**Tic-tac-toe**

import mathimport os

import random

import re

import sys

def nextMove(player, board):

Empty = '\_'

tmarker = 'O' if player == 'X' else 'X'

if emptyBoard(board):

if board[1][1] == Empty:

print("1 1")

return

else:

if searchBoard(player, board):

return

elif searchBoard(tmarker, board):

return

else:

if board[1][1] == Empty:

print("1 1")

return

elif board[0][0] == Empty:

print("0 0")

return

elif board[0][2] == Empty:

print("0 2")

return

elif board[2][0] == Empty:

print("2 0")

return

elif board[2][2] == Empty:

print("2 2")

return

def emptyBoard(board):

for row in board:

for cell in row:

if cell != '\_':

return False

return True

def searchBoard(marker, board):

Empty = '\_'

def check\_and\_print(r, c):

print(f"{r} {c}")

return True

if board[1][1] == marker:

if board[0][1] == marker and board[2][1] == Empty:

return check\_and\_print(2, 1)

elif board[1][0] == marker and board[1][2] == Empty:

return check\_and\_print(1, 2)

elif board[1][2] == marker and board[1][0] == Empty:

return check\_and\_print(1, 0)

elif board[2][1] == marker and board[0][1] == Empty:

return check\_and\_print(0, 1)

if board[0][0] == marker:

if board[0][1] == marker and board[0][2] == Empty:

return check\_and\_print(0, 2)

elif board[1][0] == marker and board[2][0] == Empty:

return check\_and\_print(2, 0)

elif board[1][1] == marker and board[2][2] == Empty:

return check\_and\_print(2, 2)

elif board[2][0] == marker and board[1][0] == Empty:

return check\_and\_print(1, 0)

elif board[0][2] == marker and board[0][1] == Empty:

return check\_and\_print(0, 1)

elif board[2][2] == marker and board[1][1] == Empty:

return check\_and\_print(1, 1)

if board[0][2] == marker:

if board[0][1] == marker and board[0][0] == Empty:

return check\_and\_print(0, 0)

elif board[1][2] == marker and board[2][2] == Empty:

return check\_and\_print(2, 2)

elif board[1][1] == marker and board[2][0] == Empty:

return check\_and\_print(2, 0)

elif board[2][2] == marker and board[1][2] == Empty:

return check\_and\_print(1, 2)

elif board[0][0] == marker and board[0][1] == Empty:

return check\_and\_print(0, 1)

elif board[2][0] == marker and board[1][1] == Empty:

return check\_and\_print(1, 1)

if board[2][0] == marker:

if board[2][1] == marker and board[2][2] == Empty:

return check\_and\_print(2, 2)

elif board[1][0] == marker and board[0][0] == Empty:

return check\_and\_print(0, 0)

elif board[1][1] == marker and board[0][2] == Empty:

return check\_and\_print(0, 2)

elif board[0][0] == marker and board[1][0] == Empty:

return check\_and\_print(1, 0)

elif board[2][2] == marker and board[2][1] == Empty:

return check\_and\_print(2, 1)

elif board[0][2] == marker and board[1][1] == Empty:

return check\_and\_print(1, 1)

if board[2][2] == marker:

if board[1][1] == marker and board[0][0] == Empty:

return check\_and\_print(0, 0)

elif board[2][1] == marker and board[2][0] == Empty:

return check\_and\_print(2, 0)

elif board[1][2] == marker and board[0][2] == Empty:

return check\_and\_print(0, 2)

elif board[0][2] == marker and board[1][2] == Empty:

return check\_and\_print(1, 2)

elif board[2][0] == marker and board[2][1] == Empty:

return check\_and\_print(2, 1)

elif board[0][0] == marker and board[1][1] == Empty:

return check\_and\_print(1, 1)

return False

if \_\_name\_\_ == '\_\_main\_\_':

player = input().strip()

board = [input().strip() for \_ in range(3)]

nextMove(player, board)

**Output**

