/* Write a program for distance vector algorithm to find suitable path for transmission */

C. Karan Naidu
1BM18CS042

```
import java.io.*;
public class DistanceVector
{
static int graph[][];
static int via[][];
static int rt[][];
static int v;
static int e;
public static void main(String args[]) throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader
                                        (System.in));
System.out.println("Please enter the number of vertices:");
v = Integer.parseInt(br.readLine());
System.out.println("Please enter the number of Edges");
  e = Integer.parseInt(br.readLine());
System.out.println("Please enter the numb
 graph = new int[v][v];
 via = new int[v][v];
 rt = new int[v][v];
  for(int i=0;i<v;i++)
 for(int j=0;j<v;j++)
  { if(i==j)
  graph[i][j]=0;
  else graph[i][j]=9999;
  }
```

```
for(int i=0; i<e; i++)
{
System.out.println("Please enter data for Edge "+(i+1)+":");
System.out.print("Source:");
int s= Integer.parseInt(br.readLine());
s--;
System.out.print("Destination:");
int d = Integer.parseInt(br.readLine());
d--;
System.out.print("Cost:");
int c= Integer.parseInt(br.readLine());
graph[s][d]=c;
graph[d][s]=c;
}
dvr_calc_disp("The initial Routing Tables are:");
System.out.println("Please enter the Source Node for the edge whose
    cost has changed");
int s= Integer.parseInt(br.readLine());
s--;
System.out.println("Please enter the Destination Node for the edge.
whose cost has changed:");
int d = Integer.parseInt(br.readLine());
d--;
System.out.println("Ente the new cost");
int c = Integer.parseInt(br.readLine());
graph[s][d]=c;
graph[d][s]=c;
dvr_calc_disp("The new Routing Tables are:");
```

C. Karan Kumar
IBM 18CS042.

```
static void dvr_calc_disp(String message)
{
 System.out.println();
 init_tables();
 update_tables();
System.out.println(message);
 print_tables();
Sop ln();
}
static void update_table(int source)
{
for(int i=0;i<v;i++)
 if(graph[source][i]!=9999)
{int dist = graph[source][i];
for(int j=0;j<v;j++)
{ int inter_dist= rt[i][j];
if(via[i][j]==source)
 inter_dist = 9999;
if (dist + inter_dist < rt[source][j])
{ rt[source][j]=dist+inter_dist;
  via[source][j]=i;
}
}
}
}
}
```

C. Karan Naidu
1BM18CS042

```
static void init_tables()
{
 for(int i=0; i<v ;i++)
 {for(int j=0; j<v ; j++)
{
  if(i==j)
  {rt[i][j]=0;
via[i][j]=i;
}
else
{ rt[i][j]=9999;
via[i][j]=100;
}
}
}
}
static void print_tables()
{
 for(int i=0; i<v; i++)
{
 for(int j=0; j<v ;j++)
 { System.out.println("Dist: "+rt[i][j]+"    ");
   }
   System.out.println();
   }
  }
```