CN-Lab.

Divyakirti Masalem
1BM18CS029

Write Up:

Q. Write a program for distance vector algorithm to find suitable path for transmission.

Code:

```
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(Syst
  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());

  graph = new int[v][v];
   via = new int[v][v];
   rt = new int[v][v];
   for (int i=0; i<v; i++)
```


Divyakirti

Divyakirti Masaun
IBM18CS029

```
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.println("Please enter data for Edge"+(i+1)+":");
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.print("Please enter the source node for this edge
                  whose cost has changed:");
```

Divyakirti Masam
IBM18CS029

```
int s = Integer.parseInt(br.readLine());
s--;
System.out.println("Please enter the Destination Node of the
                    edge whose cost has changed:");
int d = Integer.parseInt(br.readLine());
d--;
System.out.print("Please enter 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:");
}
static void dvr_disp(String message)
{
    System.out.println();
    int_table();
    update_tables();
    System.out.println(message);
    print_tables();
    System.out.println();
}
static void update_table(int source)
{
    for(int i=0; i<v; i++)
    {
        if(graph[source][i]!=9999)
        {
```


Divyakirti

Divyakriti Masaun
IBM18CS029

```
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 + int dist < rt[source][j])
    {
        rt[source][j] = dist + inter_dist;
        via[source][j] = 1;
    }
}
}

static void update_table()
{
    int k = 0;
    for (int i = 0; i < 4 * v; i++)
    {
        update_table(k);
        k++;
        if (k == v)
            k = 0;
    }
}
static void init_tables()
{
```


Divyakriti

Divyakirti Masaun
IBM18CS029

```
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.print("Dist: "+ rt[i][j] +"  ");
      }
      System.out.println();
    }
  }
}
```


@Divyakirti