Program to Implement Distance Vector Routing algorithm

```
#include<stdio.h>
#include<iostream>
using namespace std;

struct node
{
   unsigned dist[6];
   unsigned from[6];
}DVR[10];
int main()
{
   cout<<"\n\n PROGRAM TO IMPLEMENT DISTANCE VECTOR ROUTING
ALGORITHM ";
   int costmat[6][6];
   int nodes, i, j, k;
   cout<<"\n\n Enter the number of nodes : ";
   cin>>nodes; //Enter the nodes
   cout<<"\n Enter the cost matrix : \n" ;
   for(i = 0; i < nodes; i++)
    {
      for(j = 0; j < nodes; j++)
      {
         cin>>costmat[i][j];
         costmat[i][i] = 0;
         DVR[i].dist[j] = costmat[i][j]; //initialise the distance equal to cost
matrix
         DVR[i].from[j] = j;
      }
   }
         for(i = 0; i < nodes; i++) //We choose arbitary vertex k and we calculate
the
         //direct distance from the node i to k using the cost matrix and add the
distance from k to node j
         for(j = i+1; j < nodes; j++)
         for(k = 0; k < nodes; k++)
            if(DVR[i].dist[j] > costmat[i][k] + DVR[k].dist[j])
            {  //We calculate the minimum distance
               DVR[i].dist[j] = DVR[i].dist[k] + DVR[k].dist[j];
               DVR[j].dist[i] = DVR[i].dist[j];
               DVR[i].from[j] = k;
               DVR[j].from[i] = k;

            }
      for(i = 0; i < nodes; i++)
      {
         cout<<"\n\n For router: "<<i+1;
         for(j = 0; j < nodes; j++)
```

```
            cout<<"\t\n node "<<j+1<<" via "<<DVR[i].from[j]+1<<" Distance
"<<DVR[i].dist[j];
      }
   cout<<" \n\n ";
   return 0;
}
```