Run Debug Stop Share Save {} Beautify Language C

main.c

```c
#include<stdio.h>
#include<stdlib.h>
int A[20][20],visited[20],count=0,n;
int seq[20],connected=1,acyclic=1;
void DFS();
void DFSearch(int cur);
int main(){
    int i,j;
    printf("Enter no of Vertices: ");
    scanf("%d",&n);
    printf("\nEnter the Adjacency Matrix-\n");
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            scanf("%d",&A[i][j]);
    printf("\nThe matrix is-\n");
    for(i=1;i<=n;i++){
        for(j=1;j<=n;j++){
            printf("%d\t",A[i][j]);
        }
        printf("\n");
    }
    printf("\nThe Depth First Search Traversal:\n");
    DFS();
    for(i=1;i<=n;i++)
        printf("%c,%d\t",'a'+seq[i]-1,i);
    if(connected && acyclic)    printf("\n\nIt is a Connected, Acyclic Graph!");
    if(!connected && acyclic)   printf("\n\nIt is a Not-Connected, Acyclic Graph!");
    if(connected && !acyclic)   printf("\n\nGraph is a Connected, Cyclic Graph!");
    if(!connected && !acyclic)  printf("\n\nIt is a Not-Connected, Cyclic Graph!");
```

Run Debug Stop Share Save {} Beautify Language

main.c

```
29      printf("\nThe Depth First Search Traversal:\n");
30      DFS();
31      for(i=1;i<=n;i++)
32          printf("%c,%d\t",'a'+seq[i]-1,i);
33      if(connected && acyclic)    printf("\n\nIt is a Connected, Acyclic Graph!");
34      if(!connected && acyclic)   printf("\n\nIt is a Not-Connected, Acyclic Graph!");
35      if(connected && !acyclic)   printf("\n\nGraph is a Connected, Cyclic Graph!");
36      if(!connected && !acyclic)  printf("\n\nIt is a Not-Connected, Cyclic Graph!");
37      printf("\n\n");
38      return 0;
39  }
40  void DFS(){
41      int i;
42      for(i=1;i<=n;i++)
43          if(!visited[i]){
44          if(i>1) connected=0;
45          DFSearch(i);
46                }
47  }
48  void DFSearch(int cur){
49      int i,j;
50      visited[cur]=++count;
51          seq[count]=cur;
52          for(i=1;i<count-1;i++)
53                  if(A[cur][seq[i]])
54                     acyclic=0;
55      for(i=1;i<=n;i++)
56          if(A[cur][i] && !visited[i])
57             DFSearch(i);
58  }
```

Input

```
Enter no of Vertices: 4

Enter the Adjacency Matrix-
0 1 1 0
1 0 0 1
1 0 0 1
0 1 1 0

The matrix is-
0       1       1       0
1       0       0       1
1       0       0       1
0       1       1       0

The Depth First Search Traversal:
a,1     b,2     d,3     c,4

Graph is a Connected, Cyclic Graph!


...Program finished with exit code 0
Press ENTER to exit console.
```