**Exp: 9 Write the Python to Implement Travelling Salesman Problem.**

**Input:**

from itertools import permutations

def calculate\_distance(path, distance\_matrix):

    total\_distance = 0

    for i in range(len(path) - 1):

        total\_distance += distance\_matrix[path[i]][path[i+1]]

    total\_distance += distance\_matrix[path[-1]][path[0]]

    return total\_distance

def travelling\_salesman(distance\_matrix):

    n = len(distance\_matrix)

    cities = list(range(n))

    min\_distance = float('inf')

    best\_path = None

    for perm in permutations(cities[1:]):

        path = [0] + list(perm)

        current\_distance = calculate\_distance(path, distance\_matrix)

        if current\_distance < min\_distance:

            min\_distance = current\_distance

            best\_path = path

    return best\_path, min\_distance

distance\_matrix = [

    [0, 10, 15, 20],

    [10, 0, 35, 25],

    [15, 35, 0, 30],

    [20, 25, 30, 0]

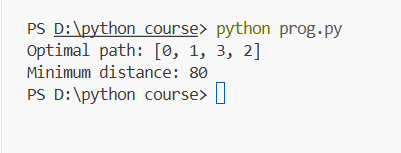
]

path, distance = travelling\_salesman(distance\_matrix)

print("Optimal path:", path)

print("Minimum distance:", distance)

**output:**

****