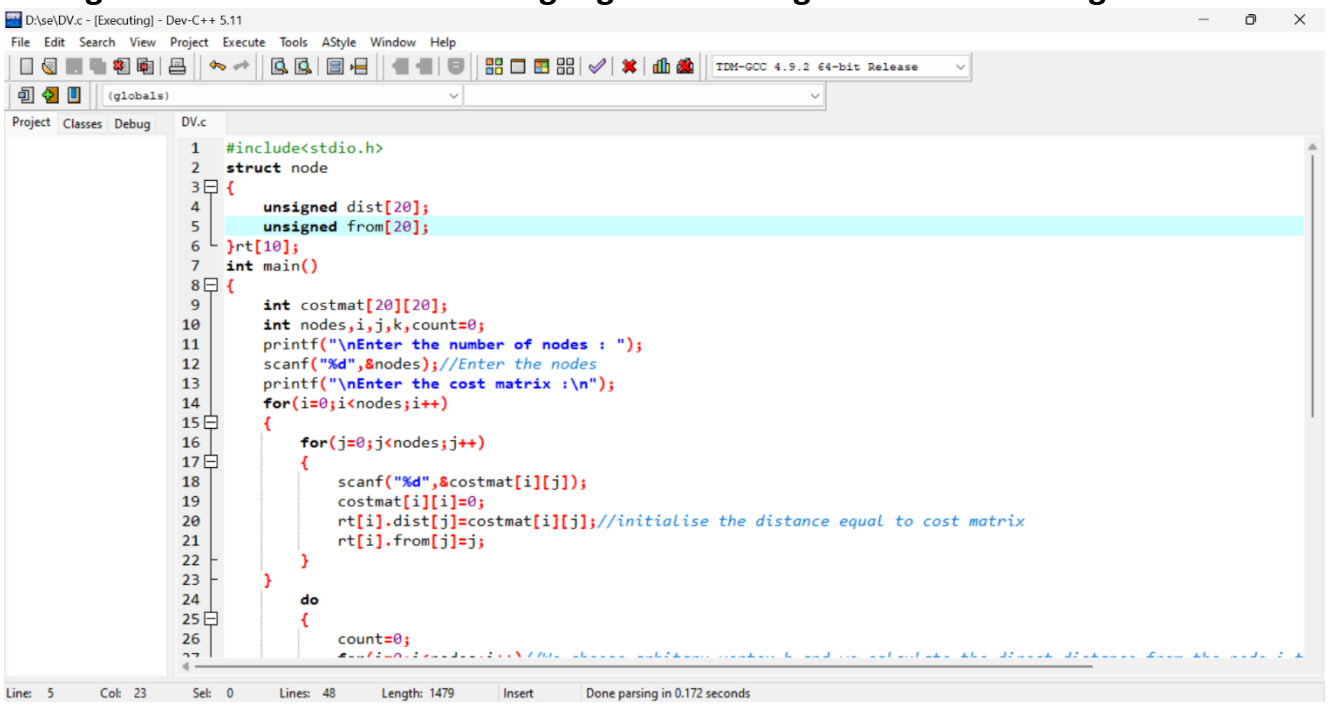
Date: 02/10/2023

Lab Practical #11:

