Program No:	**3**
Roll No :	**1525**
Title of Program :	
Objective :	Adjacency matrix

SOURCE CODE:

```
import java.util.*;
public class DFT
{

    private int[] [] adj;   //Adjacency matrix for the graph
    private boolean[] visited;      //Vector to track visited nodes
    private int[] stack;
    private int tos;

    //constructor
    public DFT(int v)
    {
        adj=new int[v][v];
        visited=new boolean[v];
        stack=new int[v];
        tos=-1;

    }//end of dft

    //add edge
    public void addEdge(int src,int dest)
    {
        adj[src][dest]=1;
        adj[dest][src]=1;
    }//end of addedge

    public void performDFT(int x)
    {
        push(x);

        System.out.println("Depth First Traversal: ");

        while(tos != -1)
        {
            int curr=pop();
            if(!visited[curr])
            {
                visited[curr]=true;
```

```
                System.out.print(curr + " "); // Changed println to print for better
formatting

                for(int i=0; i<adj.length; i++) // Changed loop to go from 0 to
adj.length
                {
                    if(adj[curr][i]==1 && !visited[i])
                    {
                        push(i);
                    }

                }//end of for
            }//end of if
        }//end of while
        System.out.println();
    }//end of performdft

    private void push(int node)
    {
        tos++;
        stack[tos]=node;
    }//end of push

    private int pop()
    {
        int tmp=stack[tos];
        tos--;

        return tmp;
    }//end of pop

    //Main
    public static void main(String[] args)
    {
        DFT g=new DFT(5);

        //Add Edges
        g.addEdge(0,1);
        g.addEdge(0,2);
        g.addEdge(0,3);
        g.addEdge(1,3);
        g.addEdge(2,4);
        g.addEdge(3,4);

        g.performDFT(0);        //DFT from node 0
    }
```

```
}//end of DFT
```

OUTPUT: