**Exp :1 Write the Python Program to solve 8-Puzzle Problem.**

**Input:**

from collections import deque

goal\_state = [[1, 2, 3],

              [4, 5, 6],

              [7, 8, 0]]

moves = [(-1, 0), (1, 0), (0, -1), (0, 1)]

def is\_valid(x, y):

    return 0 <= x < 3 and 0 <= y < 3

def get\_neighbors(state):

    neighbors = []

    x, y = [(ix, iy) for ix, row in enumerate(state)

            for iy, i in enumerate(row) if i == 0][0]

    for dx, dy in moves:

        nx, ny = x + dx, y + dy

        if is\_valid(nx, ny):

            new\_state = [row[:] for row in state]

            new\_state[x][y], new\_state[nx][ny] = new\_state[nx][ny], new\_state[x][y]

            neighbors.append(new\_state)

    return neighbors

def bfs(start):

    queue = deque()

    visited = set()

      queue.append((start, []))

    visited.add(str(start))

        while queue:

        state, path = queue.popleft()

        if state == goal\_state:

            return path + [state]

        for neighbor in get\_neighbors(state):

            s = str(neighbor)

            if s not in visited:

                visited.add(s)

                queue.append((neighbor, path + [state]))

    return None

start\_state = [[1, 2, 3],

               [4, 0, 6],

               [7, 5, 8]]

solution = bfs(start\_state)

if solution:

    print("Steps to solve:")

    for step in solution:

        for row in step:

            print(row)

        print("------")

else:

    print("No solution found.")

**output:**

