

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  #define initial 0
5  #define visited 1
6  #define MAX 100
7
8  int n;
9  int adj[MAX][MAX];
10 int state[MAX];
11 int stack[MAX];
12 int top=-1;
13
14 void DF_traversal();
15 void DFS(int v);
16
17 void push(int v);
18 void create_graph();
19 int pop();
20 int is_empty_stack();
21
22 void create_graph()
23 {
24     int i,max_edges,origin,destin;
25     printf("Enter the number of nodes:");
26     scanf("%d",&n);
27     for(i=0;i<n;i++)
28     {
29         for(int j=0;j<n;j++)
30         {
31             adj[i][j]=0;
32         }
33     }
34     max_edges=n*(n-1);
35     for(i=1;i<=max_edges;i++)
36     {
37         printf("\nEnter edge %d (-1 -1) to quit:",i);
38         scanf("%d %d",&origin,&destin);
39         if(origin==--1&&destin==--1)
40             break;
41         if(origin>=n||destin>=n||origin<0||destin<0)
42         {
43             printf("\nInvalid Edge");
44             i--;
45         }
46         else
47         {
48             adj[origin][destin]=1;
49         }
50     }
51 }
52
53 void DF_traversal()
54 {
55     int v;
56     for(v=0;v<n;v++){
57         state[v]=initial;
58     }
59     printf("\nEnter starting node depth first search : ");
60     scanf("%d",&v);
61     DFS(v);
62     printf("\n");
63 }
64
65 void DFS(int v){
66     int i;

```

```

67     push(v);
68     while(!is_empty_stack()){
69         v=pop();
70         if (state[v]==initial){
71             printf("%d ", v);
72             state[v] = visited;
73         }
74         for (i=n-1;i>=0;i--) {
75             if (adj[v][i]==1&&state[i]==initial)
76                 push(i);
77         }
78     }
79 }
80
81
82 void push(int v)
83 {
84     if(top==MAX-1){
85         printf("\nStack is Full");
86     }
87     else{
88         top=top+1;
89         stack[top]=v;
90     }
91 }
92
93 int pop()
94 {
95     int v;
96     if(top==--1){
97         printf("\nstack is empty");
98         exit(0);
99     }
100    else{
101        v=stack[top];
102        top=top-1;
103        return v;
104    }
105 }
106
107 int is_empty_stack()
108 {
109     if(top==--1)
110     {
111         return 1;
112     }
113     else
114     {
115         return 0;
116     }
117 }
118 int main()
119 {
120     create_graph();
121     DF_traversal();
122     return 0;
123 }

```