Grumpy uncle

[Due to SPOJ restrictions, this problem has been modified with respect to the original version used in the Argentinian Programming Tournament of 2016 in order to have multiple test cases per input file. The original version of this problem (in Spanish) can be found at http://torneoprogramacion.com.ar/wp-content/uploads/2016/09/tap2016.pdf]

Winters in Nlogonia are very hard, so uncle Ernie decided to buy a heater to keep himself warm this year. Although it was terribly difficult, he managed to follow his friends' advice and buy an intelligent heater, which can be controlled from his smartphone. However, uncle Ernie doesn't fully understand his smartphone, so he has trouble finding the app which is used to set the heater's temperature.

Uncle Ernie has installed on his smartphone **N** apps numbered from **1** to **N**, corresponding number **1** to the app controlling the heater. His smartphone has **M** buttons numbered from **1** to **M**, which are used to switch from one app to another. More specifically, if the smartphone has currently app number **i** opened and uncle Ernie presses button number **j**, then app **i** is closed and app $T_{i,j}$ is then opened. The problem is uncle Ernie cannot distinguish one app from the other, so he never knows if he has opened the correct app.

Uncle Ernie is very grumpy, so you have decided to help him in order to avoid having to listen to his complaints every time the heater's temperature is not quite right for him. Your task is to provide him with a list of buttons which he can use as instructions, such that if he presses the buttons in the list in the order in which they appear there, his phone will have opened the app controlling the heater. Because you don't want to provide him with more than one list, you should create one that works as explained above independently of which app is opened at the time when uncle Ernie begins executing its instructions.

Take for example the case where the phone has **N = 3** apps and **M = 2** buttons, being $T_{1,1} = T_{2,1} = 3$, $T_{3,1} = T_{1,2} = 2$ and $T_{2,2} = T_{3,2} = 1$. In this case, a sequence of buttons that could be provided to uncle Ernie would be **{1,2}**, because one of the following situations would take place:

- If uncle Ernie starts with app **1** being open, pressing button **1** will leave him with app **3** being open; then pressing button **2** would result in a final state where app **1** is again opened.
- On the other hand, if uncle Ernie starts with app **2** being open, pressing button **1** will leave him with app **3** being open; then pressing button **2** would result in a final state where app **1** is open.
- Finally, if he starts with app **3** being open, pressing button **1** will leave him with app **2** being open; then pressing button **2** would reslut in a final state where app **1** is open.

Therefore, independently of which app was open at the beginning of the sequence of button presses, uncle Ernie will always reach app **1** by the end of the sequence.

Now again, there are times when it is impossible to find a sequence of buttons that can be pressed by uncle Ernie such that the app being open when he finishes to do so is always app number **1**. For example, in the case with **N = 3** and **M = 2** if we have $T_{1,1} = T_{1,2} = 2$, $T_{2,1} = T_{2,2} = 3$ and $T_{3,1} = T_{3,2} = 1$ the application opened at the end of any sequence of button presses always

depends on which application was open at the time the sequence was started. Therefore, in this case it is impossible to find a sequence that will always leave uncle Ernie's phone with app number **1** opened.

In order not to waste time looking for sequences of button presses that do not exist, you would like to first find out if it is possible to satisfy uncle Ernie by finding a sequence as described above. If so, uncle Ernie will be happy knowing that he can turn the heater on at level 2, his favorite, whenever he wants... and will thus be eternally grateful.

Input

There are multiple test cases in the input file. The first line contains two integer numbers **N** and **M**, representing the number of applications and the number of buttons in uncle Ernie's smartphone, respectively ($1 \leq N, M \leq 1000$ with $1 \leq N \times M \leq 10^4$). Each of the following **N** lines contains **M** integer numbers, being the **j**-th number in the **i**-th line $T_{i,j}$, representing the app which is opened when button **j** is pressed while app number **i** was open ($1 \leq T_{i,j} \leq N$ for **i = 1, 2, ..., N** and **j = 1, 2, ..., M**).

Output

For each test case, print a single line containing a character, representing if it is possible to find a sequence of buttons as described in the problem statement. The character should be an **'S'** if the sequence can be found, and an **'N'** otherwise.

Example

Input:
```
3 2
3 2
3 1
2 1
3 2
2 2
3 3
1 1
```

Output:
```
S
N
```