Name - Nimit Sejal
USN - 1BM18CS065
Date - 31/12/20

CN - Lab Test (2)

Program (1)

```
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 collection
        for node in self. nodes :
            dist = collection. defaultdict (int).
            nexthop = { node : node }.
            for other_node in self. nodes :
                if other_node != node :
                    dist [other_node] = 100000000 # ∞

        for i in range (len ( self. nodes) -1):
            for edge in self. edges :
                src, dest, cost = edge.
```

(PTO)

```
        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] .

    self.print_routing_table(node, dist, next_hop) .
    print() .

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) .
```

PTO

```
t.add_direct_connection('A','B',4)
t.add direct connection('A','D',3)
t.add_direct_connection('A','C',5)
t.add-direct_connection('B','C'2)
t.add_direct_connection('B','F',3)
t.add_direct_connection('B','G',4)
t.add_direct_connection('C','D',6)
t.add_direct_connection('C','E',4)
t.add_direct_connection('C','F',4)
t.add_direct_connection('D','E',3)
t.add_direct_connection('E','F',2)
t.add_direct_connection('F','G',5)
```