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
```

Problem A
Alan Turing
Time limit: 1 second
Memory limit: 256 megabytes

Problem Description

艾倫・麥席森・圖靈 (Alan Mathison Turing) 是位偉大的英國際算機科學家、數學家。作為研究計算機科學的先鋒，提出了一個影響重大的數學抽象模型：圖靈機。

以下將簡單介紹一下圖靈機是如何運作的。圖靈機有一定數量的狀態、一條無窮長且能紀錄特定符號的帶子，依據一組特定的規則進行狀態間的轉移。起始時，圖靈機會處於一個特定的起始狀態，其讀寫頭會指在一個特定的位置。圖靈機會依據當下的狀態以及讀寫頭位置上的符號，來決定接下來將往哪一個方向移動讀寫頭或是停機、要將什麼符號寫回資料帶上、以及下一個狀態將會是哪一個。

「有沒有辦法判斷特定的圖靈機在給定特定的資料帶時，最終會停機還是無止境的持續執行？」便是所謂的「停機問題」，是圖靈機相關理論中的一個有趣問題，可已經被證明是無法判斷的了。然而，停機問題仍有些正面的結果。「特定圖靈機在給定特定的資料帶時，是否能在 10 步內停機」是完全可以判斷的。這個版本的問題，就請你寫個程式來回答吧！

Input Format

測資輸入的第一行，有一個整數 T $(T \leq 20)$ 代表測試資料的筆數。

每一筆測試資料會先描述圖靈機。測試資料的第一行，會有一個整數 n $(1 \leq n \leq 10)$ 代表這臺圖靈機有 n 個狀態。其中狀態 1 是起始狀態，而狀態 n 為唯一的停止狀態：一進入該狀態，圖靈機便停機。

接下來的 $n-1$ 行會依序描述圖靈機狀態 $1, \ldots, n-1$ 的轉移規則。請注意狀態 n 為停止狀態，因此不會有任何的狀態轉移規則。為簡化這個問題，在資料帶上會出現的符號僅有 $0, 1, 2$ 三種。我們將使用三個以一個空白隔開三元組 (3-tuple) 描述轉移規則。每個三元組 (x, y, z) 代表圖靈機將轉移到狀態 x 、$y = 1$ 時讀寫頭將往右移、$y = -1$ 時讀寫頭將往左移、z 則是寫回資料帶的符號。在本題的設定中，圖靈機將先將符號寫回資料帶，再進行讀寫頭的移動。這 $n-1$ 行中的第 i 行的三個三元組，分別代表了在狀態 i 讀寫頭讀到符號 0、i、2 的三種狀態轉移。舉例來說，第五行的第二個三元組如果是狀態 5 讀到符號 1 時該進行的狀態轉移。

接下來的測試資料會有一個整數 m 代表這組測試資料有 m 筆查詢。每筆查詢會有幾個數字寫在單獨一行。其中第一個數字 x $(0 \leq x \leq 10)$ 代表這筆查詢有連續 x 個符號預先寫在資料帶上。接下來共有 x 個數字，就代表寫在讀寫頭起始位置起，往右方延續的符號。這 x 個數字只會有 0 跟 1，符號 2 其實是空白。因此當輸入為 `3 1 0 0` 時，其實資料帶看起來是 `... 2 2 2 1 0 0 2 2 2 ...` ，且初始時讀寫頭指在 1 的位置，在開頭前方與結尾後方都有無窮多個空白 (符號 2)。

Output Format

對第 N 組測試資料請先印出 `Machine #N`。隊每一筆查詢，如果能在 10 步內停機，則印出 `yes` 否則印出 `no`。

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
```