Task: BIT2
Three Bit Computer strikes back

day 4, source file `bit2.*`, available memory 32 MB *CPSPC 2004, June 11*

The Three Bit Computer is a huge failure in spite of continuing efforts of Byteland's leading scientists, but now they have a new and powerful idea: the Quantum Three Bit Computer (QTBC). TBC can now be regarded a warm-up before the real thing — the QTBC.

They predict that the new machine will have the power, blah, blah., many silly problems still to solve, blah, blah. Now, let us get to the point.

The initialization procedure is again a major issue, but with quantum computers it is a completely different story. The trouble now is that everything you do has side effects, i.e., it influences everything else. Taking care of this is very time consuming and initializing the memory bit by bit is simply impossible. But there is another approach. The scientists have developed large scale controlled impulses (LSCI), which influence all the memory bits at the same time, and moreover, it is known exactly what their influence is. They are also very fast, so emitting even a large number of impulses is better then initializing the memory bit by bit. The first question to consider is whether there exist a sequence of impulses that will zero the whole memory. Your task is to write a computer program to answer this question.

More formally, each memory bit can be in one of n states numbered $0, \ldots, n-1$. The LSCI impulse f changes all the bits in exactly the same way, i.e. it can be viewed as a function $f : \{0, \ldots, n-1\} \to \{0, \ldots, n-1\}$. For example, $f(3) = 5$ means that if the impulse f is emitted, then every bit in state 3 will change its state to 5. The scientists know how to emit several impulses $f_1, \ldots, f_k$. You have to find out if there exists a sequence of impulses that brings all the bits to state 0 regardless of their initial state.

Task

Write a program that:

- reads the descriptions of the impulses available,
- checks if zeroing the memory is possible,
- writes the answer to the output file.

Input

The input file `bit2.in` can contain several test cases. The first line of the input file contains a single positive integer T, $(1 \le T \le 10)$, the number of test cases. The description of the test cases follows. Description of a single test case starts with a line containing two positive integers n, k, $(1 \le n \le 200,\ 1 \le k \le 5)$, where n is the number of different states of memory bits and k is the number of different impulses that can be emitted. The next k lines contain the description of the impulses, the i-th line contains the description of the i-th impulse. The description of an impulse f is a sequence of integers $f(0) \ldots f(n-1)$ describing the influence f has on the state of any memory bit. These integers are separated by single spaces.

Output

The output file `bit2.out` should have T lines, one for each test case. The i-th line should consist of a single word `YES` if zeroing the memory in the i-th test case if possible, or `NO` otherwise.

Example

For the input file `bit2.in`:

```
2
5 2
1 2 3 4 0
2 3 4 0 1
5 2
1 2 3 4 0
3 3 4 0 1
```

the correct result is the output file `bit2.out`:

```
NO
YES
```