Problem E. Odd Grammar

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

A *formal grammar* is a way of describing formal languages as $\Gamma = \langle \Sigma, N, S \in N, P \subset N^+ \times (\Sigma \cup N)^* \rangle$ where Σ is called an *alphabet* and its elements are called *terminals*, N is a set of *nonterminals*, S is the starting nonterminal, and P is a set of *production rules* of the form $\alpha \to \beta$.

Here, N^+ contains all strings of one or more elements of N (non-empty strings of nonterminals), and $(\Sigma \cup N)^*$ consists of all strings of zero, one or more elements of $(\Sigma \cup N)$ (strings of terminals and nonterminals, including the empty string).

A grammar is called *context-free* if the left side of each production rule consists of exactly one nonterminal, more formally, $P \subset N \times (\Sigma \cup N)^*$.

For example, let us consider a grammar from the second example test case with alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, set of nonterminals $N = \{S, A\}$ and two production rules:

1. $S \to \mathtt{b}A$
2. $A \to \mathtt{aa}$

One can easily see that it is a context-free grammar.

To create the language generated by a grammar, one needs to start from a string consisting of only start nonterminal S, and then apply production rules one or more times. Applying a production rule is the procedure of finding the left side of that rule somewhere in the current string and replacing it by the string from the right side of that rule. The *language* generated by Γ is the set of all strings consisting **only** of terminals that can be produced by applying production rules one or more times.

For example, there is a string `baa` in the language generated by the grammar described above. To produce it, one could apply productions $S \to \mathtt{b}A \to \mathtt{baa}$. There are no other strings in the language generated by this grammar.

Some grammars may even generate infinite languages, others may generate empty ones.

You are given a context-free grammar with an alphabet consisting of two terminals “`a`” and “`b`”. Your task is to check whether the language generated by this grammar contains a string of **odd** length.

Nonterminals in this task are enumerated from 1 to n. The starting nonterminal always has number 1.

Input

The input consists of one or more test cases.

The first line of each test case contains two integers n and m: the number of nonterminals and the number of production rules ($1 \le n \le 50000$, $1 \le m \le 200\,000$).

Each of the next m lines describes one production rule in the following manner. At first, A_i and k_i are given: the number of left side nonterminal ($1 \le A_i \le n$) and the number of characters on the right side of the production rule ($0 \le k_i \le 5000$). Then k_i objects follow, each of them is either a nonterminal $B_{i,j}$ ($1 \le B_{i,j} \le n$) or a terminal “`a`” or “`b`”. Consecutive characters are separated by single spaces.

The total sum of all n over all test cases does not exceed $50\,000$. The total sum of all m over all test cases does not exceed $200\,000$. The size of the input does not exceed 5 megabytes.

The input is terminated by a string of two zeroes.

Output

For each test case, output a separate line. It must contain “`YES`” if the language generated by the given grammar contains a string of **odd** length, otherwise the line must contain “`NO`”.

Example

standard input	standard output
2 2 1 2 a 2 2 1 b 2 2 1 2 b 2 2 2 a a 2 2 1 2 b 2 2 3 a a 1 0 0	NO YES NO