Chandan A.M.
1BM18CS025

CN Lab

1. Distance Vector Routing :

```
class Network:
    def _init_ (self, n):
        self.matrix = []
        self.n = n
    def addEdge (self, s, d, w):
        self.matrix.append ((s,d,w))
    def printPath (self, dist, src):
        print ("Vector table of {}".format(chr (ord('A')
                                                   +src)))
        print (" Dest\t Cost")
        for i in range(self.n):
            print (."{0}\t {1}".format(chr(ord('A')
                                              +i,
                                              dist(i)))

    def algo (self, src):
        dist = [99] * self.n
        dist [src] = 0
        for _ in range (self.n - 1):
            for s,d,w in self.matrix :
                if dist [s] != 99 and dist[s]+w < dist[d]:
                    dist [d] = dist [s]+w
        self.printPath (dist, src)
```

```
if __name__ = = "__main__" :
        matrix = []
        n = int ( input ("Enter no. of Routers "))
        print ("Enter the adjecency matrix ")
        for i in range (n) :
            row = list [map (int , input.split(" " ))
            matrix .append (row)
        g = Graph(n)
        for i in range(n) :
            for j in range(n) :
                if matrix [i][j] = = 1 :
                    g.addEdge (i, j, matrix[i, j])
        for i in range (n) :
            g.algo(n)
```