Problem A - Zé Manel plays with deterministic finite automata

---

Description

While attending a lecture of Theory of Computation about finite state machines, Zé Manel decided to make a game based on the topic of deterministic finite automata.

Do you remember what a deterministic finite automaton is? It consists of

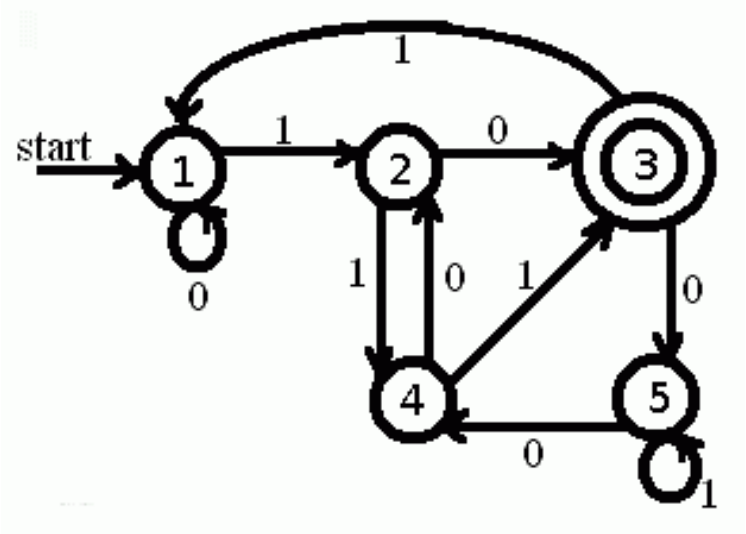
- a finite set of states Q
- a finite set of input symbols called the alphabet (Σ)
- a transition function ($\delta : Q \times \Sigma \mapsto Q$)
- a start state ($q0 \in Q$)
- a set of accept states ($F \subseteq Q$)

A automaton accepts a given string of size m if and only if there exists a sequence of m transitions between the states in Q such that

- the first state is $q0$
- given each symbol of the string, the machine will transition from state to state according to function δ
- the last symbol of the string will cause the machine to halt in one of the accept states

Zé Manel defined a deterministic finite automaton where $Q = \{1,2, ..., n\}$, $\Sigma = \{0,1\}$ and $q0 = 1$ and δ is given by a state transition table. Then, the goal is to count how many distinct strings of size m exist that can be accepted by the automaton.

A graphical example of a deterministic finite automaton with $n=5$ and $F = \{3\}$ is given in the figure below, with nodes for states, and arcs for transitions. Each node has two outgoing arcs, corresponding to the symbols 0 and 1, respectively. For such example there exist 2 strings of size 3 that can be accepted by the automaton: 010 and 111. But for size 20, there exist 209715 such strings!



Input

The first line indicates the number of states and the size of the strings. Then, the transition table is given in the following *n* lines. The *i*-th line has at least two integers, corresponding to the two states that are reachable from state *i* for symbols 0 and 1, respectively. If state *i* is an accept state, then *i*-th line ends with symbol *. Other test cases may follow.

Output

For each test case, print the number of strings can be accepted by the automaton. Note that there may be more than one accept state.

Constraints

- *1 ≤ n ≤ 20*
- *1 ≤ m ≤ 30*

Example

Example input:

```
5 20
1 2
3 4
5 1 *
2 3
4 5
5 3
```

```
1 2
3 4
5 1 *
2 3
4 5
```

Example output:

```
209715
2
```
