import java.util.Scanner;

public class Dijkstra {

public static void main(String args[]) {

int i,j;

int dis[]=new int[10],visited[]=new int[10];

int cost[][]=new int[10][10],path[]=new int[10];

Scanner in=new Scanner(System.in);

System.out.println("Enter the number of nodes");

int n=in.nextInt();

System.out.println("Enter the cost matrix");

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

cost[i][j]=in.nextInt();

System.out.println("The enterd cost matrix is ");

for(i=1;i<=n;i++)

{

for(j=1;j<=n;j++)

System.out.print(cost[i][j]+"\t");

System.out.println();

}

System.out.println("Enter the source vertex");

int sv=in.nextInt();

dij(cost,dis,sv,n,path,visited);

printpath(sv,n,dis,path,visited);

}

static void dij(int cost[][],int dist[],int sv,int n, int path[],int visited[])

{

int c=2,min,v=0;

for(int i=1;i<=n;i++)

{

visited[i]=0;

dist[i]=cost[sv][i];

if(cost[sv][i]==999)

path[i]=0;

else

path[i]=sv;

}

visited[sv]=1;

while(c<=n) {

min=999;

for(int w=1;w<=n;w++)

if(dist[w]<min&&visited[w]==0)

{

min=dist[w];

v=w;

}

visited[v]=1;

c++;

for(int w=1;w<=n;w++) {

if(dist[w]>dist[v]+cost[v][w])

{

dist[w]=dist[v]+cost[v][w];

path[w]=v;

}

}

}

}

static void printpath(int sv,int n,int dist[],int path[],int visited[])

{

for(int w=1;w<=n;w++)

{

if(visited[w]==1&&w!=sv)

{

System.out.println("The shortest distance between "+sv+"--->"+w+" is "+dist[w]);

int t=path[w];

System.out.println("The path is :");

System.out.print(" "+w);

while(t!=sv)

{

System.out.print("<--->"+t);

t=path[t];

}

System.out.println("<--->"+sv);;

}

}

}

}