

```

1  #include <bits/stdc++.h>
2
3  #define read freopen("input.txt","r",stdin)
4  #define MAX 99999
5  #define MIN -1
6
7  using namespace std;
8
9  int min_value_arr[MAX];
10 int cost[MAX][3];
11 int dis[MAX];
12 int P[MAX];
13 int v, e, s;
14
15 void Initialize_Single_Source()
16 {
17     for(int i=1; i<=v; i++)
18     {
19         dis[i] = MAX;
20         P[i] = MIN;
21     }
22     dis[s] = 0;
23 }
24 int Weight(int u, int v)
25 {
26     for(int i=1; i<=e; i++)
27     {
28         if(cost[i][0] == u && cost[i][1] == v)
29         {
30             return cost[i][2];
31         }
32     }
33 }
34 void Relax(int u, int v)
35 {
36     if (dis[v] > dis[u] + Weight(u,v))
37     {
38         dis[v] = dis[u] + Weight(u,v);
39         P[v] = u;
40     }
41 }
42 int Extract_Min()
43 {
44     int c = MAX, k;
45     for(int i=1; i<=v; i++)
46         if(dis[i] < c && !min_value_arr[i])
47         {
48             c = dis[i];
49             k = i;
50         }
51     min_value_arr[k] = 1;
52     return k;
53 }
54 void Dijkstra()
55 {
56     Initialize_Single_Source();
57     for(int i=1; i<=v; i++)
58     {
59         int m = Extract_Min();
60         for(int j=1; j<=e; j++)
61         {
62             if (cost[j][0] == m)
63                 Relax(m, cost[j][1]);
64         }
65     }
66 }

```

```
67
68 int main()
69 {
70     cin >> v >> e >> s;
71     for (int i=1; i<=e; i++)
72     {
73         cin >> cost[i][0] >> cost[i][1] >> cost[i][2];
74     }
75     Dijkstra();
76     cout << endl << "Distance of the vertices : " << endl;
77     for(int i=1; i<=v ; i++)
78     {
79         cout << dis[i] << "\t";
80     }
81     return 0;
82 }
```