

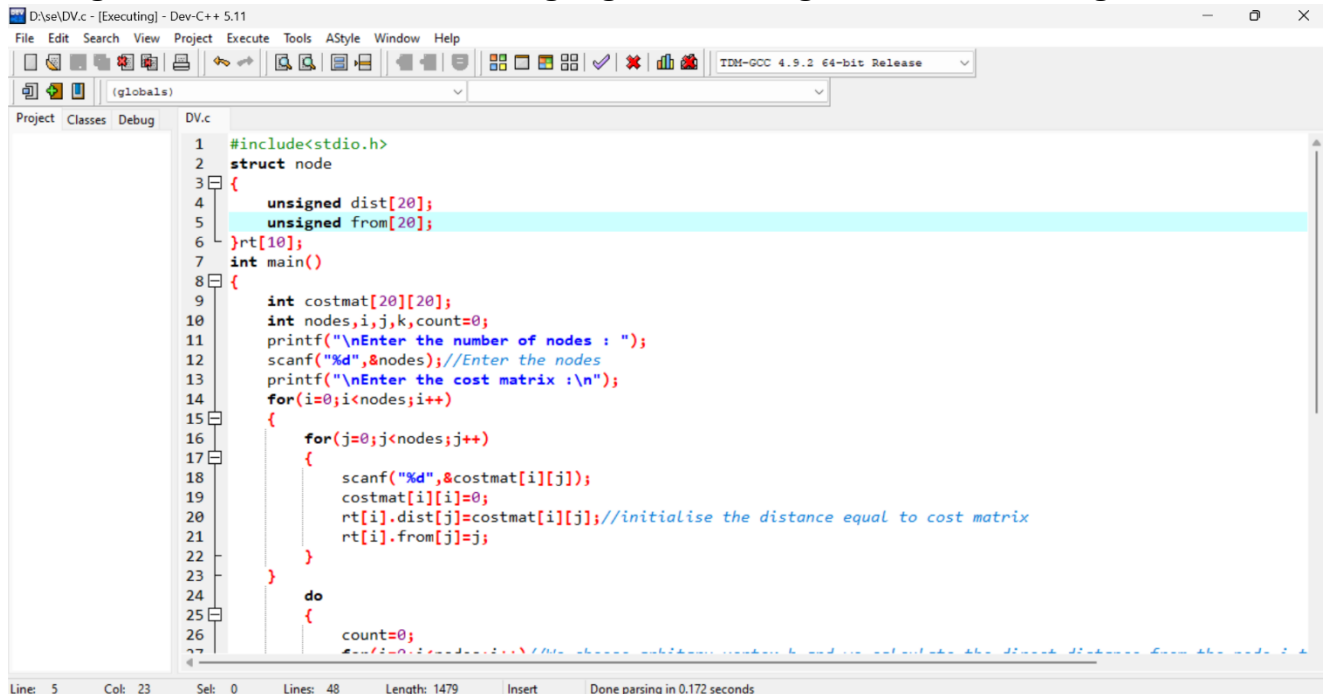
Date: 02/10/2023

## Lab Practical #11:

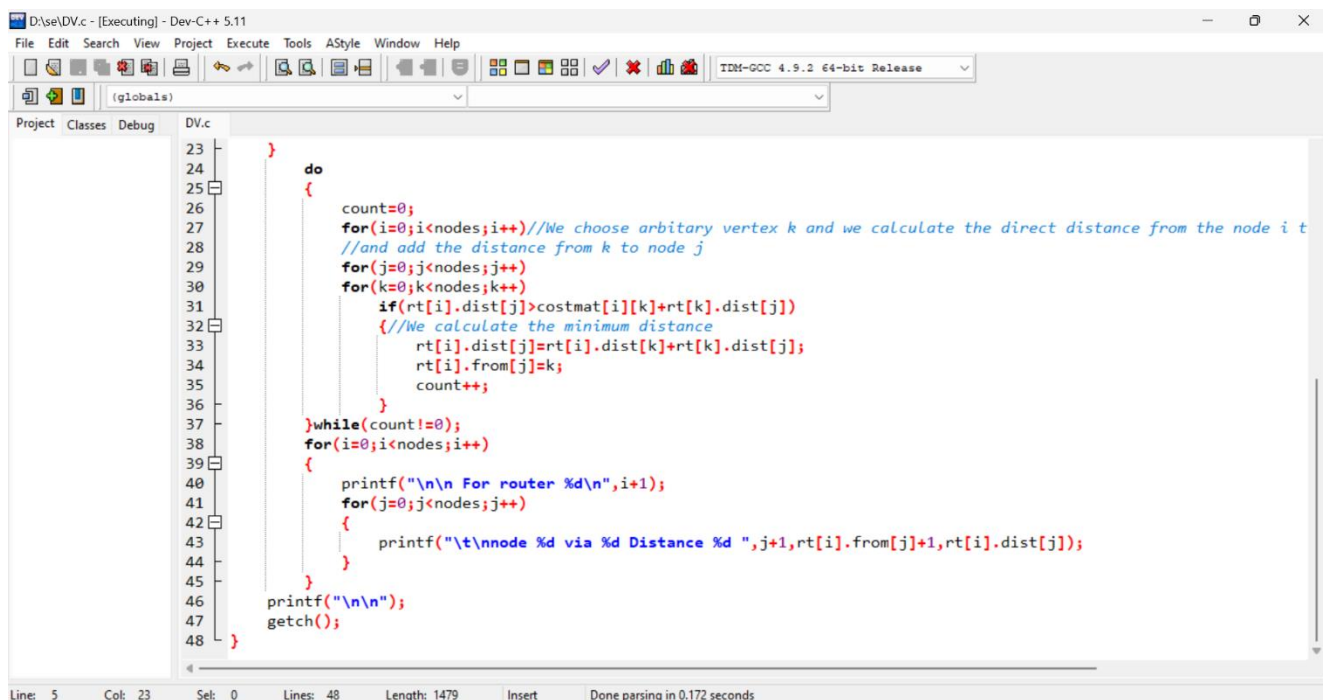
To develop network using distance vector routing protocol and link state routing protocol.

## Practical Assignment #11:

### 1. C Program: Distance Vector Routing Algorithm using Bellman Ford's Algorithm.



```
1 #include<stdio.h>
2 struct node
3 {
4     unsigned dist[20];
5     unsigned from[20];
6 }rt[10];
7 int main()
8 {
9     int costmat[20][20];
10    int nodes,i,j,k,count=0;
11    printf("\nEnter the number of nodes : ");
12    scanf("%d",&nodes); //Enter the nodes
13    printf("\nEnter the cost matrix :\n");
14    for(i=0;i<nodes;i++)
15    {
16        for(j=0;j<nodes;j++)
17        {
18            scanf("%d",&costmat[i][j]);
19            costmat[i][i]=0;
20            rt[i].dist[j]=costmat[i][j]; //initialise the distance equal to cost matrix
21            rt[i].from[j]=j;
22        }
23    }
24    do
25    {
26        count=0;
```



```
23 }
24 do
25 {
26     count=0;
27     for(i=0;i<nodes;i++) //We choose arbitrary vertex k and we calculate the direct distance from the node i to
28     //and add the distance from k to node j
29     for(j=0;j<nodes;j++)
30     for(k=0;k<nodes;k++)
31     if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
32     { //We calculate the minimum distance
33         rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
34         rt[i].from[j]=k;
35         count++;
36     }
37 }while(count!=0);
38 for(i=0;i<nodes;i++)
39 {
40     printf("\n\n For router %d\n",i+1);
41     for(j=0;j<nodes;j++)
42     {
43         printf("\tnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
44     }
45 }
46 printf("\n\n");
47 getch();
48 }
```

