

Lab Program 1:

Write program to do the following:

- a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
- b. Check whether a given graph is connected or not using DFS method

Code:

```
#include<stdio.h>

int a[10][10] , vis[10] , q[10], n, f = 0 , r = 0;

void dfs(int);
void bfs(int);

int main()
{
    printf("\nEnter the number of vertices for DFS: ");
    scanf("%d",&n);
    printf("\nEnter the Adjacency Matrix for DFS\n");
    for(int i = 0 ; i < n ; i++){
        for(int j = 0 ; j < n ; j++){
            {
                scanf("%d",&a[i][j]);
            }
            vis[i] = 0;
        }
        printf("\nGraph in DFS: ");
        dfs(0);

        printf("\nGraph in BFS: ");
        bfs(0);

        return 0;
    }

    void dfs(int i){
```

```

printf("%d ",i+1);
vis[i]= 1;
for(int j = 0; j < n ; j ++){
    if(a[i][j] == 1 && vis[j] == 0){
        dfs(j);
    }
}
}
}

```

```

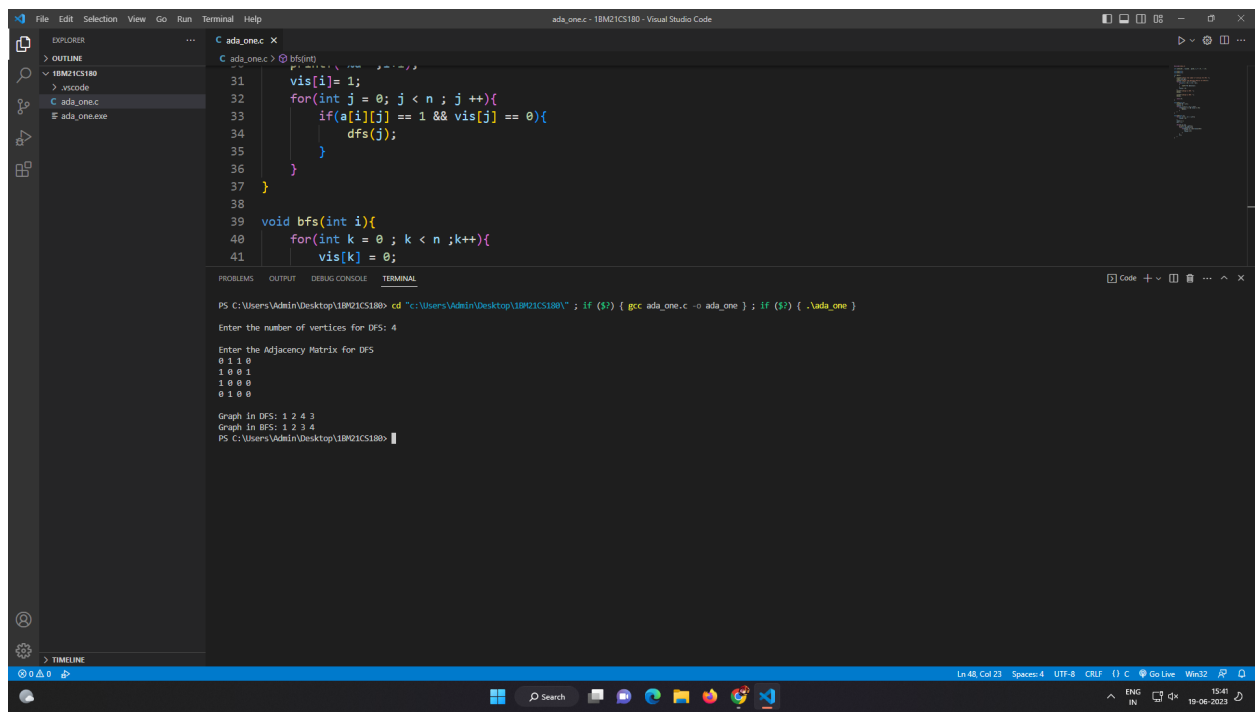
void bfs(int i){
    for(int k = 0 ; k < n ;k++){
        vis[k] = 0;
    }
    vis[i] = 1;
    q[r] = i;

```

```

while(f <= r){
    printf("%d ",q[f]+1);
    for(int j=0;j<n;j++){
        if(a[f][j] == 1 && vis[j]==0){
            q[++r] = j;
            vis[j] = 1;
        }
    }
    f++;
}
}

```



```

ada_one.c
31 vis[i]= 1;
32 for(int j = 0; j < n ; j ++){
33     if(a[i][j] == 1 && vis[j] == 0){
34         dfs(j);
35     }
36 }
37 }
38
39 void bfs(int i){
40     for(int k = 0 ; k < n ;k++){
41         vis[k] = 0;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\Adm\Desktop\18P21CS180> cd "C:\Users\Adm\Desktop\18P21CS180\" ; if ($?) { gcc ada_one.c -o ada_one ; if ($?) { .\ada_one }
Enter the number of vertices for DFS: 4
Enter the Adjacency Matrix for DFS
0 1 1 0
1 0 0 1
1 0 0 0
0 1 0 0
Graph in DFS: 1 2 4 3
Graph in BFS: 1 2 3 4
PS C:\Users\Adm\Desktop\18P21CS180>

```