

Programming assignment 1:

My program in Java:

```
import java.io.File;
import java.util.Scanner;

public class DFSGraph
{
    static int time = 0;
    static int[] pre;
    static int[] post;
    static int[] color;
    static int[][] adj;
    static int[] deg;

    public static void main(String[] args) throws Exception
    {
        String filename = "dfsinput.dat";
        if (args.length > 0)
        {
            filename = args[0];
        }
        Scanner sc = new Scanner(new File(filename));

        String s1 = sc.nextLine();
        int n = Integer.parseInt(s1.trim());

        adj = new int[n][100];
        deg = new int[n];

        while (sc.hasNextLine() == true)
        {
            String line = sc.nextLine();
            if (line.length() < 2)
                continue;

            int colonPos = line.indexOf(":");
            if (colonPos == -1)
            {
                continue;
            }

            String leftStr = line.substring(0, colonPos);
            String rightStr = line.substring(colonPos + 1, line.length());

            int bracketPos = leftStr.indexOf("(");
            int u = 0;
            if (bracketPos != -1)
```

```

        {
            String numStr = leftStr.substring(0, bracketPos);
            u = Integer.parseInt(numStr.trim());
        }
        else
        {
            u = Integer.parseInt(leftStr.trim());
        }

        Scanner sc2 = new Scanner(rightStr);
        while (sc2.hasNextInt() == true)
        {
            int v = sc2.nextInt();
            adj[u][deg[u]] = v;
            deg[u] = deg[u] + 1;
        }
        sc2.close();
    }
    sc.close();

    pre = new int[n];
    post = new int[n];
    color = new int[n];
    time = 0;

    System.out.println("DFS Edge:");
    for (int i = 0; i < n; i++)
    {
        if (color[i] == 0)
        {
            dfs(i);
        }
    }
}

static void dfs(int u)
{
    color[u] = 1;
    time++;
    pre[u] = time;

    for (int i = 0; i < deg[u]; i++)
    {
        int v = adj[u][i];

        if (color[v] == 0)
        {
            System.out.println(u + " -> " + v + " : Tree Edge");

```

```

        dfs(v);
    }
    else if (color[v] == 1)
    {
        System.out.println(u + " -> " + v + " : Back Edge");
    }
    else if (color[v] == 2)
    {
        if (pre[u] < pre[v])
        {
            System.out.println(u + " -> " + v + " : Forward Edge");
        }
        else
        {
            System.out.println(u + " -> " + v + " : Cross Edge");
        }
    }
}

color[u] = 2;
time++;
post[u] = time;
}
}

```

My output:

```

raphn
DFS Edge:
0 -> 1 : Tree Edge
1 -> 0 : Back Edge
1 -> 2 : Tree Edge
2 -> 1 : Back Edge
2 -> 3 : Tree Edge
3 -> 1 : Back Edge
3 -> 2 : Back Edge
2 -> 4 : Tree Edge
4 -> 2 : Back Edge
4 -> 5 : Tree Edge
5 -> 2 : Back Edge
5 -> 4 : Back Edge
2 -> 5 : Forward Edge
2 -> 6 : Tree Edge
6 -> 0 : Back Edge
6 -> 2 : Back Edge
6 -> 7 : Tree Edge
7 -> 6 : Back Edge
1 -> 3 : Forward Edge
0 -> 6 : Forward Edge
0 -> 8 : Tree Edge
8 -> 0 : Back Edge
8 -> 9 : Tree Edge
9 -> 8 : Back Edge
0 -> 10 : Tree Edge
10 -> 0 : Back Edge
PS D:\draft_priv\Algo\Programming assignment>

```