We are examining two specific classes of languages (a possibly infinite set of strings) in this problem. Fortunately (or maybe unfortunately), we are not given the strings contained in each language directly, rather we are given two deterministic finite automatons that describe such languages.

Your task here is simple: to determine if the languages described by the two automatons are the same.

For those of you who are unfamiliar with the concept of deterministic finite automata, you may use the following information:

A deterministic finite automata (DFA) is a 5-tuple, (S, Σ, T, s, A), consisting of

- a finite set of states (S)
- a finite set called the alphabet (Σ)
- a transition function ($T : S \times \Sigma \rightarrow S$)
- a start state ($s \in S$)
- a set of accept states ($A \subseteq S$)

Let M be a DFA such that $M = (S, \Sigma, T, s, A)$, and $X = x_0 x_1 ... x_n$ be a string over the alphabet Σ. M accepts the string X if a sequence of states, $r_0, r_1, ..., r_n$, exists in S with the following conditions:

1. $r_0 = s$
2. $r_{i+1} = T(r_i, x_i)$, for $i = 0, ..., n - 1$
3. $r_n \in A$.

- Wikipedia

Input

There are multiple test cases in the input file.

The first line of each test case is one integer, T ($2 \leq T \leq 26$), the size of the alphabet. In each test case, the description of automaton A comes before that of automaton B. The description of each automaton starts with one line containing N ($1 \leq N \leq 2000$), the number of states in the automaton, followed by N lines, each line of the format: $F, X_0, X_1, ..., X_{T-1}$, ($F \in \{0, 1\}$, $-1 \leq X_i < N$). If $F = 1$, then the state is an accepting state; also, if $X_i = -1$, it means that the state has no corresponding transition available for character i. The start state of both finite automatons will always be state 0.

Two successive test cases are separated by a blank line. A case with a single `0' indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print `Yes' if the two languages described by the automatons are equivalent; output `No' otherwise.

Sample Input

```
2
3
1 -1 1
0 -1 2
0 -1 0
2
1 -1 1
0 -1 0

3
4
1 -1 -1 1
1 -1 -1 2
1 -1 -1 3
1 -1 -1 1
2
1 -1 -1 1
1 -1 -1 0

0
```

Sample Output

```
Case #1: No
Case #2: Yes
```

Hangzhou 2008-2009