DIJKSTRAS ALGORITHM

```
#include <bits/stdc++.h>
#include <limits.h>
#include <stdio.h>
using namespace std;
#define V 4
int mindist(int dist[], bool sptset[])
{
int min = INT_MAX, min_index;
for (int v=0; v<V; v++)
if (SptSet[v] == false && dist[v] <= min)
min = dist[v], min_index = v;
return min_index;
void printSolutions (int dist[])
{
printf("Vertex");
for (int i=0; i<V; i++)
printf("%d \t %d \n", i, dist[i]);
}
void dijkstra (int graph[V][V], int src)
{
int dist[V];
bool sptset[V];
for (int i = 0; i<V; i++)
  dist[i] = INT_MAX, sptSet[i] = false;
  dist[src]=0;
for (int count=0; count<V-1; count++) {
  int u= mindist(dist, sptset);
  sptSet[u] = true;
  for (int v=0; v<V; v++)
```

```
if (!sptSet[v] && graph[u][v] && dist[v] !=
  INT_MAX && dist[u]+ graph[u][v]< dist[v])
  dist[v] dist[u]+graph[u][v];
}
printSol(dist);
}
int main()
{
int graph[V][V];
cout<<"Enter the graph"<<endl;
for(int i=0;i<V;i++)
{  for(j=0;j<V;j++)
     cin>>graph[i][j];
}
dijkstra(graph,0);
}
```

Output:- Enter the graph.

```
0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0
```

vertex	Distance from Source
0	0
1	8
2	2
3	5

```
Enter no. of vertices:4

Enter the adjacency matrix:
0 5 9999 9999
2 0 4 9999
9999 9999 0 6
4 7 5 0

Enter the starting node:0

Distance of node1=5
Path=1<-0
Distance of node2=9
Path=2<-1<-0
Distance of node3=15
Path=3<-2<-1<-0
```