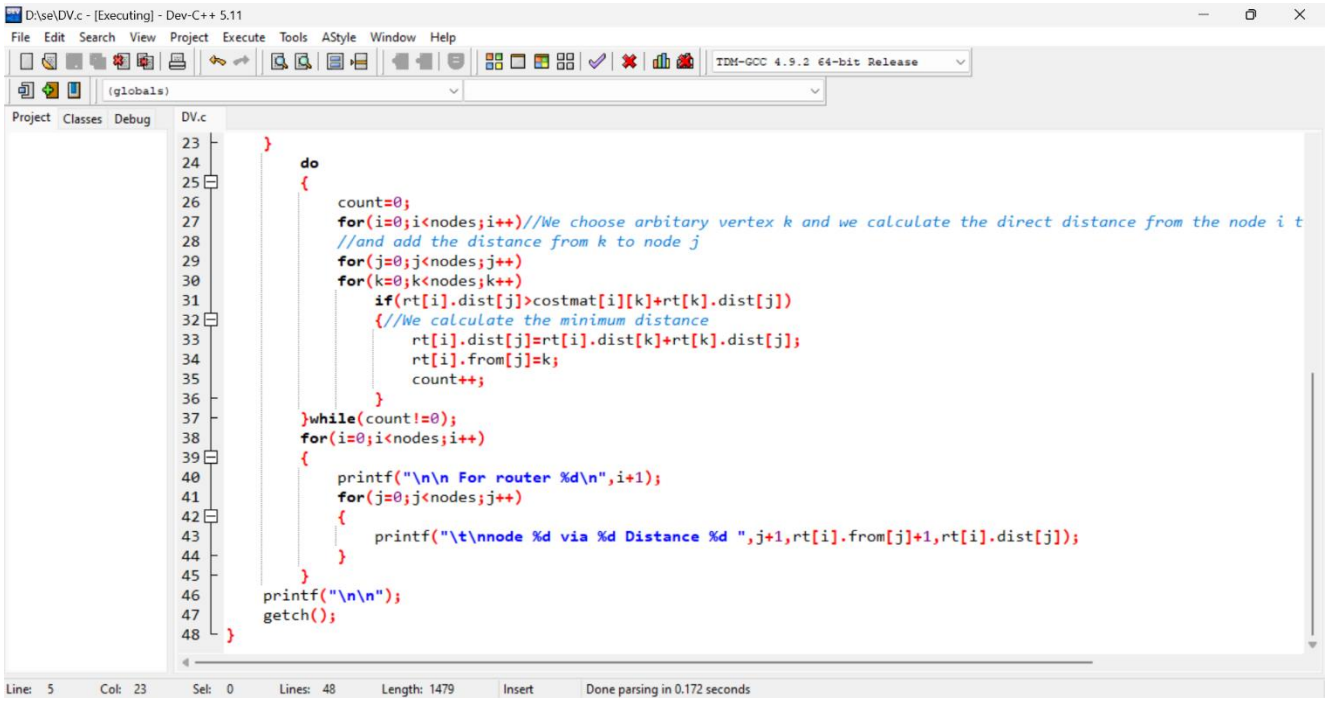
To develop network using distance vector routing protocol and link state routing protocol.

Practical Assignment #11:

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



```
#include<stdio.h>
struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];
int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);//Enter the nodes
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost matrix
            rt[i].from[j]=j;
        }
    }
        do
        {
            count=0;
```



