Czech ACM Student Chapter
CTU in Prague, Dept. of Computer Science and Engineering
Charles University in Prague, MFF
VŠB TU Ostrava, FEI, Dept. of Computer Science
SUT Bratislava, FEI, Dept. of Computer Science and Engineering

CTU Open Contest 2004

Herbalists

`h.p`, `h.c`, `h.C`

A group of herbalists has decided to move to a new village on the boundary of a huge forest.

The arrangement of houses and paths in the village turned out to be a major problem: Many of the herbalists are friends and want to visit each other often, so they want to have a path between their houses. However the herbalists are also known for quarreling with everyone else. To avoid the chance of meeting with someone they do not like, it is required that no two paths intersect – there should be no crossroads in the village[1]. Each herbalist also wants to be able to visit the forest, of course without having to cross any path – i.e., it is also necessary to make it possible to get from each house to "infinity" without crossing a path.

You are given a description of the relationships between herbalists. Your task is to decide whether such an ideal village can be built.

[1] And of course also no other means of crossing; overpasses spoil the scenery and underpasses destroy the root systems of precious herbs.

Input Specification

The input consists of several instances.

The first line of each instance contains two integers $1 \le H \le 10\,000$ and $F \ge 0$ separated by a single space. H is the number of herbalists. The herbalists have assigned integers between 0 and $H-1$. F is the number pairs of friends. The following F lines of the instance describe the pairs of friends. Each of the lines contains two integers $0 \le h_1, h_2 < H$ separated by a single space, meaning that herbalists with numbers h_1 and h_2 are friends. Each pair of friends is described exactly once.

The instances follow each other immediately, without any separator. The input is terminated by a line containing two zeros.

Output Specification

The output consists of several lines. The i-th line of the output corresponds to the i-th instance. If it is possible to build a village for the corresponding input instance, the output line consists of "Yes, village can be built" string. If it is not possible to build such a village, the output line consists of "No, village cannot be built" string.

Sample Input

```
3 3
0 1
0 2
1 2
4 6
0 1
0 2
0 3
1 2
1 3
2 3
0 0
```

Output for Sample Input

```
Yes, village can be built
No, village cannot be built
```