

## Practical-6

**Name:** Abhijeet Vidwan Vyavhare

**Roll No:** 232

**PRN:** 202202040012

### Problem Statement:

Write a program to implement Distance Vector Routing / Link State Routing.

### Code:

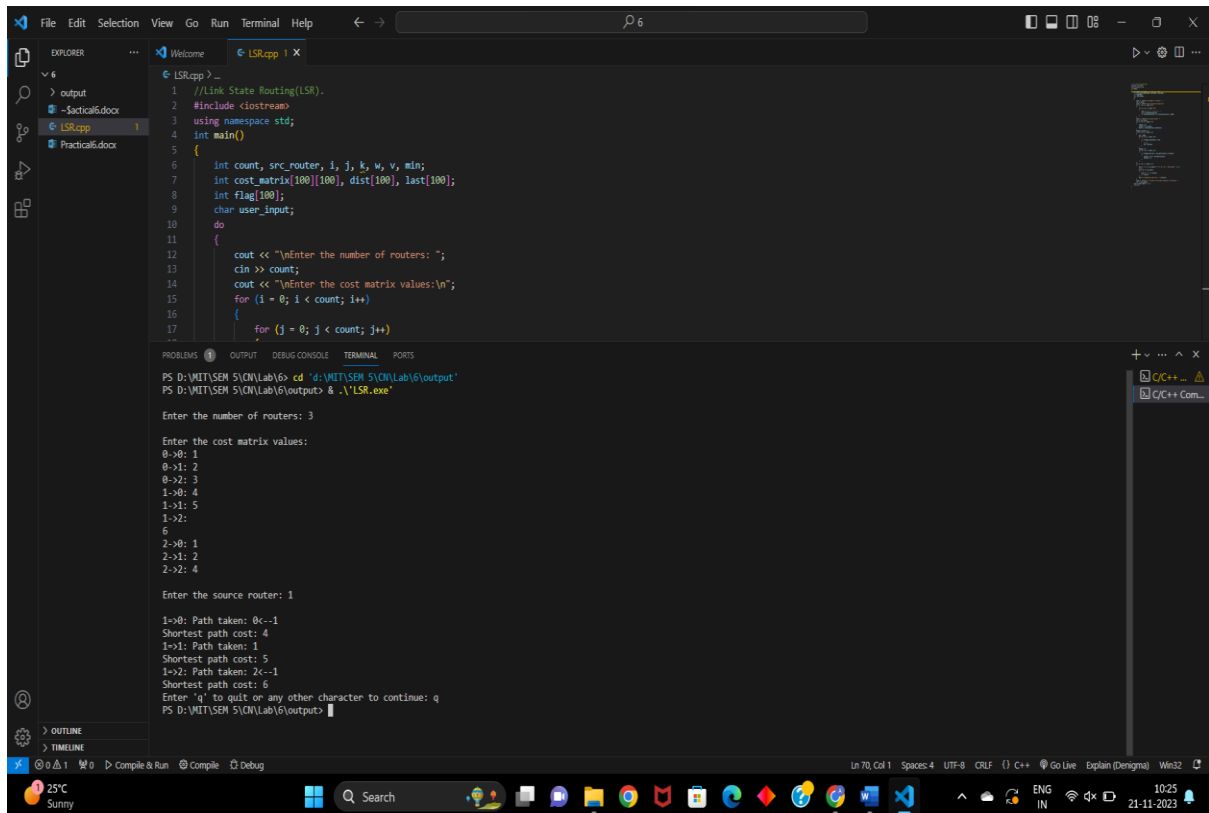
```
//Link State Routing(LSR).
#include <iostream>
using namespace std;
int main()
{
    int count, src_router, i, j, k, w, v, min;
    int cost_matrix[100][100], dist[100], last[100];
    int flag[100];
    char user_input;
    do
    {
        cout << "\nEnter the number of routers: ";
        cin >> count;
        cout << "\nEnter the cost matrix values:\n";
        for (i = 0; i < count; i++)
        {
            for (j = 0; j < count; j++)
            {
                cout << i << "->" << j << ": ";
                cin >> cost_matrix[i][j];
                if (cost_matrix[i][j] < 0) cost_matrix[i][j] = 1000;
            }
        }
        cout << "\nEnter the source router: ";
        cin >> src_router;
        for (v = 0; v < count; v++)
        {
            flag[v] = 0;
            last[v] = src_router;
            dist[v] = cost_matrix[src_router][v];
        }
        flag[src_router] = 1;
        for (i = 0; i < count; i++)
        {
            min = 1000;
            for (w = 0; w < count; w++)
            {
                if (!flag[w] && dist[w] < min)
```

```

        {
            v = w;
            min = dist[w];
        }
    }
    flag[v] = 1;
    for (w = 0; w < count; w++)
    {
        if (!flag[w] && (min + cost_matrix[v][w] < dist[w]))
        {
            dist[w] = min + cost_matrix[v][w];
            last[w] = v;
        }
    }
}
for (i = 0; i < count; i++)
{
    cout << "\n" << src_router << "=>" << i << ": Path taken: " << i;
    w = i;
    while (w != src_router)
    {
        cout << "<--" << last[w];
        w = last[w];
    }
    cout << "\nShortest path cost: " << dist[i];
}
cout << "\nEnter 'q' to quit or any other character to continue: ";
cin >> user_input;
}while (user_input != 'q');
return 0;
}

