**DIJKSTRA’S ALGORITHM**

#include<iostream.h>

void dij(int,int[20][20],int[20],int[20],int);

void main()

{

int i,j,n,visited[20],source,cost[20][20],d[20];

cout<<"enter no. of vertices:";

cin>>n;

cout<<"enter the cost adjacency matrix\n";

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

{

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

{

cin>>cost[i][j];

}

}

cout<<"\nEnter the source node:";

cin>>source;

dij(source,cost,visited,d,n);

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

if(i!=source)

cout<<"\n shortest path from"<<source<<"to"<<i<<"is"<<d[i];}

void dij(int source,int cost[20][20],int visited[20],int d[20],int n)

{

int i,j,min,u,w;

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

{

visited[i]=0;

d[i]=cost[source][i];

}

visited[source]=1;

d[source]=0;

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

{

min=999;

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

{

if(!visited[i])

{

if(d[i]<min)

{

min=d[i];

u=i;

}

}

}

visited[u]=1;

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

{

if(cost[u][w]!=999 &&visited[w]==0)

{

if(d[w]>cost[u][w]+d[u])

d[w]=cost[u][w]+d[u];

}

}

}

}

**OUTPUT:**

Enter no. of vertices: 5

Enter the cost Adjacency matrix:

0 5 12 17 999

999 0 999 8 7

999 9999 0 9 999

999 999 999 0 999

999 999 999 999 0

Enter the source node: 1

shortest path from1 to 2 is 5

shortest path from1 to 3 is 12

shortest path from1 to 4 is 13

shortest path from1 to 5 is 12