**Program 4 Design and implement C/C++ Program to find shortest paths from a given vertex in a**

**weighted connected graph to other vertices using Dijkstra's algorithm**

**#i**nclude<stdio.h>

#define INF 999

void dijkstra(int c[10][10],int n,int s,int d[10])

{

int v[10],min,u,i,j;

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

{

d[i]=c[s][i];

v[i]=0;

}

v[s]=1;

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

{

min=INF;

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

if(v[j]==0 && d[j]<min)

{

min=d[j];

u=j;

}

v[u]=1;

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

if(v[j]==0 && (d[u]+c[u][j])<d[j])

d[j]=d[u]+c[u][j];

}

}

int main()

{

int c[10][10],d[10],i,j,s,sum,n;

printf("\nEnter n value:");

scanf("%d",&n);

printf("\nEnter the graph data:\n");

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

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

scanf("%d",&c[i][j]);

printf("\nEnter the souce node:");

scanf("%d",&s);

dijkstra(c,n,s,d);

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

printf("\nShortest distance from %d to %d is %d",s,i,d[i]);

return 0;

}

Program 5: Design and implement C/C++ Program to obtain the Topological ordering of vertices in a

given digraph

#include<stdio.h>

int temp[10],k=0;

void sort(int a[][10],int id[],int n)

{

int i,j;

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

{

if(id[i]==0)

{

id[i]=-1;

temp[++k]=i;

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

{

if(a[i][j]==1 && id[j]!=-1)

id[j]--;

}

i=0;}}}

void main()

{

int a[10][10],id[10],n,i,j;

// clrscr();

printf("\nEnter the n value:");

scanf("%d",&n);

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

id[i]=0;

printf("\nEnter the graph data:\n");

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

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

{

scanf("%d",&a[i][j]);

if(a[i][j]==1)

id[j]++;

}

sort(a,id,n);

if(k!=n)

printf("\nTopological ordering not possible");

else

{

printf("\nTopological ordering is:");

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

printf("%d ",temp[i]);

}

}