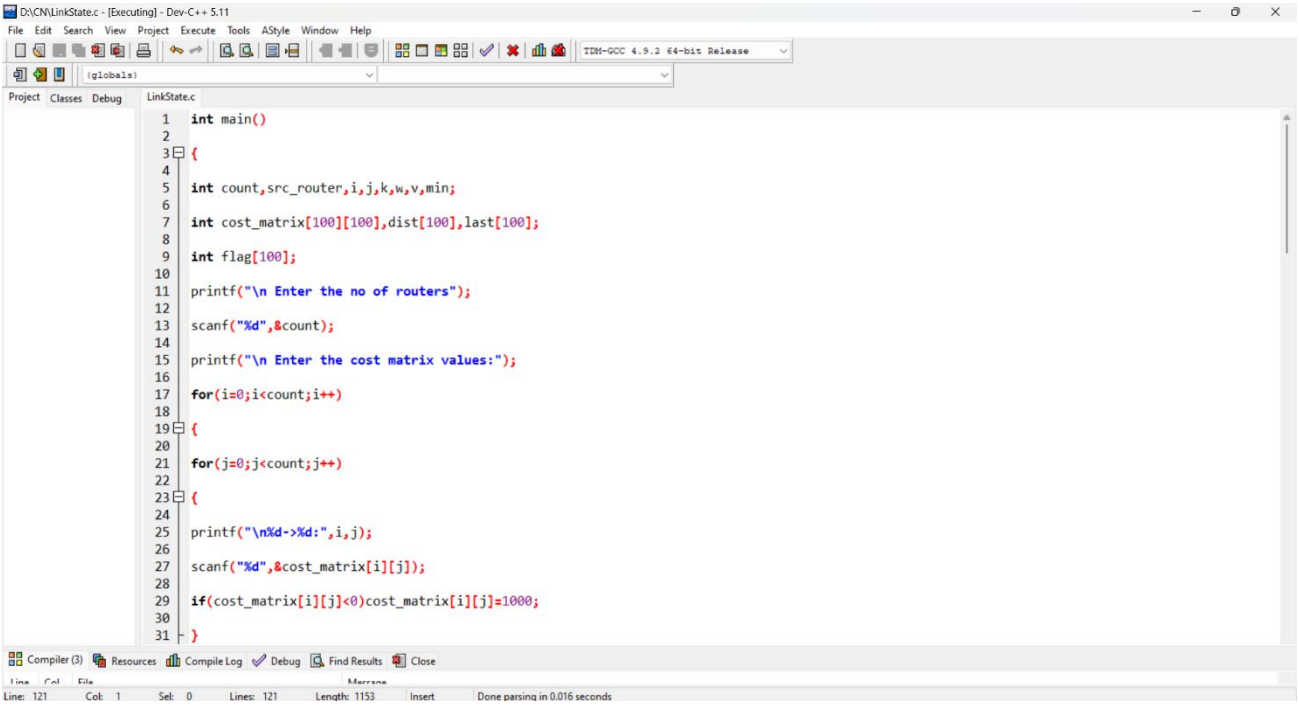
```
    }
        do
        {
            count=0;
            for(i=0;i<nodes;i++)//We choose arbitary vertex k and we calculate the direct distance from the node i t
            //and add the distance from k to node j
            for(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
                if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                {//We calculate the minimum distance
                    rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                    rt[i].from[j]=k;
                    count++;
                }
        }while(count!=0);
        for(i=0;i<nodes;i++)
        {
            printf("\n\n For router %d\n",i+1);
            for(j=0;j<nodes;j++)
            {
                printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
        }
    printf("\n\n");
    getch();
}
```

Date: 02/10/2023

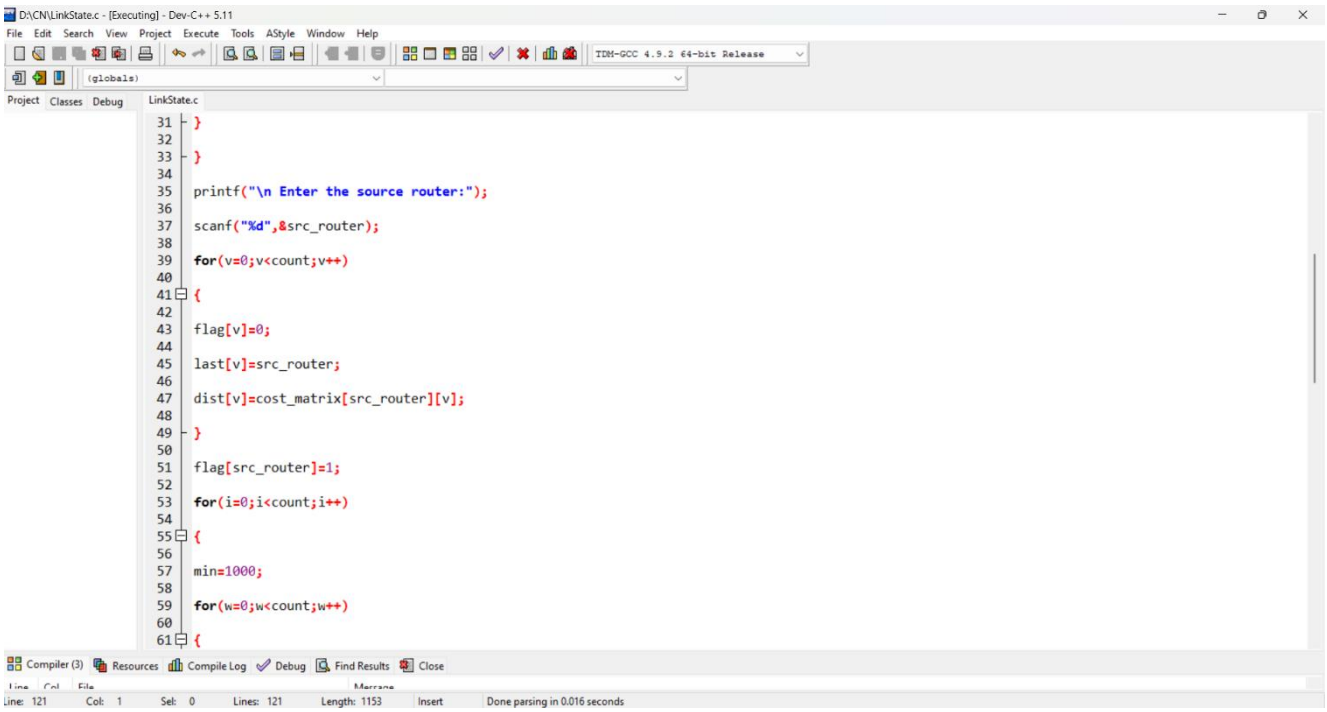
Output:

```
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
```

