DVR:

```
#include <stdio.h>
int costMatrix[20][20], n;
struct routers
{int distance[20];
 int adjNodes[20];} node[20];
void readCostMatrix()
{int i, j;
 printf("\nEnter cost matrix\n");
for (i = 0; i < n; ++i)
{for (j = 0; j < n; ++j)
{ scanf("%d", &costMatrix[i][j]);
costMatrix[i][i] = 0;
node[i].distance[j] = costMatrix[i][j];
node[i].adjNodes[j] = j;
}}}
void calcRoutingTable()
{int i, j, k;
 for (i = 0; i < n; ++i){
for (j = 0; j < n; ++j)
 {for (k = 0; k < n; ++k){
if (node[i].distance[j] > costMatrix[i][k] +
node[k].distance[j])
{node[i].distance[j] = node[i].distance[k] +
node[k].distance[j];
node[i].adjNodes[j] = k;
}}}}}
void displayRoutes()
{int i, j;
for (i = 0; i < n; ++i){
printf("\nRouter %d\n", i + 1);
for (j = 0; j < n; ++j)
{printf("Node %d via %d : Distance %d\n", j + 1,
node[i].adjNodes[j] + 1,
node[i].distance[j]);}
printf("\n");}}
int main(){
int i, j;
printf("Number of nodes: ");
scanf("%d", &n);
readCostMatrix();
calcRoutingTable();
displayRoutes();
return 0;}
```

dvr

Output:

```
Number of nodes: 3
Enter cost matrix
0 1 5
1 0 2
5 2 0

Router 1
Node 1 via 1 : Distance 0
Node 2 via 2 : Distance 1
Node 3 via 2 : Distance 3

Router 2
Node 1 via 1 : Distance 1
Node 2 via 2 : Distance 0
Node 3 via 3 : Distance 2

Router 3
Node 1 via 2 : Distance 3
Node 2 via 2 : Distance 2
Node 3 via 3 : Distance 0
```