

## 2420 NTA

The NTA (Non-deterministic Tree Automata) is a kind of tree structure device. The device is built in a set of operating rules. With these rules the device can produce several signals, which will form a signal system. In such a system, one signal is the starting signal, several signals are the acceptable signals, and the others are the auxiliary ones. A pair of signals is said to be an acceptable pair if both two signals of the pair are acceptable.

The trees discussed here are all binary trees. Every non-leaf node has two successors. In any finite tree, each node has a signal-transmitting element. When a signal arrives at one node, the signal meets the signal transmitting substance, and triggers off signal reactions, which will produce several pairs of signals. Then the device selects a pair of signals non-deterministically and sends them to its successors. The first signal in the signal pair is sent to the left successive node and the second one is sent to the right successive node.

The whole operation for an NTA is as follows:

The device first sends the starting signal to the root node. According to the signal transmitting substance at the root node, the device selects a pair of signals non-deterministically and sends the first to the left son and the second to the right son. Each of the two signals then meets the signal transmitting substance at the corresponding node and produces another two signals. The course proceeds down until the signals arrive at the leaves.

If a signal reaches one leaf and the leaf can produce a pair of acceptable signals, we say the leaf is “shakable”. A transmission of signals from the root to leaves is said to be valid if all leaves are “shakable”. A tree structure with signal transmitting substance is valid if there exists such a valid transmission. A tree is invalid if all the transmissions are invalid.

For simplicity, we denote the signal transmitting elements by consecutive lowercase letters “a”, “b”, “c”, etc.. The signals of an NTA are consecutive numbers 0, 1, 2, ..., and so on. The first signal 0 is always a starting signal. Thus the signals for a 4-signal NTA are “0” “1” “2” and “3”. Accepting signals are arranged at the end of the number sequence so that if a 4-signal NTA has two accepting signals, the accepting signals are “2” and “3”. The transition rules of signals are based on a transition table. For example, the following table describes a transition table with four signals “0”, “1”, “2”, “3” and with three signal transmitting elements “a”, “b” and “c”.

T	a	b	c
0	(1,2)	(2,1)	(1,0)
1	(2,2)	(0,2)(1,0)	(3,2)
2	(2,2)	(2,3)	(1,2)
3	(1,2)	(2,1)	(3,2)

In this transition table some reactions of signals on certain signal transmitting elements are deterministic, and others are non-deterministic. In the example above, if signal “1” reaches the node with the transmitting element “b”, the reaction is non-deterministic.

Now your task is to write a program to judge if a tree structure with certain signal transmitting substance is valid.

### Input

The input file contains several cases. Each case describes a sequence of NTA descriptions and some initial tree configurations. The first line for each case consists of three integers  $n$ ,  $m$  and  $k$ . The integer

$n$  is the number of signals,  $m$  indicates the number of accepting signals, and  $k$  is number of signal transmitting elements. The following  $n \times k$  lines describe the transition table in row-major order. Each transition of a signal on signal transmitting element is given on a separate line. On such line every two numbers represent a possible transition.

This is followed by the description of tree structures. For every tree structure a number  $L$  is given on a separate line to indicate the level of the tree. The following  $L + 1$  lines containing a sequence of letters describe the tree structure. Each level is described in one line. There exist one space between two successive letters. The 0-th level begins firstly. In the tree structure, the empty nodes are marked by '\*'. The tree structure with  $L = -1$  terminates the configurations of tree structures for that NTA, and this structure should not be judged.

The input is terminated by a description starting with  $n = 0$ ,  $m = 0$  and  $k = 0$ . This description should not be processed.

**Note:** In each case, NTA will have at most 15 signals and 10 characters. The level of each tree will be no more than 10.

## Output

For each NTA description, print the number of the NTA ('NTA1', 'NTA2', etc.) followed by a colon. Then for each initial tree configuration of the NTA print the word 'Valid' or 'Invalid'.

Print a blank line between NTA cases.

## Sample Input

```
4 2 3
1 2
2 1
1 0
2 2
0 2 1 0
3 2
2 2
2 3
1 2
1 2
2 1
3 2
3
a
b c
a b c b
b a b a c a * *
2
b
a b
b c * *
-1
0 0 0
```

## Sample Output

NTA1:

Valid

Invalid