J.J. Meghana
1BM18CS039

⇒ Write a program for distance vector algorithm to find suitable path for transmission.

Code

```
class distvector:
    def __init__(self, n):
        self.matrix = []
        self.n = n
    def newEdge(self, u, v, w):
        self.matrix.append((u, v, w))
    def display(self, dist, src):
        print("Vector table of {}".format
                                (chr(ord('A')+src)))
        for i in range(self.n):
            print("{0}\t{1}".format(chr(ord('A')+i),
                                    dist[i]))
    def start(self, src):
        dist = [90] * self.n
        dist[src] = 0
        for _ in range(self.n-1):

            for u, v, w in self.matrix:
                if dist[u] != 99 and dist[u] + w <
                                        dist[v]:
                    dist[v] = dist[u]+w
        self.display(dist, src)

matrix=[]
print("Enter the routers:")
n = int(input())
```

Meg

J.S. Meghana
1BM18CS039

```
print("Enter the adjacency matrix
                (enter 999 for infinity): ")
for i in range(n):
    m = list(map(int, input().split(" ")))
    matrix.append(m)
g = distVector(n)
for i in range(n):
    for j in range(n):
        if matrix[i][j] == 1:
            g.newEdge(i, j, 1)

for _ in range(n):
    g.start(_)
```

Meg.