```cpp
#include<iostream.h>
#include<conio.h>

void main()
{
    int graph[50][50];
    int i,j,k,t;
    int nn;

    cout<<"\n Enter Number of Nodes:";
    cin>>nn;

    /* Initialize graph*/
    for (i=0;i<nn;i++)
    {
        for(j=0;j<nn;j++)
        {
            graph[i][j]=-1;
        }
    }

    /* Vertex names */
    char ch[7]={'A','B','C','D','E','F','G'};

    /* Get input */
    for (i=0;i<nn;i++)
    {
        for(j=0;j<nn;j++)
        {
            if(i==j)
            {
                graph[i][j]=0;
            }
            if(graph[i][j]==-1)
            {
                cout<<"\n Enter Distance between "<<ch[i]<<" - "<<ch[j]<<" : ";
                cin>>t;
                graph[i][j]=graph[j][i]=t;
            }
        }
    }

    /* Initializing via */
    int via[50][50];
    for (i=0;i<nn;i++)
    {
        for(j=0;j<nn;j++)
        {
            via[i][j]=-1;
        }
    }

    cout<<"\n After Initialization";
    /* Display table initialization */
    for (i=0;i<nn;i++)
    {
        cout<<"\n"<<ch[i]<<" Table";
        cout<<"\nNode\tDist\tVia";
        for(j=0;j<nn;j++)
        {
            cout<<"\n"<<ch[j]<<"\t"<<graph[i][j]<<"\t"<<via[i][j];
        }
    }
```

```
//sharing table
int sh[50][50][50];
for(i=0;i<nn;i++)
{
    for(j=0;j<nn;j++)
    {
        for (k=0;k<nn;k++)
        {
            /* Check if edge is present or not*/
            if((graph[i][j]>-1)&&(graph[j][k]>-1))
            {
                sh[i][j][k]=graph[j][k]+graph[i][j];
            }
            else
            {
                sh[i][j][k]=-1;
            }
        }
    }
}

/*displaying shared table */
for(i=0;i<nn;i++)
{
    cout<<"\n\n For "<<ch[i];
    for (j=0;j<nn;j++)
    {
        cout<<"\n From "<<ch[j];
        for(k=0;k<nn;k++)
        {
            cout<<"\n "<<ch[k]<<" "<<sh[i][j][k];
        }
    }
}

/* Updating */
int final[50][50];
for(i=0;i<nn;i++)
{
    for(j=0;j<nn;j++)
    {
        /* Copy initial value from input graph*/
        final[i][j]=graph[i][j];
        via[i][j]=i;

        /*Update them*/
        /* Check condition a - b - c */
        for(k=0;k<nn;k++)
        {

            if((final[i][j]>sh[i][k][j]) || (final[i][j] == -1))
            {
                if(sh[i][k][j]>-1)
                {
                    final[i][j]=sh[i][k][j];
                    via[i][j]=k;
                }
            }
        }
        /* After considering three vertex if final not found
            consider 4th
            a- b- c- d
```

```
            */

            if(final[i][j]==-1)
            {
                for(k=0;k<nn;k++)
                {

                    if((final[i][k]!=-1)&&(final[k][j]!=-1))
                    {
                        if((final[i][j]==-1) || ((final[i][j]!=-1) &&(final[i][j]>final[i][k]
                        +final[k][j])))
                        {
                            if(final[i][k]+final[k][j]>-1)
                            {
                                final[i][j]=final[i][k]+final[k][j];
                                via[i][j]=k;
                            }
                        }
                    }

                }
            }
        }
    }

    cout<<"\n After Update :";
    /* Display table Updation */
    for (i=0;i<nn;i++)
    {
        cout<<"\n"<<ch[i]<<" Table";
        cout<<"\nNode\tDist\tVia";
        for(j=0;j<nn;j++)
        {
            cout<<"\n"<<ch[j]<<"\t"<<final[i][j]<<"\t";
            if(i==via[i][j])
                cout<<"-";
            else
                cout<<ch[via[i][j]];
        }
    }

    getch();
}
```