Data Structure Lab

Assignment No-11

Name : Sanket Shivaji Jadhav.

Prn: 2020BTECS00005

1. Implementation of Graph ,BFS and DFS.

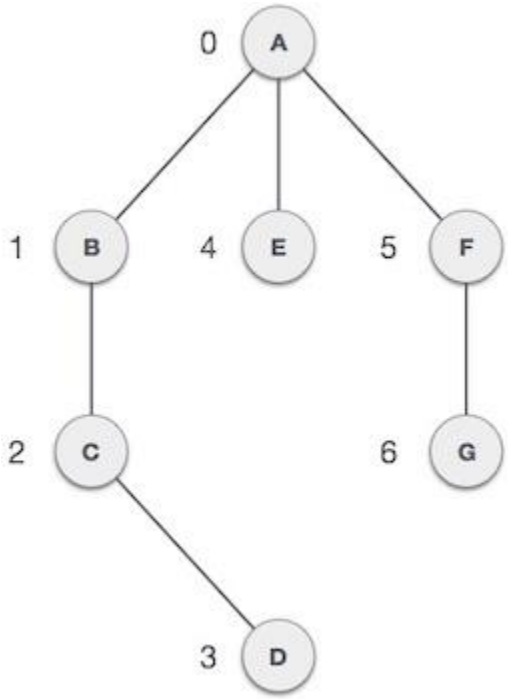
Code:

```
    // Graph creation
#include<bits/stdc++.h>
using namespace std;

void addEdge(vector<int> adj[], int u, int v)
{
    adj[u].push_back(v);
    adj[v].push_back(u);
}


void BFS(int s,int V,vector<int>adj[])
{
    bool *visited = new bool[V];
    for(int i = 0; i < V; i++)
        visited[i] = false;

   list<int> queue;

    visited[s] = true;
    queue.push_back(s);

    while(!queue.empty())
    {
```

```
        s = queue.front();
        cout << s << " ";
        queue.pop_front();

        for (auto i = adj[s].begin(); i != adj[s].end();
++i)
        {
            if (!visited[*i])
            {
                visited[*i] = true;
                queue.push_back(*i);
            }
        }
    }
}


bool *visited = new bool[5];
void DFS(int v,vector<int>adj[])
{
    visited[v] = true;
    cout << v << " ";

    for (auto i = adj[v].begin(); i != adj[v].end(); ++i)
        if (!visited[*i])
            DFS(*i,adj);
}



int main()
{//non directred graph
    int V = 5;
    //Number of edges of the graph
```

```
    vector<int> adj[V];
    addEdge(adj, 0, 1);
    addEdge(adj, 0, 4);
    addEdge(adj, 1, 2);
    addEdge(adj, 1, 3);
    addEdge(adj, 1, 4);
    addEdge(adj, 2, 3);
    addEdge(adj, 3, 4);
    BFS(2,V,adj);
    cout<<endl;
    DFS(2,adj);
    return 0;
}
```

INPUT:



OUTPUT:


Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\sai\Desktop\dsa> cd "c:\Users\sai\Desktop\dsa\" ; if ($?) { g++ EX_11_graph.cp
p -o EX_11_graph } ; if ($?) { .\EX_11_graph }
BFS is : 2 1 3 0 4
DFS is : 2 1 0 4 3
PS C:\Users\sai\Desktop\dsa>