

Dijkstra's Algorithm

```
# include <bits/stdc++.h>
using namespace std;

# define V9

int mindist(int dist[], bool sptset[])
{
    int min = 9999, 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 printPath(int parent[], int j)
```

12/9/20


```

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[u] + graph[u][v]
            < dist[v])
        {
            parent[v] = u;
            dist[v] = dist[u] + graph[u][v];
        }
    }
    printSol(dist, v, parent);
}

int main()
{
    int graph[V][V];
    cout << "Enter the graph: " << endl;

```


J.S. Meghana

IBM18CS039

```
for (int i=0; i<V; i++)  
{  
    for (int j=0; j<V; j++)  
        cin >> graph[i][j];  
}  
cout << "Enter the source: " << endl;  
int src;  
dijkstra (graph, src);  
return 0;  
}
```