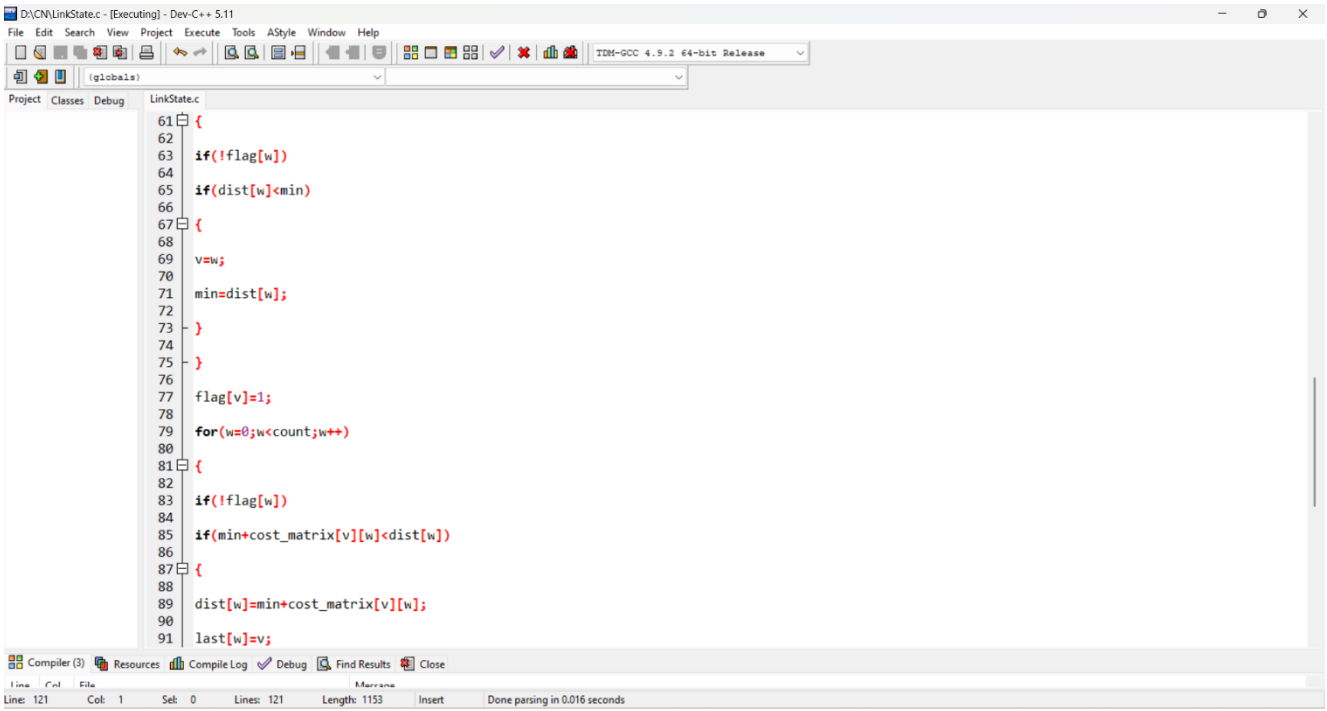
2. C Program: Link state routing algorithm.



