

**BI-EP2: Efektivní programování 2**

Bicoloring

`bicolor.c`, `bicolor.cc`, `bicolor.java`

The *four-color* theorem states that every planar map can be colored using only four colors in such a way that no region is colored using the same color as a neighbor. After being open for over 100 years, the theorem was proven in 1976 with the assistance of a computer.

Here you are asked to solve a simpler problem. Decide whether a given graph can be bicolored, i.e., can the vertices be painted red and black such that no two adjacent vertices have the same color.

To simplify the problem, you can assume the graph will be undirected, and not contain self-loops (i.e., edges from a vertex to itself).

Input Specification

The input consists of several test cases. Each test case starts with a line containing the number of vertices n , where $1 \leq n \leq 200$. Each vertex is labeled by a number from 0 to $n - 1$. The second line contains the number of edges l . After this, l lines follow, each containing two vertex numbers specifying an edge.

An input with $n = 0$ marks the end of the input and is not to be processed.

Output Specification

Decide whether the input graph can be bicolored, and print the result as either “YES” or “NO”.

Sample Input

```
3
3
0 1
1 2
2 0
9
8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0
```

Output for Sample Input

```
NO
YES
```