Problem C - Lt. Surge's New Puzzle

Fed up with people rummaging through trash cans while walking around randomly and tossing stuff all over his gym floor, Lt. Surge replaced the system controlling his electric gates.

In this new system, there are **N** wires ($1 \leq \mathbf{N} \leq 100$) connected to the gate, and **M** switches ($1 \leq \mathbf{M} \leq 100$) on the wall. Each wire is connected to exactly one switch, but a switch may be connected to multiple wires, and it is impossible to tell just by looking which switch each wire is connected to.

Flipping on a switch causes a current to flow through all of the wires the switch is connected to. It is possible to attach a probe to one wire at a time to detect whether it carries a current.

To deactivate the gates, you need to send current through a specific set of wires. To figure this out, you need to first learn which switch controls each wire. Can you solve this new puzzle?

Input

This is an interactive problem. This means that your program will interact with a judge program on the server through `STDIN` and `STDOUT`.

The first line contains the integer **T**, then **T** test cases follow.

Each test case contains 2 space separated integers, **N** and **M**. Initially, all the switches are off, and there is no current in any of the wires.

Two operations are available to you:
FLIP i: Flip switch i (1-indexed)
PROBE i: Check if wire i carries current (1-indexed)

After each **PROBE** command, **YES** or **NO** will be returned representing whether wire i has current flowing.

Lt. Surge doesn't like people snooping around, so if after **2000** total operations you still have not figured out the mapping, he will throw you out of the gym (i.e. the judge returns Wrong Answer). If you take too long, he will throw you out as well (e.g. If your program does not output any operations or if it does not terminate after finishing **T** test cases, it will be judged as Time Limit Exceeded).

Note: Be sure to flush your output buffer after each probe. In C++, this means using either `endl` or `fflush`.

Output

For each test case, when you have determined the matchings, output two lines containing:
DONE
S1 S2 ... SN
where **Si** is the **1-based** index of the switch that is connected to the i-th wire.

Sample Input

```
1
2 3
NO
NO
YES
NO
```

Sample Output

```
FLIP 1
PROBE 1
PROBE 2
FLIP 2
PROBE 1
PROBE 2
DONE
2 3
```