Write up:

```
Class Network:
    def __init__(self, n):
        self.matrix = []
        self.n = n
    def linkadd(self, u, v, w):
        self.matrix.append(u, v, w)
    def printTheTable(self, dist, sr):
        print("Router table entries for route {}".format(chr(ord('A')+sr)))
        print("{0}\t{1}".format("Dest", "Cost"))
        for i in range(self.n):
            print("{0}\t{1}".format(chr(ord('A')+i), dist[i]))
    def solution(self, src):
        dist = [99]*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.printTheTable(dist, src)

if __name__ == "__main__":
    matrix = []
    n = int(input("no.of routers"))
    print("Adjacency matrix")
    for i in range(n):
        row = list(map(int, input().split(" ")))
        matrix.append(row)
    g = Network(n)
    for i in range(n):
        for j in range(n):
            if matrix[i][j] == 1:
                g.link add (i,j,1)
        for i in range(n):
            g.solution(i)
```

Output:

```
no. of routers : 3
Adjacency matrix:
  0  1  99
 99  0  1
  1  99  0
Router Table entries for router A
   Dest    Cost
    A       0
    B       1
    C       2
Router Table entries for router B
   Dest    Cost
    A       2
    B       0
    C       1
Router Table entries for router C
   Dest   Cost
    A       1
    B       2
    C       0
```