Date: 02/10/2023

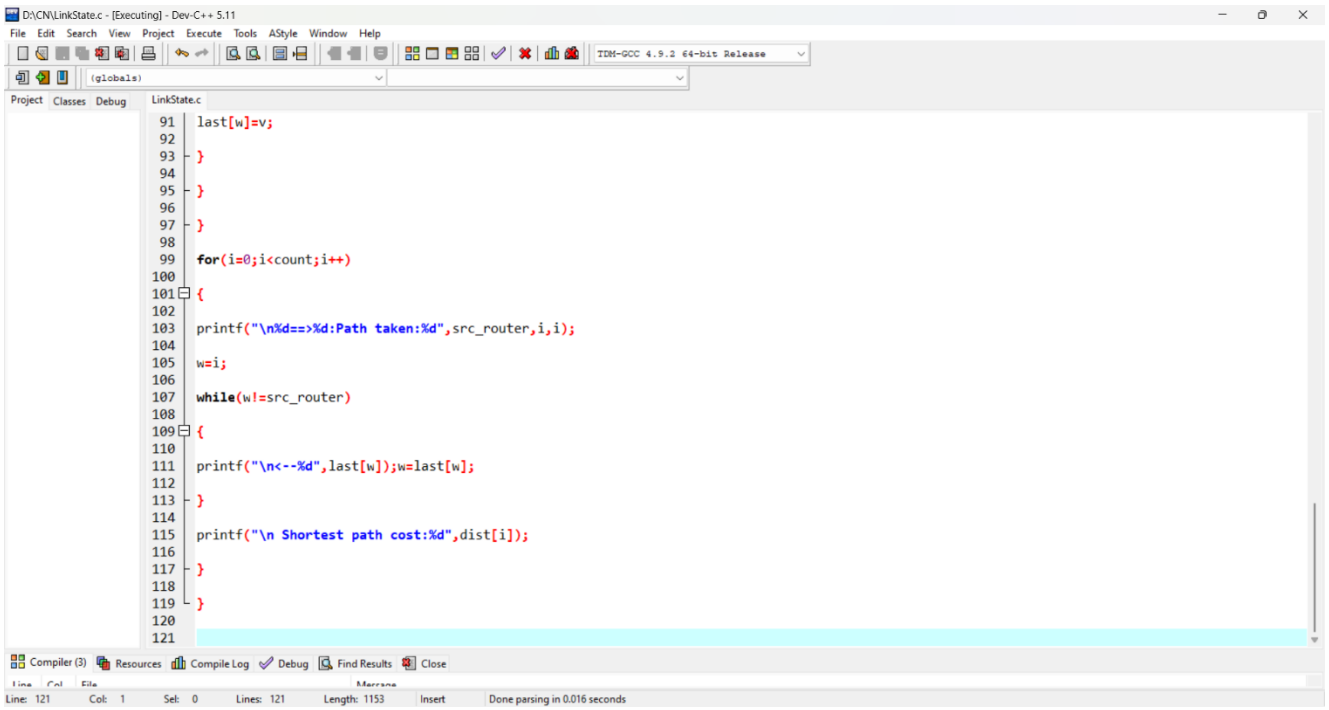


```
31  }
32
33  }
34
35  printf("\n Enter the source router:");
36
37  scanf("%d",&src_router);
38
39  for(v=0;v<count;v++)
40
41  {
42
43  flag[v]=0;
44
45  last[v]=src_router;
46
47  dist[v]=cost_matrix[src_router][v];
48
49  }
50
51  flag[src_router]=1;
52
53  for(i=0;i<count;i++)
54
55  {
56
57  min=1000;
58
59  for(w=0;w<count;w++)
60
61  {
```



```
61  {
62
63  if(!flag[w])
64
65  if(dist[w]<min)
66
67  {
68
69  v=w;
70
71  min=dist[w];
72
73  }
74
75  }
76
77  flag[v]=1;
78
79  for(w=0;w<count;w++)
80
81  {
82
83  if(!flag[w])
84
85  if(min+cost_matrix[v][w]<dist[w])
86
87  {
88
89  dist[w]=min+cost_matrix[v][w];
90
91  last[w]=v;
```

Date: 02/10/2023



Output :