program-13

#include<stdio.h>

int min(int,int);

void floyds(int p[10][10],int n)

{

int i,j,k;

for(k=1;k<=n;k++)

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if(i==j)

p[i][j]=0;

else

p[i][j]=min(p[i][j],p[i][k]+p[k][j]);

}

int min(int a,int b)

{

if(a<b)

return(a);

else

return(b);

}

void main()

{

int p[10][10],w,n,e,u,v,i,j;;

printf("\n Enter the number of vertices:");

scanf("%d",&n);

printf("\n Enter the number of edges:\n");

scanf("%d",&e);

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

p[i][j]=999;

}

for(i=1;i<=e;i++)

{

printf("\n Enter the end vertices of edge%d with its weight \n",i);

scanf("%d%d%d",&u,&v,&w);

p[u][v]=w;

}

printf("\n Matrix of input data:\n");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

printf("%d \t",p[i][j]);

printf("\n");

}

floyds(p,n);

printf("\n The shortest paths are:\n");

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

{

if(i!=j)

printf("\n <%d,%d>=%d",i,j,p[i][j]);

}

}

output:

[shilpa@shilpa-linux ADA\_LAB-PROG-main]$ ./shortest\_path

Enter the number of vertices:4

Enter the number of edges:

4

Enter the end vertices of edge1 with its weight

1 2 10

Enter the end vertices of edge2 with its weight

2 3 20

Enter the end vertices of edge3 with its weight

3 1 40

Enter the end vertices of edge4 with its weight

3 4 50

Matrix of input data:

999 10 999 999

999 999 20 999

40 999 999 50

999 999 999 999

The shortest paths are:

<1,2>=10

<1,3>=30

<1,4>=80

<2,1>=60

<2,3>=20

<2,4>=70

<3,1>=40

<3,2>=50

<3,4>=50

<4,1>=999

<4,2>=999

<4,3>=999[shilpa@shilpa-linux ADA\_LAB-PROG-main]$