**9. Write the python to implement Travelling Salesman Problem**

**Program:**

import itertools

def calculate\_total\_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 to start

return total\_distance

def tsp\_brute\_force(distance\_matrix):

n = len(distance\_matrix)

cities = list(range(n))

min\_path = None

min\_distance = float('inf')

for perm in itertools.permutations(cities[1:]): # Fix the start at city 0

current\_path = [0] + list(perm)

current\_distance = calculate\_total\_distance(current\_path, distance\_matrix)

if current\_distance < min\_distance:

min\_distance = current\_distance

min\_path = current\_path

return min\_path, min\_distance

# Example usage

distance\_matrix = [

[0, 10, 15, 20],

[10, 0, 35, 25],

[15, 35, 0, 30],

[20, 25, 30, 0]

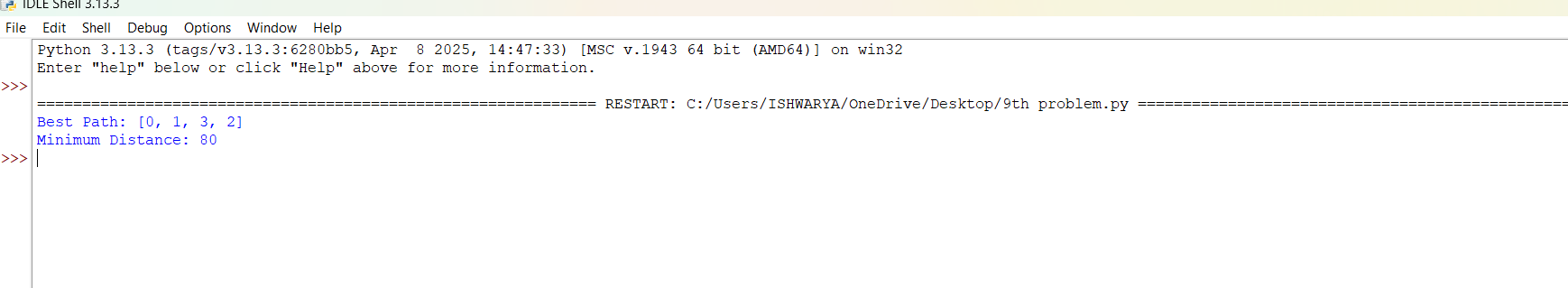
]

best\_path, best\_distance = tsp\_brute\_force(distance\_matrix)

print("Best Path:", best\_path)

print("Minimum Distance:", best\_distance)

**output:**

****