LAB ASSESSMENT (GRAPH IMPLEMENTATION)

Name- Arya Dubey

Registration Number-20BCE0908

Faculty-Professor Gopinath M.P.

CODE:

```
//implementation of depth first search
#include<iostream>
using  namespace std;
#include<conio.h>
/*class dfs represents a directed graph using adjacency list representation*/
class dfs

{
      private:
      int a[10][10];
      int n,*visit;
      public:
      dfs();
      void read();
      void searchfrom(int k);
      void print();
};
dfs::dfs()
{
      cout<<"Depth first Search"<<endl;
      cout<<"Enter the no of node:";
      cin>>n;
```

```
        visit=new int[n];
        for(int i=0;i<=n;i++)
        {
                visit[i]=0;
                for(int j=0;j<=n;j++)
                {
                        a[i][j]=0;
                }
        }
}
void dfs::read() // to read values
{
        for(int i=1;i<n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        if(i!=j)
                        {
                        cout<<"\n Enter the valuesof:"<<i<<","<<j<<"->";
                        cin>>a[i][j]; // inputing values from user
                        }
                }
        }
}
void dfs::print()
{
        cout<<"\n Nodes are visited inthe order:"<<endl;
        for(int i=1;i<=n;i++)
        {
                if(visit[i]==0)
                {
```

```
            searchfrom(i);
            }
        }
}
void dfs::searchfrom(int k) //dfs traversal
{
                cout<<k<<"->";//prints the current node
        visit[k]=1; // marks the current node as visited


        /*repeat the process for all the vertices
            adjacent to the current*/
        for(int i=1;i<=n;i++)
        {
            if(visit[i]==0 && a[k][i]!=0)
            {
                searchfrom(i);
            }
        }
}
int main()
{

        dfs d1; // creates a graph
        d1.read();// function
        d1.print();//call through d1

        return 0;
}
```

OUTPUT:

```
Depth first Search
Enter the no of node: 4

 Enter the valuesof:1,2-> 1

 Enter the valuesof:1,3-> 0

 Enter the valuesof:1,4-> 1

 Enter the valuesof:2,1-> 1

 Enter the valuesof:2,3-> 1

 Enter the valuesof:2,4-> 0

 Enter the valuesof:3,1-> 0

 Enter the valuesof:3,2-> 1

 Enter the valuesof:3,4-> 1

 Nodes are visited inthe order:
1->2->3->4->
```