

```

1  /*****
2
3      Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to comp
6
7  *****/
8  #include <stdio.h>
9  #define INFINITY 9999
10 #define MAX 10
11
12 void Dijkstra(int Graph[MAX][MAX], int n, int start);
13
14 void Dijkstra(int Graph[MAX][MAX], int n, int start) {
15     int cost[MAX][MAX], distance[MAX], pred[MAX];
16     int visited[MAX], count, mindistance, nextnode, i, j;
17
18     for (i = 0; i < n; i++)
19         for (j = 0; j < n; j++)
20             if (Graph[i][j] == 0)
21                 cost[i][j] = INFINITY;
22             else
23                 cost[i][j] = Graph[i][j];
24
25     for (i = 0; i < n; i++) {
26         distance[i] = cost[start][i];
27         pred[i] = start;
28         visited[i] = 0;
29     }
30
31     distance[start] = 0;
32     visited[start] = 1;
33     count = 1;
34
35     while (count < n - 1) {
36         mindistance = INFINITY;
37
38

```

input



```

main.c
38
39 for (i = 0; i < n; i++)
40     if (distance[i] < mindistance && !visited[i]) {
41         mindistance = distance[i];
42         nextnode = i;
43     }
44
45     visited[nextnode] = 1;
46     for (i = 0; i < n; i++)
47         if (!visited[i])
48             if (mindistance + cost[nextnode][i] < distance[i]) {
49                 distance[i] = mindistance + cost[nextnode][i];
50                 pred[i] = nextnode;
51             }
52     count++;
53 }
54
55
56 for (i = 0; i < n; i++)
57     if (i != start) {
58         printf("\nDistance from source to %d: %d", i, distance[i]);
59     }
60 }
61 int main() {
62     int Graph[MAX][MAX], n, u;
63     n = 7;
64
65     Graph[0][0] = 0;
66     Graph[0][1] = 0;
67     Graph[0][2] = 1;
68     Graph[0][3] = 2;
69     Graph[0][4] = 0;
70     Graph[0][5] = 0;
71     Graph[0][6] = 0;
72
73     Graph[1][0] = 0;
74     Graph[1][1] = 0;
75     Graph[1][2] = 2;

```

main.c

```

62 int Graph[MAX][MAX], n, u;
63 n = 7;
64
65 Graph[0][0] = 0;
66 Graph[0][1] = 0;
67 Graph[0][2] = 1;
68 Graph[0][3] = 2;
69 Graph[0][4] = 0;
70 Graph[0][5] = 0;
71 Graph[0][6] = 0;
72
73 Graph[1][0] = 0;
74 Graph[1][1] = 0;
75 Graph[1][2] = 2;
76 Graph[1][3] = 0;
77 Graph[1][4] = 0;
78 Graph[1][5] = 3;
79 Graph[1][6] = 0;
80
81 Graph[2][0] = 1;
82 Graph[2][1] = 2;
83 Graph[2][2] = 0;
84 Graph[2][3] = 1;
85 Graph[2][4] = 3;
86 Graph[2][5] = 0;
87 Graph[2][6] = 0;
88
89 Graph[3][0] = 2;
90 Graph[3][1] = 0;
91 Graph[3][2] = 1;
92 Graph[3][3] = 0;
93 Graph[3][4] = 0;
94 Graph[3][5] = 0;
95 Graph[3][6] = 1;
96
97 Graph[4][0] = 0;
98 Graph[4][1] = 0;
99 Graph[4][2] = 3;

```

I



```

88
89     Graph[3][0] = 2;
90     Graph[3][1] = 0;
91     Graph[3][2] = 1;
92     Graph[3][3] = 0;
93     Graph[3][4] = 0;
94     Graph[3][5] = 0;
95     Graph[3][6] = 1;
96
97     Graph[4][0] = 0;
98     Graph[4][1] = 0;
99     Graph[4][2] = 3;
100    Graph[4][3] = 0;
101    Graph[4][4] = 0;
102    Graph[4][5] = 2;
103    Graph[4][6] = 0;
104
105    Graph[5][0] = 0;
106    Graph[5][1] = 3;
107    Graph[5][2] = 0;
108    Graph[5][3] = 0;
109    Graph[5][4] = 2;
110    Graph[5][5] = 0;
111    Graph[5][6] = 1;
112
113    Graph[6][0] = 0;
114    Graph[6][1] = 0;
115    Graph[6][2] = 0;
116    Graph[6][3] = 1;
117    Graph[6][4] = 0;
118    Graph[6][5] = 1;
119    Graph[6][6] = 0;
120
121    u = 0;
122    Dijkstra(Graph, n, u);
123
124    return 0;
125 }

```