

Floyd's Algorithm

```
#include<stdio.h>
#include<conio.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("\nEnter the number of vertices: ");
    scanf("%d",&n);
    printf("\nEnter the number of edges: ");
    scanf("%d",&e);
    for (i=1;i<=n;i++) {
        for (j=1;j<=n;j++)
            if(i ==j)
                p[i][j] = 0;
            else
                p[i][j]=999;
    }
    for (i=1;i<=e;i++) {
        printf("\nEnter the end vertices of edge %d with its weight\n",i);
        scanf("%d%d%d",&u,&v,&w);
        p[u][v]=w;
    }
    printf("\nMatrix 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\nThe final matrix is:\n");
    for (i=1;i<=n;i++) {
        for (j=1;j<=n;j++)
            printf("%d \t",p[i][j]);
        printf("\n");
    }
    printf("\n\nThe 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]);
        }
    }
    getch();
}

```

OUTPUT:

```
Enter the number of vertices: 4

Enter the number of edges: 5

Enter the end vertices of edge 1 with its weight
1 2 3

Enter the end vertices of edge 2 with its weight
2 4 1

Enter the end vertices of edge 3 with its weight
4 1 6

Enter the end vertices of edge 4 with its weight
2 3 7

Enter the end vertices of edge 5 with its weight
3 1 2

Matrix of input data:
0      3      999      999
999     0       7       1
2      999     0      999
6      999     999     0

The final matrix is:
0      3      10      4
7      0       7       1
2      5       0       6
6      9      16      0

The shortest paths are:

<1,2>=3
<1,3>=10
<1,4>=4
<2,1>=7
<2,3>=7
<2,4>=1
<3,1>=2
<3,2>=5
<3,4>=6
<4,1>=6
<4,2>=9
<4,3>=16
```