

```

#include<stdio.h>

#include<conio.h>

#define INFINITY 9999

#define MAX 10

void dijkstra(int G[MAX][MAX],int n,int startnode);

int main()
{
    int G[MAX][MAX],i,j,n,u;
    printf("Enter no. of vertices:");
    scanf("%d",&n);
    printf("\nEnter the adjacency matrix:\n");
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    scanf("%d",&G[i][j]);
    printf("\nEnter the starting node:");
    scanf("%d",&u);
    dijkstra(G,n,u);
    return 0;
}

void dijkstra(int G[MAX][MAX],int n,int startnode)
{

    int cost[MAX][MAX],distance[MAX],pred[MAX];
    int visited[MAX],count,mindistance,nextnode,i,j;
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    if(G[i][j]==0)
    cost[i][j]=INFINITY;

```

```

else
cost[i][j]=G[i][j];
for(i=0;i<n;i++)
{
distance[i]=cost[startnode][i];
pred[i]=startnode;
visited[i]=0;
}
distance[startnode]=0;
visited[startnode]=1;
count=1;
while(count<n-1)
{
mindistance=INFINITY;
for(i=0;i<n;i++)
if(distance[i]<mindistance&&!visited[i])
{
mindistance=distance[i];
nextnode=i;
}
visited[nextnode]=1;
for(i=0;i<n;i++)
if(!visited[i])
if(mindistance+cost[nextnode][i]<distance[i])
{
distance[i]=mindistance+cost[nextnode][i];
pred[i]=nextnode;
}
count++;
}

```

```

for(i=0;i<n;i++)
if(i!=startnode)
{
printf("\nDistance of node%d=%d",i,distance[i]);
printf("\nPath=%d",i);
j=i;
do
{
j=pred[j];
printf("<-%d",j);
}while(j!=startnode);
}
}

```

```

C:\Users\nagir\OneDrive\Documents\exp 23.exe
Enter no. of vertices:5
Enter the adjacency matrix:
1 2 3 4 5 6 7 8 9 0 1 2 3 4 4 5 5 6 7 7 8 8 2 3 4 5 6 7 8 8 9 0 0 9 8 7 6 5 4
Enter the starting node:
Distance of node0=0
Path=0<-5
Distance of node1=0
Path=1<-5
Distance of node2=1
Path=2<-5
Distance of node3=0
Path=3<-5
Distance of node4=0
Path=4<-5
-----
Process exited after 40.08 seconds with return value 0
Press any key to continue . . .

```