

## 2424 Fishing Net

In a highly modernized fishing village, inhabitants there make a living on fishery. Their major tools, fishing nets, are produced and fixed by computer. After catching fishes each time, together with plenty of fishes, they will bring back the shabby fishing nets, which might be full of leaks. Then they have to inspect those nets. If there exist large leaks, they have to repair them before launching out again.

Obviously, the smaller the leaks in the fishing nets are, the more fishes they will catch. So after coming back, those fishermen will input the information of the fishing nets into the computer to check whether the nets have leaks.

The checking principle is very simple: The computer regards each fishing net as a simple graph constructed by nodes and edges. In the graph, if any circle whose length (the number of edges) is larger than 3 must have at least one chord, the computer will output **Perfect** indicating that the fishnet has no leaks. Otherwise, **Imperfect** will be displayed and the computer will try to repair the net.

**Note:** A circle is a closed loop, which starts from one node, passes through other distinct nodes and back to the starting node. A chord is an edge, which connects two different nodes on the circle, but it does not belong to the set of edges on the circle.

### Input

The input file contains several test cases representing different fishing nets. The last test case in the input file is followed by a line containing '0 0'.

The first line of each test case contains two integers,  $n$  and  $m$ , indicating the number of nodes and edges on the net respectively,  $1 \leq n \leq 1000$ . It is followed by  $m$  lines accounting for the details of the edges. Each line consists of two integers  $x_i$  and  $y_i$ , indicating there is an edge between node  $x_i$  and node  $y_i$ .

### Output

For each test case, display its checking results. The word **Imperfect** suggests that the corresponding fishing net is leaking, while the word **Perfect** stands for a fishing net in good condition.

Follow the output for each net with a blank line.

### Sample Input

```
4 4
1 2
2 3
3 4
4 1
3 3
1 2
2 3
3 1
0 0
```

**Sample Output**

Imperfect

Perfect