|  |
| --- |
| CODE |

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*File : p8.cpp

\*Description : Write a program for distance vector algorithm to find suitable path for transmission.

\*Author : Arpith (1PE10CS018)

\*Tools : Ubuntu 13.04, gcc 4.7.3 compiler, Code::Blocks

\*Date : 25 October 2013

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <stdio.h>

using namespace std;

struct node

{

int dist[20];

int from[20];

}rt[10];

int main()

{

int distance\_matrix[20][20];

int n,i,j,k,count=0,src,dest;

cout<<"\nEnter the number of nodes : ";

cin>>n;

cout<<"\nEnter the cost/distance matrix :\n";

for(i=0;i<n;i++)

for(j=0;j<n;j++)

{

cin>>distance\_matrix[i][j];

distance\_matrix[i][i]=0;

rt[i].dist[j]=distance\_matrix[i][j];

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

}

do

{

count=0;

for(i=0;i<n;i++)

for(j=0;j<n;j++)

for(k=0;k<n;k++)

if(rt[i].dist[j]>distance\_matrix[i][k]+rt[k].dist[j])

{

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<n;i++)

{

cout<<"\nRouting table for router"<<i+1<<":\nDest\tNextHop\tDist\n";

for(j=0;j<n;j++)

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

}

cout<<"\nEnter source and destination : ";

cin>>src>>dest;

src--; dest--;

printf("Shortest path : \n Via router : %d\n Shortest distance : %d \n",rt[src].from[dest]+1,rt[src].dist[dest]);

return 0;

}

|  |
| --- |
| **OUTPUT** |

