**3.Routing algorithm-Distance Vector Routing Algorithm**

/\*

Distance Vector Routing in this program is implemented using Bellman Ford Algorithm:-

\*/

#include<stdio.h>

**struct** node

{

**unsigned** dist[20];

**unsigned** from[20];

}rt[10];

**int** main()

{

**int** costmat[20][20];

**int** nodes,i,j,k,count=0;

printf("\nEnter the number of nodes : ");

scanf("%d",&nodes);//Enter the nodes

printf("\nEnter the cost matrix :\n");

**for**(i=0;i<nodes;i++)

{

**for**(j=0;j<nodes;j++)

{

scanf("%d",&costmat[i][j]);

costmat[i][i]=0;

rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost matrix

rt[i].from[j]=j;

}

}

**do**

{

count=0;

**for**(i=0;i<nodes;i++)//We choose arbitary vertex k and we calculate the direct distance from the node i to k using the cost matrix

//and add the distance from k to node j

**for**(j=0;j<nodes;j++)

**for**(k=0;k<nodes;k++)

**if**(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])

{//We calculate the minimum distance

rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];

rt[i].from[j]=k;

count++;

}

}**while**(count!=0);

**for**(i=0;i<nodes;i++)

{

printf("\n\n For router %d\n",i+1);

**for**(j=0;j<nodes;j++)

{

printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);

}

}

printf("\n\n");

getch();

}

/\*

A sample run of the program works as:-

Enter the number of nodes :

3

Enter the cost matrix :

0 2 7

2 0 1

7 1 0

For router 1

node 1 via 1 Distance 0

node 2 via 2 Distance 2

node 3 via 3 Distance 3

For router 2

node 1 via 1 Distance 2

node 2 via 2 Distance 0

node 3 via 3 Distance 1

For router 3

node 1 via 1 Distance 3

node 2 via 2 Distance 1

node 3 via 3 Distance 0

\*/