```
1 #include <bits/stdc++.h>
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++)</pre>
18
19
         dis[i] = MAX;
20
          P[i] = MIN;
21
22
       dis[s] = 0;
23 }
24 int Weight(int u, int v)
25 {
       for(int i=1; i<=e; i++)</pre>
26
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
           dis[v] = dis[u] + Weight(u,v);
38
           P[v] = u;
39
40
41 }
42 int Extract_Min()
43 {
44
       int c = MAX , k;
45
       for(int i=1; i<=v; i++)</pre>
46
          if(dis[i] < c && !min_value_arr[i])</pre>
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();
       for(int i=1; i<=v; i++)</pre>
57
58
59
           int m = Extract_Min();
60
           for(int j=1; j<=e; j++)</pre>
61
62
               if (cost[j][0] == m)
63
                  Relax(m,cost[j][1]);
64
65
66 }
```

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