The NFA Counter

Let us recall some necessary definitions. The concepts defined here are probably very well known to you, we provide the definitions just for the sake of completeness.

A **nondeterministic finite automaton (NFA)** X is a five-tuple $(\Sigma, Q, q_0, \delta, F)$ where

- Σ is an alphabet consisting of A ordered characters $a_0 < a_1 < \dots a_{A-1}$, $(1 \le A < \infty)$,
- Q is a nonempty set of states,
- Q_0 is a start state, $q_0 \in Q$,
- δ is a transition function δ: $Q \times \Sigma \rightarrow \mathcal{P}(Q)$,
- F is a nonempty subset of Q, it is a set of final states.

Symbol $\mathcal{P}(Q)$ (power set) denotes the set of all subsets of Q including Q itself and empty set.

The Task

Your task is either to count all different words that are accepted by a given input nondeterministic automaton X or detect that the given input NFA X accepts infinitely many words.

Input

Input specifies NFA $X = (\Sigma, Q, q_0, \delta, F)$.

The first line contains two integer numbers S and A, where S represents number of states of X and A represents size of alphabet of X.

We assume that $Q=\{0,1,2, \dots, S-1\}$ for $S>0$, $q_0=0$, and Σ is a subset of {'a', 'b', …,'z'} where $\Sigma=\{a_0, a_1, \dots, a_{A-1}\}$ for $1 \le A \le 26$.

Next S lines contain definition of the transition function δ. Each line starts with state number q_j and then contains *simple set format listings* of sets $\delta(q_j, a_0)$, $\delta(q_j, a_0)$, ..., $\delta(q_j, a_{A-1})$, where *simple set format listing* of set M is a sequence of k+1 integers (k=|M|) where k is the first element of the sequence followed by all elements of M in arbitrary order.

The last line of the input contains *simple set format listing* of the set F.

All values on any input line are separated by one or more spaces.

You may assume that $S \times A \le 10000$.

Output

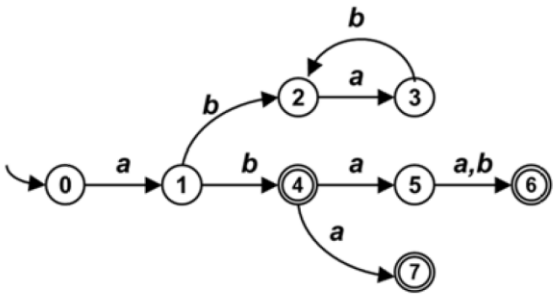
Output contains only one line with the number of all different words accepted by the input NFA X. If the input NFA X accepts infinitely many words then the output is -1.

You may assume that the output integer does not exceed 2^{60}.

Example 1

Input:

```
8 2
0   1 1     0
1   0       2 2 4
2   1 3     0
3   0       1 2
4   2 5 7   0
5   1 6     1 6
6   0       0
7   0       0
3 4 6 7
```



Output:

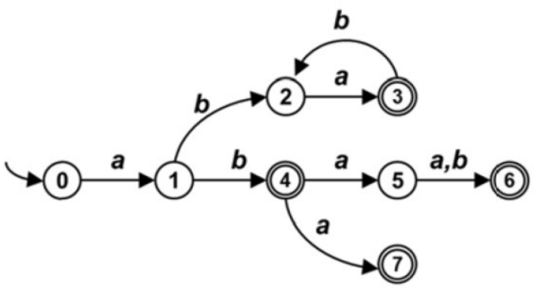
4

The transition diagram of the input automaton is depicted on the right-hand side of the input data.

Example 2

Input:

```
8 2
0   1 1     0
1   0       2 2 4
2   1 3     0
3   0       1 2
4   2 5 7   0
5   1 6     1 6
6   0       0
7   0       0
4 3 4 6 7
```



Output:

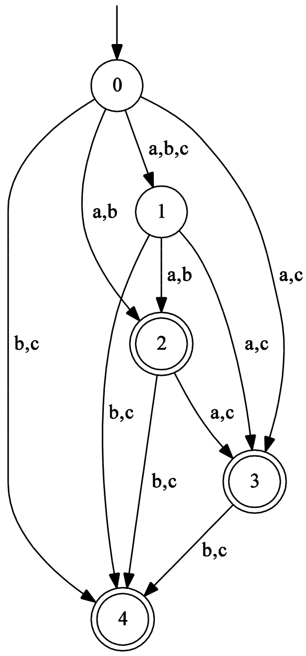
-1

The transition diagram of the input automaton is depicted on the right-hand side of the input data. Please note, that Example 1 and Example 2 differ only in their final states specifications.

Example 3

Input:

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



Output:

60

The transition diagram of the input automaton is depicted on the right-hand side of the input data.