**Program:**

from queue import PriorityQueue

def ucs(start, goal, graph): queue = PriorityQueue() queue.put((0, start)) visited = set()

while not queue.empty():

cost, node = queue.get()

if node == goal:

return cost

if node not in visited:

visited.add(node)

for neighbor, weight in graph[node].items():

if neighbor not in visited:

queue.put((cost + weight, neighbor))

return -1

graph = {}

noOfVertices = int(input("Enter the number of vertices: "))

for i in range(noOfVertices):

vertex = input(f"\nEnter vertex {i+1}: ")

numOfNeighbors = int(input("Enter the number of neighbors: ")) neighbors = {}

for j in range(numOfNeighbors):

neighbor = input(f"Enter neighbor {j+1}: ")

weight = int(input(f"Enter the weight of edge ({vertex}, {neighbor}): "))

neighbors[neighbor] = weight

graph[vertex] = neighbors

start = input("\nEnter the start node: ")

goal = input("Enter the goal node: ")

cost = ucs(start, goal, graph)

if cost == -1:

print("\nPath not found.")

else:

print(f"\nThe minimum cost from {start} to {goal} = {cost}")

**Output:**