Date: 02/10/2023

### Output:

```
D:\se\DV.exe
Enter the number of nodes : 3

Enter the cost matrix :
0 2 7
2 0 1
7 1 0

For router 1
node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 2 Distance 3

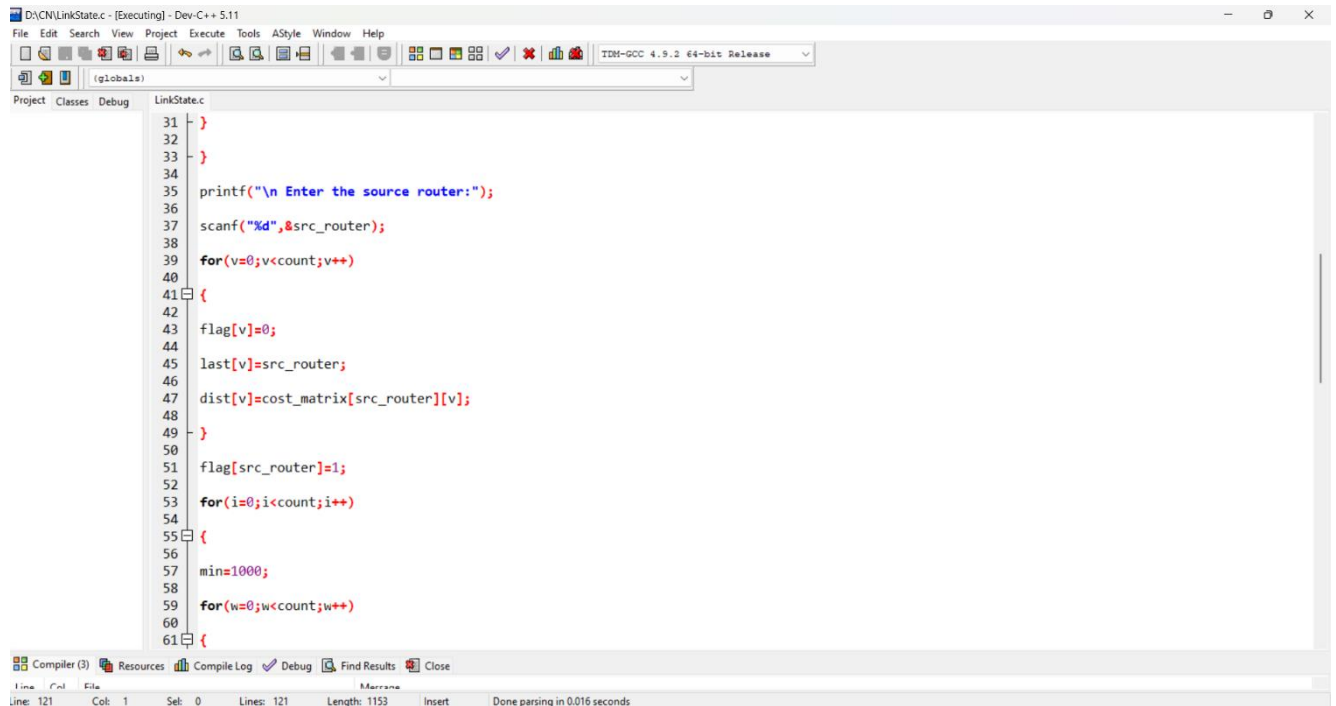
For router 2
node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1

For router 3
node 1 via 2 Distance 3
node 2 via 2 Distance 1
node 3 via 3 Distance 0
```

