**EXPERIMENT 9:**

import itertools

distance\_matrix = [

[0, 10, 15, 20],

[10, 0, 35, 25],

[15, 35, 0, 30],

[20, 25, 30, 0]

]

def calculate\_total\_distance(route, matrix):

total\_distance = 0

for i in range(len(route) - 1):

total\_distance += matrix[route[i]][route[i+1]]

total\_distance += matrix[route[-1]][route[0]]

return total\_distance

def tsp\_brute\_force(matrix):

n = len(matrix)

cities = list(range(n))

min\_distance = float('inf')

best\_route = None

for perm in itertools.permutations(cities):

current\_distance = calculate\_total\_distance(perm, matrix)

if current\_distance < min\_distance:

min\_distance = current\_distance

best\_route = perm

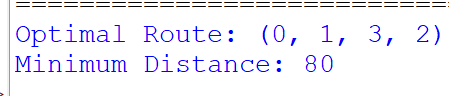
return best\_route, min\_distance

route, distance = tsp\_brute\_force(distance\_matrix)

print("Optimal Route:", route)

print("Minimum Distance:", distance)

**OUTPUT:**

****