PRACTICAL NO: 7

Name: Saloni Vishwakarma

Roll no: 13

Batch : C1

Aim: Write a c program on Dijkstra's algorithm for distance vector routing.

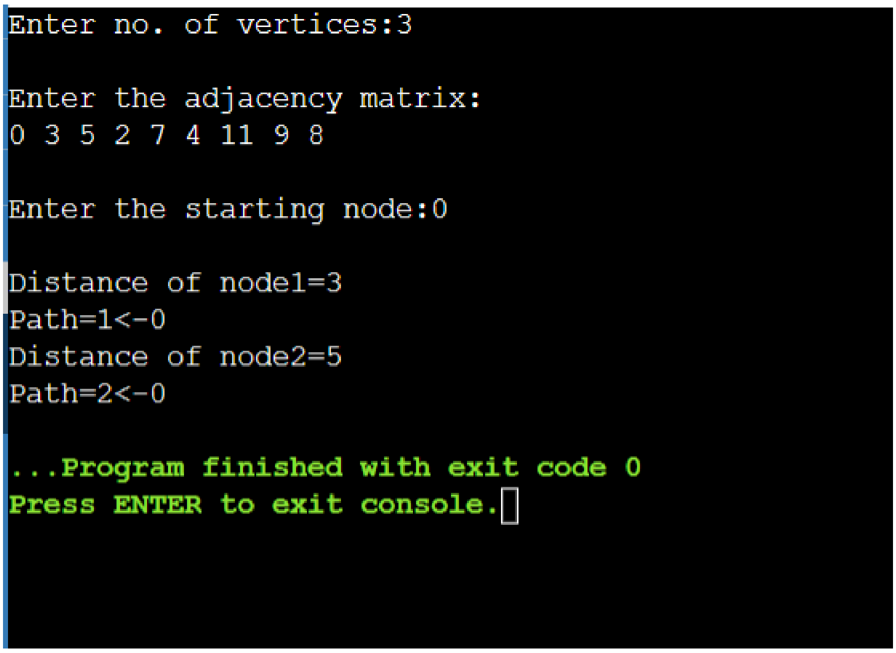
CODE:

```
#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;
//pred[] stores the predecessor of each node
//count gives the number of nodes seen so far
//create the cost matrix
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];
//initialize pred[],distance[] and visited[]
for(i=0;i<n;i++)
{
distance[i]=cost[startnode][i];
pred[i]=startnode;
```

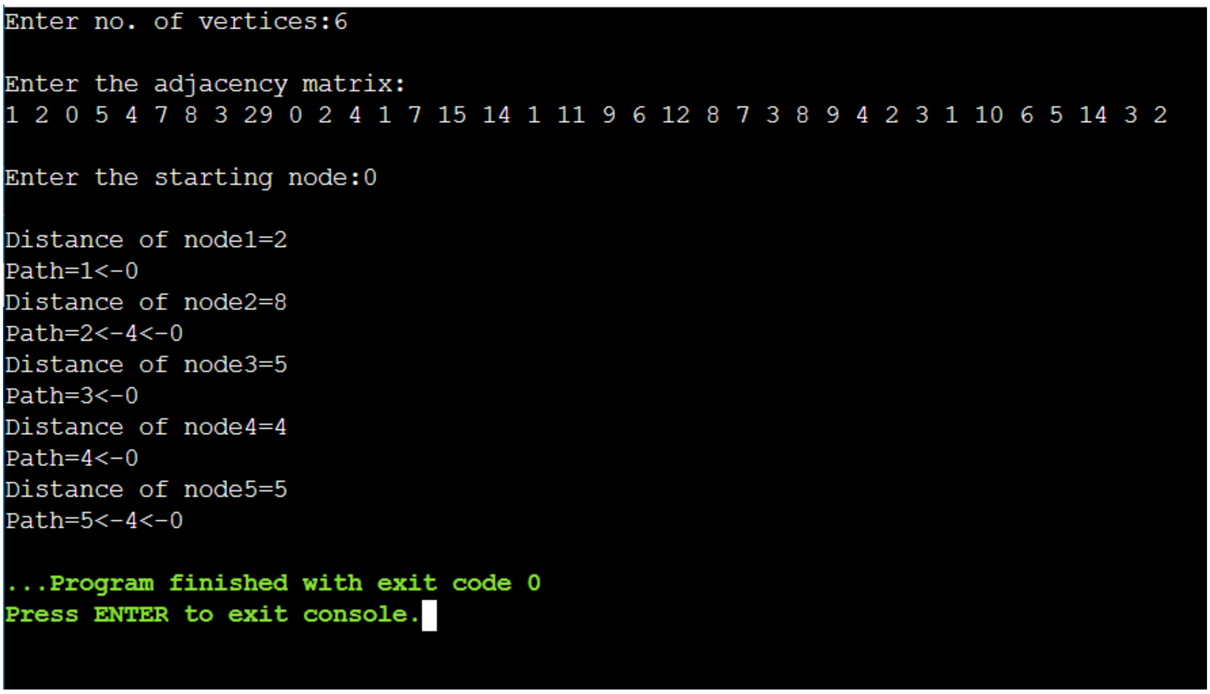
```
visited[i]=0;
}
distance[startnode]=0;
visited[startnode]=1;
count=1;
while(count<n-1)
{
mindistance=INFINITY;
//nextnode gives the node at minimum distance
for(i=0;i<n;i++)
if(distance[i]<mindistance&&!visited[i])
{
mindistance=distance[i];
nextnode=i;
}
//check if a better path exists through nextnode
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++;
}

//print the path and distance of each node
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);
}
}
```

OUTPUT:



```
Enter no. of vertices:3

Enter the adjacency matrix:
0 3 5 2 7 4 11 9 8

Enter the starting node:0

Distance of node1=3
Path=1<-0
Distance of node2=5
Path=2<-0

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter no. of vertices:6

Enter the adjacency matrix:
1 2 0 5 4 7 8 3 29 0 2 4 1 7 15 14 1 11 9 6 12 8 7 3 8 9 4 2 3 1 10 6 5 14 3 2

Enter the starting node:0

Distance of node1=2
Path=1<-0
Distance of node2=8
Path=2<-4<-0
Distance of node3=5
Path=3<-0
Distance of node4=4
Path=4<-0
Distance of node5=5
Path=5<-4<-0

...Program finished with exit code 0
Press ENTER to exit console.
```