

linkstate.c

```
#include<stdio.h>
```

```
void shortest_path(int n , int cost[n][n] , int src)
{
    int dist[n];
    int visited[n];
    int i;
    int last[n];
    int count;

    for (i = 0; i < n ; i++)
    {
        dist[i] = 1000;
        visited[i] = 0;
        last[i] = src;
    }

    dist[src] = 0;

    for (count = 0 ; count < n-1 ; count++)
    {
        int min = 1000;
        int u;
        for (i = 0 ; i < n ; i++)
        {
            if (visited[i] == 0 && dist[i]<= min)
            {
                min = dist[i];
                u = i;
            }
        }

        visited[u] = 1;

        for (i = 0;i < n ; i++)
        {
            if (visited[i] == 0 && dist[u] + cost[u][i] < dist[i])
            {
                dist[i] = dist[u] + cost[u][i];

                if(last[i] == src)
                {
                    last[i] = u;
                }
            }
        }
    }

    printf(" Routing Table of Node %d \n" , src + 1);
```

```

printf("Destination\tCost\tNext Hop \n");

for(i = 0 ; i < n ; i++)
{
    if(i == src - 1)
    {
        printf(" %d\t\t - \t\t - \n", src + 1);
    }
    else
    {
        if(last[i] == src)
        {
            printf(" %d\t\t%d\t\t-\n", i + 1, dist[i]);
        }
        else
            printf(" %d\t\t%d\t\t%d\n", i + 1, dist[i] , last[i] + 1);
    }
}

printf("\n");

for(i = 0;i < n ; i++)
{
    if(i != src)
    {
        printf(" The cost of the shortest path from router %d to %d is %d\n", src + 1 ,
i + 1 , dist[i]);
    }
}

}

int main()
{
    int n;
    int i;
    int j;
    int src;
    printf("Enter the Number of Nodes : ");

    scanf("%d",&n);

    int cost[n][n];

    printf(" Enter the cost between Nodes : \n");

    for(i = 0; i < n ; i++)
    {
        for(j = 0 ;j < n ; j++)
        {
            if(i != j)
            {
                printf("Cost from %d->%d : ",i + 1 ,j + 1);

```

```

scanf("%d",&cost[i][j]);

if(cost[i][j] == 0)
{
    cost[i][j] = 1000;
}
else
{
    cost[i][j] = 0;
}
}
}

printf(" Enter the source Node : ");

scanf("%d",&src);

printf("Routing Table of Node %d\n",src);

printf("Destination\tCost\tNext Hop\n");

for(i = 0; i < n ; i++)
{
    if(i == src - 1)
    {
        printf(" %d\t\t\t\t\n",src);
    }
    else
    {
        if(cost[src-1][i] == 0)
        {
            printf(" %d\t\t\t\t\n",i+1);
        }
        else
        {
            printf(" %d\t\t%d\t\t\n",i+1,cost[src-1][i]);
        }
    }
}

printf("After Applying Dijkstra's Algorithm\n\n");

shortest_path(n , cost , src-1);

return 0;
}

```

Output

```

gcc linkstate.c
net@inlab:~$ ./a.out

```

Enter the Number of Nodes : 3

Enter the cost between Nodes :

Cost from 1->2 : 2

Cost from 1->3 : 3

Cost from 2->1 : 4

Cost from 2->3 : 2

Cost from 3->1 : 2

Cost from 3->2 : 4

Enter the source Node : 1

Routing Table of Node 1

Destination	Cost	Next Hop
1	-	-
2	2	-
3	3	-

After Applying Dijkstra's Algorithm

Routing Table of Node 1

Destination	Cost	Next Hop
1	0	-
2	2	-
3	3	-

The cost of the shortest path from router 1 to 2 is 2

The cost of the shortest path from router 1 to 3 is 3

net@inlab:~\$