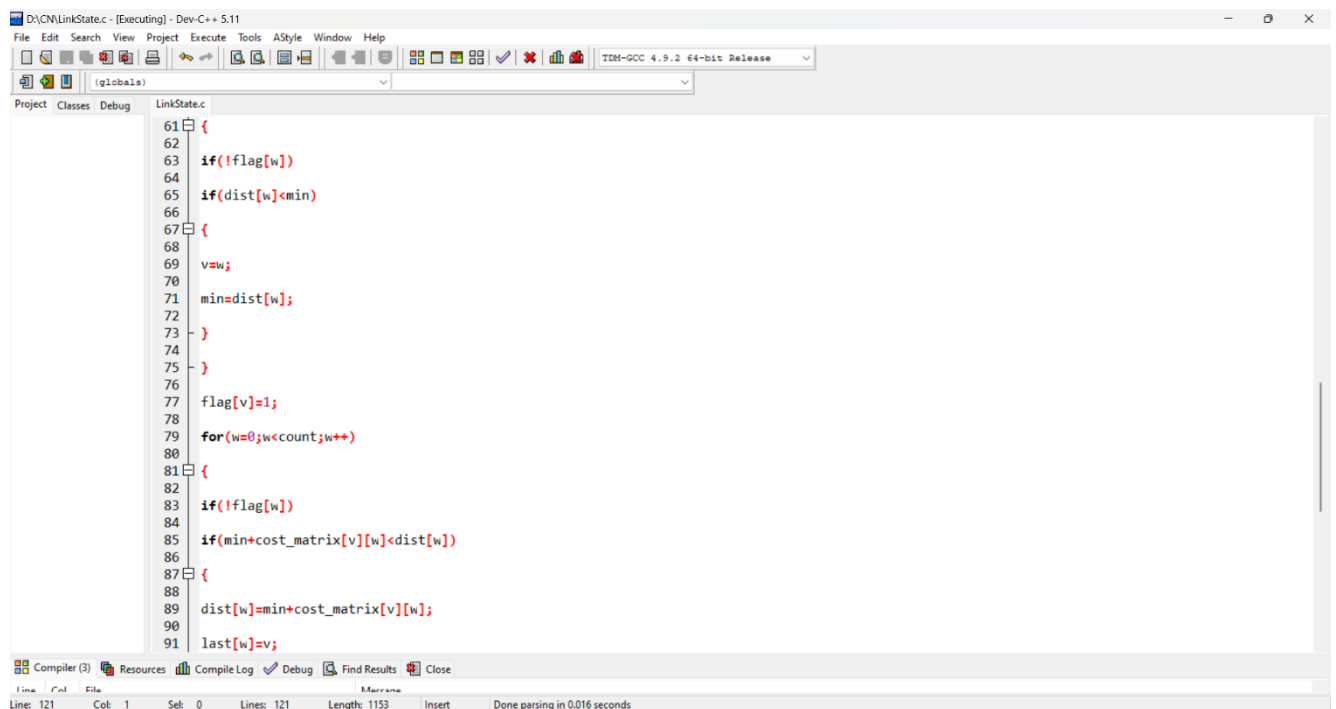
## 2. C Program: Link state routing algorithm.

```
D:\CN\LinkState.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug LinkState.c
1 int main()
2
3 {
4
5 int count,src_router,i,j,k,w,v,min;
6
7 int cost_matrix[100][100],dist[100],last[100];
8
9 int flag[100];
10
11 printf("\n Enter the no of routers");
12
13 scanf("%d",&count);
14
15 printf("\n Enter the cost matrix values:");
16
17 for(i=0;i<count;i++)
18 {
19 {
20
21 for(j=0;j<count;j++)
22 {
23 {
24
25 printf("\n%d->%d:",i,j);
26
27 scanf("%d",&cost_matrix[i][j]);
28
29 if(cost_matrix[i][j]<0)cost_matrix[i][j]=1000;
30
31 }
```

