### Implement All Pair Shortest paths problem using Floyd's algorithm.

#### Code:

#include<iostream>

using namespace std;

#define V 4

void print(int dist[][V])

{

    cout<<"Distance matrix\n";

    for (int i=0;i<V;i++)

    {

        for (int j=0;j<V;j++)

        {

                cout<<dist[i][j]<<" ";

        }

       cout<<endl;

    }

}

void floyd(int graph[][V])

{

    int dist[V][V];

    int i, j, k;

    for (i = 0; i < V; i++)

        for (j = 0; j < V; j++)

        {

          if(graph[i][j]==-1)

              dist[i][j]=999;

          else

            dist[i][j] = graph[i][j];

        }

    for (k = 0; k < V; k++)

    {

        for (i = 0; i < V; i++)

        {

            for (j = 0; j < V; j++)

            {

                if (dist[i][k] + dist[k][j] < dist[i][j])

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

            }

        }

    }

    print(dist);

}

int main()

{

   int g[V][V];

    cout<<"Enter adjacency matrix\nEnter -1 if no edges between vertices\n";

    for (int i=0;i<V;i++)

    {

        for(int j=0;j<V;j++)

        {

            cin>>g[i][j];

        }

    }

    floyd(g);

    return 0;

}

#### Output:

