EXP 1 :-

PROGRAM :

import itertools

def tsp\_brute\_force(graph):

all\_nodes = set(graph.keys())

start\_node = next(iter(all\_nodes))

min\_cost = float('inf')

min\_path = None

for path in itertools.permutations(all\_nodes - {start\_node}):

current\_path = [start\_node] + list(path) + [start\_node]

total\_cost = sum(graph[current\_path[i]][current\_path[i + 1]] for i in range(len(current\_path) - 1))

if total\_cost < min\_cost:

min\_cost = total\_cost

min\_path = current\_path

return min\_path, min\_cost

# Example graph representation (node: {neighbor: cost})

graph = {

'A': {'B': 10, 'C': 15, 'D': 20},

'B': {'A': 10, 'C': 35, 'D': 25},

'C': {'A': 15, 'B': 35, 'D': 30},

'D': {'A': 20, 'B': 25, 'C': 30}

}

best\_path, min\_distance = tsp\_brute\_force(graph)

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

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

OUTPUT :-

