

```

1  #include<stdio.h>
2  #define infinity 999
3  void dij(int n, int v, int cost[20][20], int dist[]){
4  int i,u,count,w,flag[20],min;
5  for(i=1;i<=n;i++)
6  flag[i]=0, dist[i]=cost[v][i]; count=2;
7  while(count<=n){
8  min=999; for(w=1;w<=n;w++)
9  if(dist[w]<min && !flag[w]) {
10     min=dist[w];
11     u=w;
12 }
13 flag[u]=1;
14 count++;
15 for(w=1;w<=n;w++)
16 if((dist[u]+cost[u][w]<dist[w]) && !flag[w]) dist[w]=dist[u]+cost[u][w];
17 }
18 }
19 int main(){
20 int n,v,i,j,cost[20][20],dist[20];
21 printf("enter the number of nodes:");
22 scanf("%d",&n);
23 printf("\n enter the cost matrix:\n");
24 for(i=1;i<=n;i++)
25 for(j=1;j<=n;j++){
26 scanf("%d",&cost[i][j]); if(cost[i][j] == 0)
27 cost[i][j]=infinity;
28 }
29 printf("\n enter the source matrix:");
30 scanf("%d",&v);
31 dij(n,v,cost,dist);
32 printf("\n shortest path : \n");
33 for(i=1;i<=n;i++)
34 if(i!=v)
35 printf("%d->%d,cost=%d\n",v,i,dist[i]);
36 }
37
38

```