***Code:***

#include<bits/stdc++.h>

using namespace std ;

int cost[100][100] , vertices ;

int getMin(int dist[] , bool visited[]){

int key = 0 ;

int min = INT\_MAX ;

for(int i=0;i < vertices ; i++){

if(!visited[i] && dist[i]<min){

min = dist[i] ;

key = i ;

}

}

return key ;

}

void display(int dist[] , int par[] ){

for(int i =0 ;i < vertices ;i++){

int temp = par[i] ;

cout<<i << " <- " ;

while(temp!=-1)

{

cout<< temp << " <- " ;

temp = par[temp] ;

}

cout<<endl;

cout<<"::::Distance = " << dist[i] ;

cout<<endl;

}

}

void dijkstra(int src ){

int par[100] , dist[100] ;

bool visited[100] ={0} ;

fill(dist , dist+vertices , INT\_MAX ) ;

dist[src] =0 ;

par[src] =-1 ;

for(int g = 0 ;g<vertices-1 ; g++){

int u = getMin( dist , visited ) ;

visited[u] = true ;

cout<< " min = " << u <<endl;

for(int v =0 ; v< vertices ;v++){

if(!visited[v] && (dist[u]+cost[u][v]) < dist[v] && cost[u][v]!=9999)

{

par[v] = u ;

dist[v] = dist[u] + cost[u][v] ;

}

}

}

display(dist , par) ;

}

int main() {

cout<<"Enter the number of vertices: " ;

cin>>vertices ;

cout<<"Enter cost matrix : \n" ;

for(int i = 0 ;i < vertices ; i++){

for(int j = 0 ; j< vertices ; j++){

cin>>cost[i][j] ;

}

}

int start ;

cout<<"Enter start : " ;

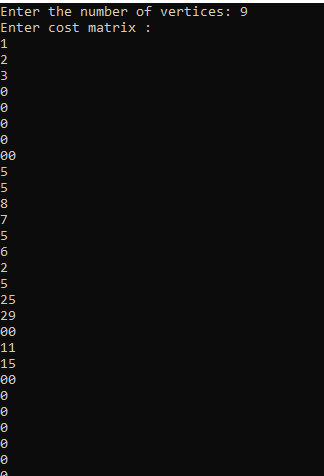
cin>>start ;

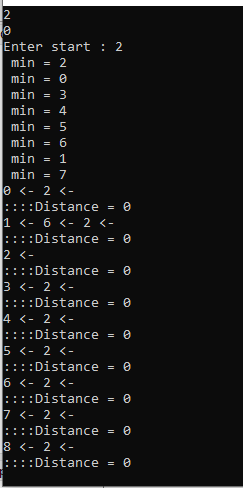
dijkstra(start) ;

return 0;

}

**Output:**

****

****