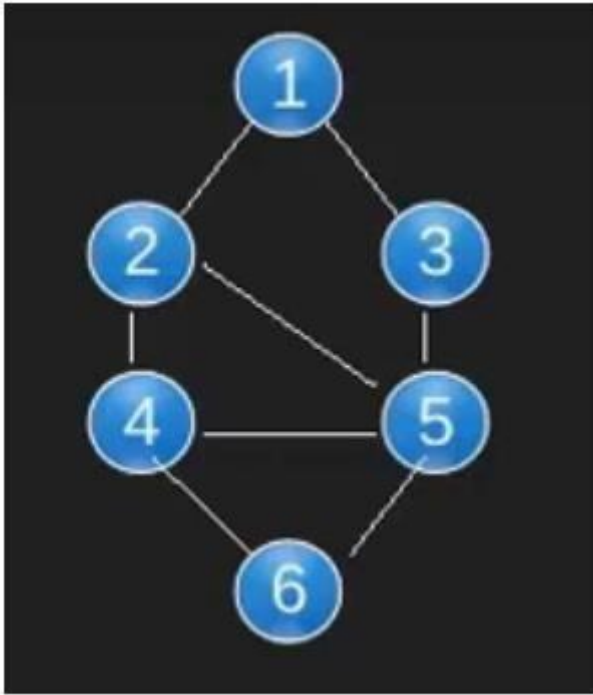


TITLE: Graphs.

AIM: To find the breadth first search transversal using java program:



PROGRAM:

```
import java.util.*;
public class Main
{
    private Queue<Node> queue;          static
    ArrayList<Node> nodes=new ArrayList<Node>(); static
    class Node
    {
        int data;
        boolean visited;                List<Node>
        neighbours;

        Node(int data)
        {
            this.data=data;
            this.neighbours=new ArrayList<>();
        }
        public void addneighbours(Node neighbourNode)
        {
            this.neighbours.add(neighbourNode);
        }
    }
}
```

```

    }
    public List<Node> getNeighbours() {
return neighbours;
    }
    public void setNeighbours(List<Node> neighbours) {    this.neighbours =
neighbours;
    }
}

public Main()
{
    queue = new LinkedList<Node>();
}

public void bfs(Node node)
{
queue.add(node);
node.visited=true;
    while (!queue.isEmpty())
    {

        Node element=queue.remove();
        System.out.print(element.data + " ");
        List<Node> neighbours=element.getNeighbours();
        for (int i = 0; i < neighbours.size(); i++) {
            Node n=neighbours.get(i);
if(n!=null && !n.visited)
            {
                queue.add(n);
                n.visited=true;

            }
        }
    }
}

public static void main(String args[])
{

```

```

        Node node1 =new Node(1);
        Node node2 =new Node(2);
        Node node3 =new Node(3);
        Node node4 =new Node(4);
        Node node5 =new Node(5);
        Node node6 =new Node(6);
node1.addneighbours(node2);        node2.addneighbours(node1);
        node1.addneighbours(node3);
node3.addneighbours(node1);        node2.addneighbours(node5);
        node5.addneighbours(node2);
node3.addneighbours(node5);        node5.addneighbours(node3);
        node4.addneighbours(node5);
node5.addneighbours(node4);        node2.addneighbours(node4);
        node4.addneighbours(node2);
node4.addneighbours(node6);        node6.addneighbours(node4);
        node5.addneighbours(node6);
node6.addneighbours(node5);

        System.out.println(" BFS traversal of the graph is : ");
        Main obj = new Main();
        obj.bfs(node1);
        System.exit(0);
    }
}

```

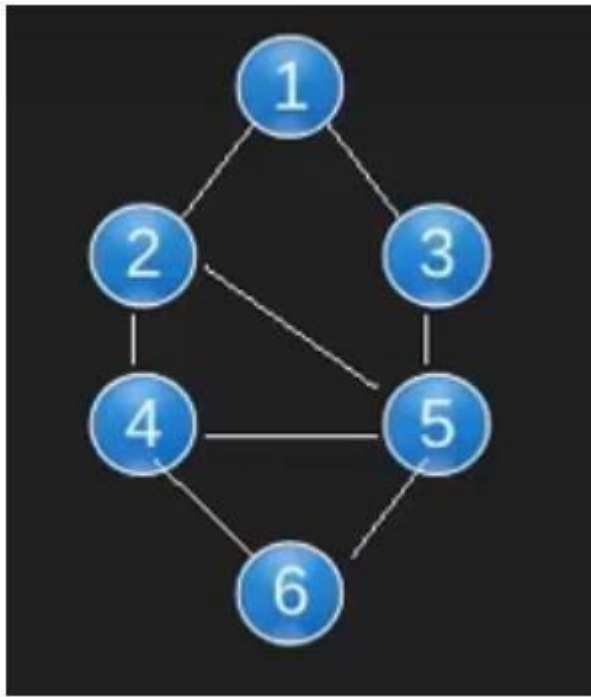
OUTPUT:

```

BFS traversal
of the graph is
:
                1
2   3   5   4   6

```

AIM: To find the depth first search transversal using java program:



Program:

```
import    java.util.*;

import java.io.*; public

class Main

{

    static class Node

    {

int data;

boolean visited;

List<Node> neighbours;
```

```

        Node(int data)
        {
            this.data=data;

            this.neighbours=new ArrayList<>();    }

    public void addneighbours(Node neighbourNode)
    {
        this.neighbours.add(neighbourNode);
    }

    public List<Node> getNeighbours() {
        return neighbours;
    }

    public void setNeighbours(List<Node> neighbours) {
this.neighbours = neighbours;
    }

}

    public void DFS(Node node)
    {
        Stack<Node>    stack=new        Stack<Node>();
        stack.add(node); while (!stack.isEmpty())
        {
            Node element=stack.pop();

            if(!element.visited)
            {
                System.out.print(element.data + " ");

                element.visited=true;

```

```

    }

    List<Node> neighbours=element.getNeighbours();

    for (int i = 0; i < neighbours.size(); i++) {

        Node n=neighbours.get(i);

        if(n!=null && !n.visited)

            {

                stack.add(n);

            }

        }

    }
}

```

```

public static void main(String arg[])
{

```

```

    Node node1 =new Node(1);

```

```

    Node node2 =new Node(2);

```

```

    Node node3 =new Node(3);

```

```

    Node node4 =new Node(4);

```

```

    Node node5 =new Node(5);

```

```

    Node node6 =new Node(6);

```

```

    node1.addneighbours(node2);

```

```

    node2.addneighbours(node1);    node1.addneighbours(node3);
    node3.addneighbours(node1);    node2.addneighbours(node5);
    node5.addneighbours(node2);    node3.addneighbours(node5);
    node5.addneighbours(node3);    node4.addneighbours(node5);
    node5.addneighbours(node4);    node2.addneighbours(node4);

```

```
node4.addneighbours(node2);    node4.addneighbours(node6);  
node6.addneighbours(node4);    node5.addneighbours(node6);  
node6.addneighbours(node5);
```

```
    Main obj = new Main();
```

```
    System.out.println(" DFS traversal of the graph using stack :");
```

```
    obj.DFS(node1);
```

```
    }
```

```
}
```

OUTPUT:

```
DFS traversal of the graph using stack :
```

```
1 3 5 6 4 2
```