Problem A
Alan Turing

Time limit: 1 second
Memory limit: 256 megabytes

Problem Description

Alan Mathison Turing was a British computer scientist and mathematician. He was a pioneer in theoretical computer science who proposed the Turing machine, an abstract machine defined by a mathematical model of computation.

Here is how a Turing Machine works. It has a certain number of states, a tape of infinite length, on it can a certain set of symbols be written, and it runs within a certain set of transition rules. Starting from an initial state, and a head that points at a certain location, it can decide which direction it wants its head to move towards or to halt, and which symbol it wants to write back to the tape, according to the transition rules which account the current state and the current symbol the head is pointing at on the tape.

A interesting question (halting problem) is if it is possible to determine if a certain Turing machine will eventually halt on a given input (which is the initial symbols on the tape) or just run forever. That is proven to be impossible to determine. However, there are still some good news! It is totally possible to prove if a Turing Machine will halt within 10 steps. And as a challenge to you, we would like you to tell us the answer.

Input Format

The first line of the input will be a single integer T ($T \leq 20$) denoting the number of test cases.

In each test case first the description of a Turing Machine will be given. It starts with an integer n ($1 \leq n \leq 10$) on a line representing the number of states there are in this Turing Machine. In which state 1 is the starting state and state n is the only halting state, i.e., the Turung Machine will halt after entering it.

The next $n-1$ lines will be the transition rules of state 1 to $n-1$, note that since state n is the halting state, it does not have any transition rule. To simplify the problem, there will only be three symbols used in the Turing Machines of this problem, namely 0, 1 and 2. Each line of the transition rule contains 3 tuples in the form of (x, y, z) separated by a blank, in which x represents the next state the Turing Machine will transition to, y represents the direction the head is going, 1 as right and -1 as left, and finally z represents what should be written back onto the tape. Note that the head will first write back the symbol, then move. For line i in these $n-1$ lines, the three 3-tuples will represent the transition of state i when it reads symbol 0, 1 and 2 respectively. For instance that the second 3-tuple of the fifth line will represent the transition rule of state 5 if the head is pointing at a symbol 1.

After that is an integer m ($m \leq 100$) on a line indicating the number of queries there are. Each query will be written on a single line in which the first number x ($0 \leq x \leq 10$) is the number of symbols in this input, after that x symbols would follow. The symbols will be written one after the other on the tape with the initial head position pointing at the first symbol. There will only be symbol 0 and 1 in the input as symbol 2 is actually representing a blank symbol in the Turing Machine. And because of that, you may assume that every symbol on the tape before the input and after the input is 2. For instance, an input `3 1 0 0` will actually look like this on the tape : "`... 2 2 2 1 0 0 2 2 2 ...`", where the head will point at the symbol 1 initially and that the symbol 2 and the start and the beginning will extend infinitely on both ends.

Output Format

For each test case, first print "`Machine #N:`", where N is the number of test case. Then for each query, print "`yes`" on a line if the machine does halt in 10 steps and "`no`" if not.

Sample Input

```
3
3
(2, 1, 0) (2, 1, 1) (2, 1, 2)
(3, 1, 0) (1, -1, 0) (1, -1, 2)
3
3 1 0 0
2 1 1
1 1
4
(2, 1, 1) (3, 1, 0) (1, -1, 2)
(1, -1, 0) (1, -1, 1) (1, -1, 2)
(2, 1, 1) (2, 1, 0) (4, -1, 2)
2
3 0 0 0
5 0 0 0 0 0
2
(1, -1, 2) (2, -1, 0) (1, 1, 2)
2
2 0 0
1 1
```

Sample Output

```
Machine #1:
yes
yes
no
Machine #2:
yes
no
Machine #3:
no
yes
```