

```

1: // Given a graph, each node A knows the shortest path to node Z and node A can determine
2: // its shortest path to Z by calculating the minimum cost. Now when packet flows through a
3: // path it incurs some cost to the network, find shortest paths from source to all nodes in the
4: // given graph using Distance vector routing Algorithm.
5:
6: #include<stdio.h>
7: struct node
8: {   unsigned dist[20];
9:     unsigned from[20];
10: }rt[10];
11: int main()
12: {   int costmat[20][20];
13:     int nodes,i,j,k,count=0;
14:     printf("\nEnter the number of nodes : ");
15:     scanf("%d",&nodes);//Enter the nodes
16:     printf("\nEnter the cost matrix :\n");
17:     for(i=0;i<nodes;i++)
18:         for(j=0;j<nodes;j++)
19:             {   scanf("%d",&costmat[i][j]);
20:                 costmat[i][i]=0;//initialise the distance equal to cost matrix
21:                 rt[i].dist[j]=costmat[i][j];
22:                 rt[i].from[j]=j;
23:             }
24:     do
25:     {   count=0;
26:         //We choose arbitrary vertex k and we calculate the direct distance from
27:         //the node i to k using the cost matrix and add the distance from k to node j
28:         for(i=0;i<nodes;i++)
29:             for(j=0;j<nodes;j++)
30:                 for(k=0;k<nodes;k++)
31:                     if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
32:                         //We calculate the minimum distance
33:                         rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
34:                         rt[i].from[j]=k;
35:                         count++;
36:                     }
37:     }while(count!=0);
38:     for(i=0;i<nodes;i++)
39:     {   printf("\n\n For router %d\n",i+1);
40:         for(j=0;j<nodes;j++)
41:             printf("\t\nNode %d via %d Distance = %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
42:     }
43:     printf("\n\n");
44: }

```