Date: 02/10/2023

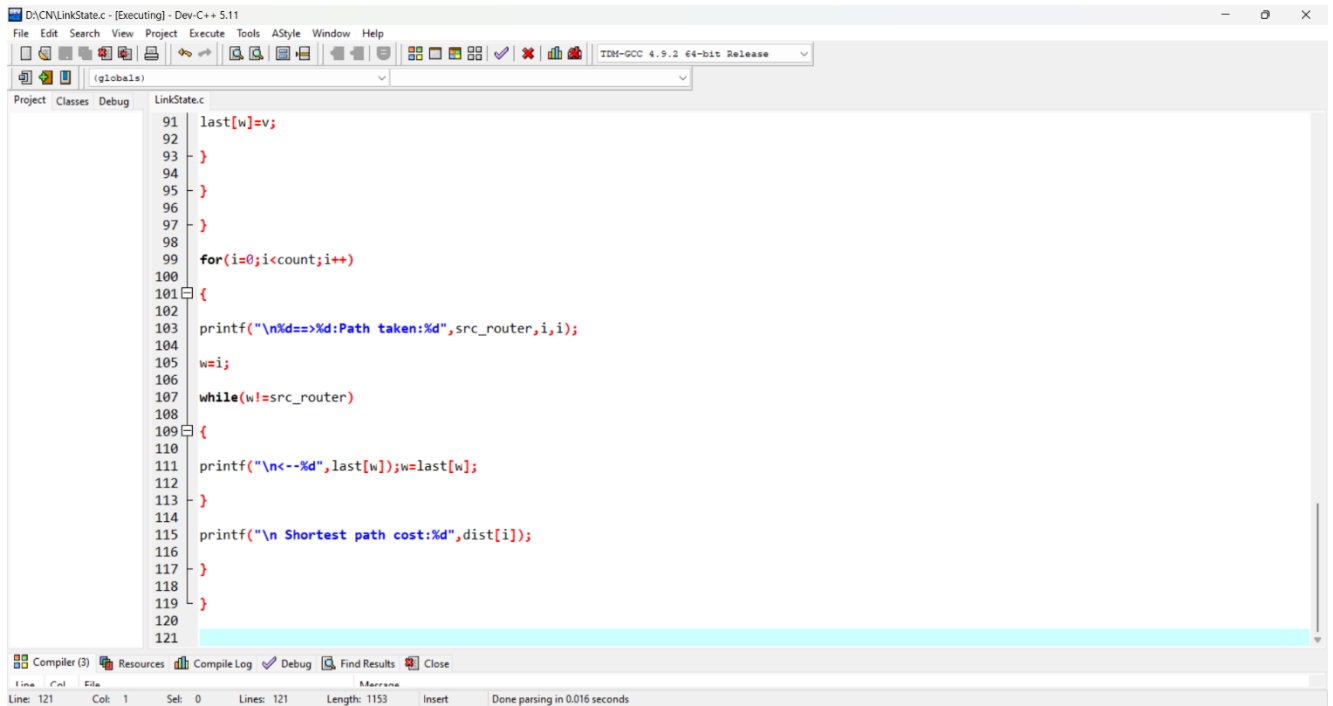


```
31 }
32 }
33 }
34
35 printf("\n Enter the source router:");
36
37 scanf("%d",&src_router);
38
39 for(v=0;v<count;v++)
40 {
41     flag[v]=0;
42     last[v]=src_router;
43     dist[v]=cost_matrix[src_router][v];
44 }
45
46 flag[src_router]=1;
47
48 for(i=0;i<count;i++)
49 {
50     min=1000;
51     for(w=0;w<count;w++)
52     {
```



