```
#include<stdio.h>
#include<stdlib.h>
int distance[30][30],n,reach[30],path[30],dist[30];
void shortpath(int);
void printpath(int);
void shortpath(int s)
{
  int i,w,count,min;
  for(i=0;i<=n;i++)
  {
    dist[i]=distance[s][i];
    if(dist[i]!=999)
     path[i]=s;
    else
     path[i]=0;
    }
     reach[s]=1;

     for(count=2;count<=n;count++)
     {
       min=999;
       for(i=1;i<=n;i++)
       {
         if(dist[i]<min && reach[i]==0)
         {
          min=dist[i];
          w=i;
          }
          }
          reach[w]=1;
          for(i=1;i<=n;i++)
```

```
    {
      if(dist[i]>dist[w]+distance[w][i] && reach[i]==0)
       {
         dist[i]=dist[w]+distance[w][i];
         path[i]=w;
         }
         }
         }
         }
        void printpath(int s)
        {
        int i,t;
        for(i=1;i<=n;i++)
        {
          if(reach[i]==1 && i!=s)
            {
               printf("shortest distance between %d & %d is %d & \n The path is:",s,i,dist[i]);
               t=path[i];
               printf("%d",i);
               while(t!=s)
               {
                 printf("  %d",t);
                 t=path[i];
                 }
                 printf("%d",s);
                 }
                 }
                 }
void main()
{
  int i,j,s;
```

```
printf("Dijikstras algorithm");
printf("Enter the number of vertices:");
scanf("%d",&n);
printf("Enter the cost matrix:\n");
  for(i=1;i<=n;i++)
  {
    for(j=1;j<=n;j++)
     {
       scanf("%d",&distance[i][j]);
       }
       }
       printf("\nEnter the source vertex");
       scanf("%d",&s);
       printf("\nThe cost matrix is\n");
       for(i=1;i<=n;i++)
       {
         for(i=1;i<=n;i++)
         {
           printf("%d\t",distance[i][j]);
           }
           printf("\n");
           }
           shortpath(s);
           printf("\nThe shortest path from the source vertex %d is:\n",s);
           printpath(s);


           }
```