LAB TEST - 2

1BM18CS064
Nitesh Garg

Program 3

```
class Topology:
    def __init__(self, array_of_points):
        self.nodes = array_of_points
        self.edges = [ ]

    def add_direct_connection(self, p1, p2, cost):
        self.edges.append((p1, p2, cost))
        self.edges.append((p2, p1, cost))

    def distance_vector_routing(self):
        import collections
        for node in self.nodes:
            dist = collections.defaultdict(int)
            next_hop = {node: node}
            for other_node in self.nodes:
                if other_node != node:
                    dist[other_node] = 999999

        for i in range(len(self.nodes)-1):
            for edge in self.edges:
                src, dest, cost = edge
                if dist[src] + cost < dist[dest] :
                    dist[dest] = dist[src] + cost
                    if src == node:
                        next_hop[dest] = dest
                    elif src in next_hop:
                        next_hop[dest] = next_hop[src]
```

```
def print_routing_table(self, node, dist, next_hop):
    print(f'Routing table for {node}:')
    print('Dest \t Cost \t Next hop')
    for dest, cost in dist.items():
        print(f'{dest} \t {cost} \t {next_hop[dest]}')

nodes = ['A', 'B', 'C', 'D', 'E', 'F', 'G']

t = topology(nodes)

~~t.add_direct_connection('A', 'B',~~
t.add_diret_connection('A', 'B', 2)
        .
        .
        .
        .

t.distance_vector_routing()
```