H
Peer Review

For scientific conferences, scientists submit papers presenting their ideas, and then review each other's papers to make sure only good papers are presented at the conference. Each paper must be reviewed by several scientists, and scientists must not review papers written by people they collaborate with (including themselves), or review the same paper more than once.

You have been asked to write a program to check if your favorite conference is doing things right. Whether a paper is being reviewed too much, too little, or by the wrong people - the organizers must know before it is too late!

Input

The first line in each test case has two integers, K $(1 \leq K \leq 5)$ and N $(1 \leq N \leq 1000)$. K is the number of reviews that each paper will receive, while N is the number of papers to be reviewed. The conference only accepts papers with a single author, and authors can only present a single paper at the conference.

Each of the next N lines describes an author and includes the name of the institution to which the author belongs, followed by the list of the K papers he or she has been requested to review. It is assumed that researchers from the same institution collaborate with each other, whereas researchers from different institutions don't. All institution names are shorter than 10 characters, and contain only upper or lowercase letters and no whitespace. Since we have as many papers as authors, papers are identified by their author's index; paper 1 was written by the first author in the list, and paper N was written by the last author.

The end of the test cases is marked with a line containing $K = 0$ and $N = 0$. You should generate no output for this line.

Output

For each test case, your program should output `NO PROBLEMS FOUND` (if all rules are being followed) or P `PROBLEMS FOUND`, where P is the number of rule violations found (counting at most 1 violation per paper). If there is exactly one rule violation overall, your program should output `1 PROBLEM FOUND`.

Sample Input

```
2 3
UCM 2 3
UAM 1 3
UPM 1 2
2 3
UCM 2 3
UAM 1 2
UPM 2 2
0 0
```

Sample Output

```
NO PROBLEMS FOUND
3 PROBLEMS FOUND
```