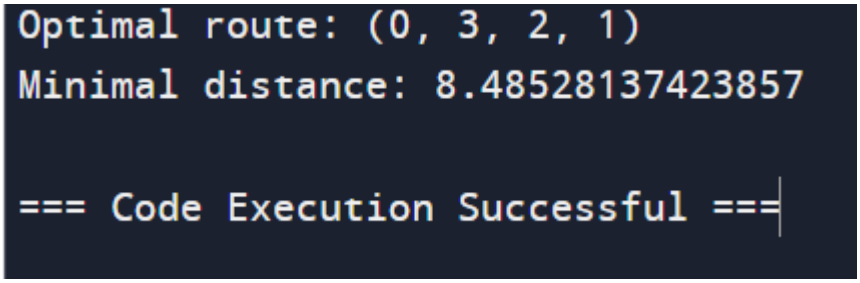
101. Traveling Salesman Problem
PROGRAM:-

```
import itertools

def calculate_distance(city1, city2):
    return ((city1[0] - city2[0]) ** 2 + (city1[1] - city2[1]) ** 2) ** 0.5

def total_distance(route, distance_matrix):
    return sum(distance_matrix[route[i]][route[i + 1]] for i in range(len(route) - 1)) +
distance_matrix[route[-1]][route[0]]

def travelling_salesman_brute_force(cities):
    n = len(cities)
    distance_matrix = [[calculate_distance(cities[i], cities[j]) for j in range(n)] for i in range(n)]

    min_route = None
    min_distance = float('inf')

    for route in itertools.permutations(range(n)):
        current_distance = total_distance(route, distance_matrix)
        if current_distance < min_distance:
            min_distance = current_distance
            min_route = route

    return min_route, min_distance

# Example usage
cities = [(0, 0), (1, 1), (2, 2), (3, 3)]  # Example coordinates of 4 cities
route, distance = travelling_salesman_brute_force(cities)
print("Optimal route:", route)
print("Minimal distance:", distance)
```

OUTPUT:-



TIME COMPLEXITY:-O(n!)