1

return 1

# Checking if left diagonal is safe

def isLeftDiagonalSafe(r,c):

while(r>=0 and c>=0):

if(a[r][c] == 1):

return 0

r = r-1

c = c-1

return 1

# Checking if right diagonal is safe

def isRightDiagonalSafe(r,c):

while(r>=0 and c<n):

if(a[r][c]==1):

return 0

r = r-1

c = c+1

return 1

def isSafe(r,c):

if(isColumnSafe(r,c) and isLeftDiagonalSafe(r,c) and isRightDiagonalSafe(r,c)):

return True

return False

def chessboard(r,c):

if(r>=n):

return

p = 0

while c<n:

p = isSafe(r,c)

if p == 1:

a[r][c] = 1

b.update({r:c})

break

c=c+1

if p==1:

chessboard(r+1,0)

else:

a[r-1][b.get(r-1)]=0

chessboard(r-1,int(b.get(r-1))+1)

chessboard(0,0)

print("Matrix is:- ",a)

print("Dictio

**OUTPUT**

Matrix is:- [[0, 1, 0, 0], [0, 0, 0, 1], [1, 0, 0, 0], [0, 0, 1, 0]]

Dictionary is:- {0: 1, 1: 3, 2: 0, 3: 2}

=== Code Execution Successful ===

nary is:- ",b)