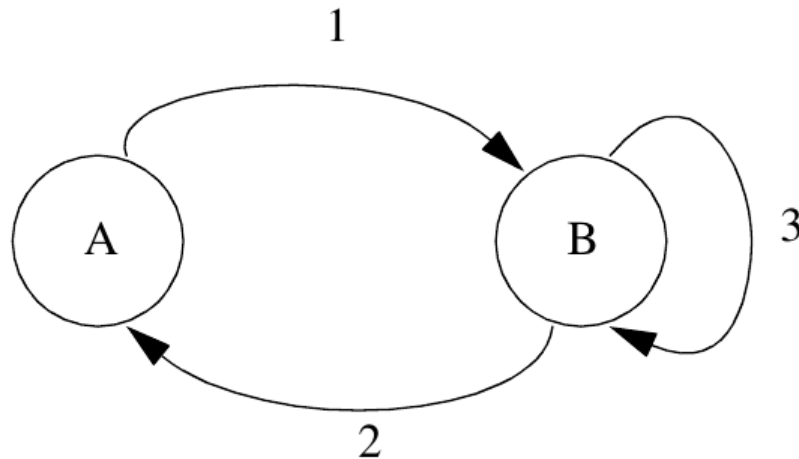


## 5436 The Finite State Text-Processing Machine

A finite state machine (FSM) is essentially a directed graph. Each node in the graph is called a state; at any point during execution of the FSM, one of the states is said to be the current state. Each directed edge between two states is called a transition. When the conditions are right, one of the transitions from the current state is said to occur, and the current state changes to a new state as determined by the transition. Consider the following very simple example.



There are two states in this FSM, labelled A and B, and three transitions, labelled 1, 2 and 3. If the current state is A, and the conditions for transition 1 are met, then the current state becomes B. When the current state is B, and the conditions for transition 2 are met, the current state becomes A again. If the current state is B and the conditions for transition 3 are met, the current state remains B.

In this problem the input will be descriptions of several FSMs. Each transition in these FSMs has an associated set of characters called the input set, and a string called the output string. A transition can occur when the current input data character is in the transition's input set. When the transition occurs, the output string is printed.

### Input

The input consists of a sequence of pairs {FSM description, input for the FSM}. A FSM is described by the following sequence of items separated by whitespace (blanks, tabs, and end of line characters):

- An integer specifying the number of states in the FSM. For each of these states there will be the following items, in order:
  - A one to eight character name for the state. State names are case significant, and unique.
  - An integer specifying the number of transitions that leave the current state. For each of these transitions there will be the following data items, in order:
    - \* The set of input characters that enable the transition (see below).
    - \* The name of the new current state when this transition occurs.
    - \* The output string (see below).

The input set and the output string are given as sequences of printable characters with no embedded whitespace. Several special constructions may appear in these, however. When ‘\b’ appears it is to be interpreted as a blank. Treat ‘\n’ as an end of line, and ‘\’ as a single backslash. The construction ‘\0’ (that is, backslash followed by zero) will appear only as an output string, and means to print nothing when the transition occurs. The construction ‘\c’ appearing as an input set matches anything. That is, if none of the other transitions are enabled and a transition has ‘\c’ specified as its input set, then it is enabled. When ‘\c’ appears in an output string, it means to print the current input character. This could appear several times in the same output string.

After the last item in the description of a FSM has been read begin “executing” the FSM using the characters that start on the first complete line following the description. The beginning state will always be called ‘START’. The final state will always be called ‘END’, but will not appear as a state in the description of a FSM. All input is guaranteed to be correct.

## Output

For each input your program should display the FSM number (1, 2, ...) and, beginning on the next line, the output that results from those transitions that occur. The examples below illustrate this.

### Examples:

The first example FSM replaces each vowel in a single line of input with an asterisk.

The second example will delete each vowel that follows a lower or upper case X, again processing only a single line of input.

The final example changes the case of each odd-numbered vowel in the input; processing stops when an exclamation point is encountered, and the remainder of the input line is ignored.

## Sample Input

```

1
START 3
    AEIOUaeiou  START *
    \n          END  \n
    \c          START \c
This is input for example one.
2
START 3
    \c  START \c
    Xx  SKIP  \c
    \n  END  \n
SKIP 4
    AEIOU START \0
    aeiou START \0
    Xx  SKIP  \c
    \n  END  \n
XaXxe      Test  XIo  iXix0
3
START 12
    \c START \c  !  FINIS \0
    A TWO a E TWO e
    I TWO i O TWO o
    U TWO u a TWO A
    e TWO E i TWO I
    o TWO O u TWO U

```

```
TWO 4
  \c TWO  \c  AEIOU START \c
    aeiou START \c ! FINIS \0
FINIS 2
  \c FINIS \0  \n  END  \n
This is some data for FSM number 3.
!  IGNORED
0
```

### Sample Output

```
Finite State Machine 1:
Th*s *s *np*t f*r *x*mpl* *n*.
Finite State Machine 2:
XXx  Test  Xo  iXx
Finite State Machine 3:
ThIs is sOme dAta fOr FSM numbEr 3.
```