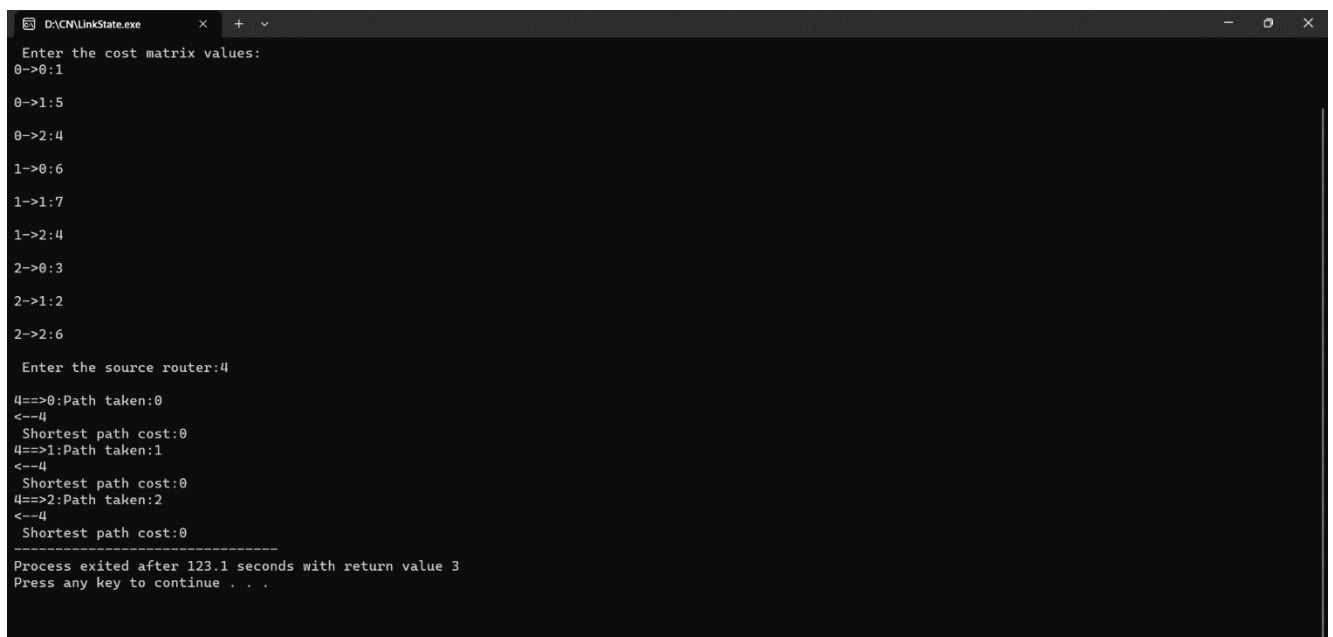
```
61 {
62     if(!flag[w])
63     {
64         if(dist[w]<min)
65         {
66             v=w;
67             min=dist[w];
68         }
69     }
70     flag[v]=1;
71     for(w=0;w<count;w++)
72     {
73         if(!flag[w])
74         {
75             if(min+cost_matrix[v][w]<dist[w])
76             {
77                 dist[w]=min+cost_matrix[v][w];
78                 last[w]=v;
79             }
80         }
81     }
82 }
```

Date: 02/10/2023



```
91 last[w]=v;  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 for(i=0;i<count;i++)  
100 {  
101 {  
102 }  
103 printf("\n%d==>%d:Path taken:%d",src_router,i,i);  
104 }  
105 w=i;  
106 }  
107 while(w!=src_router)  
108 {  
109 {  
110 }  
111 printf("\n<--%d",last[w]);w=last[w];  
112 }  
113 }  
114 }  
115 printf("\n Shortest path cost:%d",dist[i]);  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }
```

### Output :



```
D:\CN\LinkState.exe  
Enter the cost matrix values:  
0->0:1  
0->1:5  
0->2:4  
1->0:6  
1->1:7  
1->2:4  
2->0:3  
2->1:2  
2->2:6  
  
Enter the source router:4  
4==>0:Path taken:0  
<--4  
Shortest path cost:0  
4==>1:Path taken:1  
<--4  
Shortest path cost:0  
4==>2:Path taken:2  
<--4  
Shortest path cost:0  
-----  
Process exited after 123.1 seconds with return value 3  
Press any key to continue . . .
```