```

## OUTPUT:



```
1 //Link State Routing(LSR).
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     int count, src_router, i, j, k, w, v, min;
7     int cost_matrix[100][100], dist[100], last[100];
8     int flag[100];
9     char user_input;
10    do
11    {
12        cout << "\nEnter the number of routers: ";
13        cin >> count;
14        cout << "\nEnter the cost matrix values:\n";
15        for (i = 0; i < count; i++)
16        {
17            for (j = 0; j < count; j++)
18            {
19                if (i == j)
20                {
21                    cost_matrix[i][j] = 0;
22                }
23                else
24                {
25                    cost_matrix[i][j] = 100;
26                }
27            }
28        }
29        cout << "\nEnter the source router: ";
30        cin >> src_router;
31        for (i = 0; i < count; i++)
32        {
33            dist[i] = 100;
34            last[i] = -1;
35            flag[i] = 0;
36        }
37        dist[src_router] = 0;
38        while (1)
39        {
40            min = 100;
41            for (i = 0; i < count; i++)
42            {
43                if (flag[i] == 0)
44                {
45                    for (j = 0; j < count; j++)
46                    {
47                        if (flag[j] == 1)
48                        {
49                            if (dist[j] + cost_matrix[j][i] < min)
50                            {
51                                min = dist[j] + cost_matrix[j][i];
52                                last[i] = j;
53                            }
54                        }
55                    }
56                }
57            }
58            flag[i] = 1;
59        }
60        cout << "\nShortest path cost: ";
61        for (i = 0; i < count; i++)
62        {
63            cout << last[i] << " ";
64        }
65        cout << "\n";
66        user_input = getch();
67        if (user_input == 'q')
68            break;
69    }
70    return 0;
71 }
```

PS D:\MIT\SEM 5\CN\Lab\6> cd "d:\MIT\SEM 5\CN\Lab\6\output"

PS D:\MIT\SEM 5\CN\Lab\6\output> .\LSR.exe

Enter the number of routers: 3

Enter the cost matrix values:

0->0: 1  
0->1: 2  
0->2: 3  
1->0: 4  
1->1: 5  
1->2: 6  
2->0: 1  
2->1: 2  
2->2: 4

Enter the source router: 1

1->0: Path taken: 0<-1  
Shortest path cost: 4  
1->1: Path taken: 1  
Shortest path cost: 5  
1->2: Path taken: 2<-1  
Shortest path cost: 6  
Enter 'q' to quit or any other character to continue: q  
PS D:\MIT\SEM 5\CN\Lab\6\output>