Boatherds

`b.p`, `b.c`, `b.C`

Boatherds Inc. is a sailing company operating in the country of Trabantustan and offering boat trips on Trabantian rivers. All the rivers originate somewhere in the mountains and on their way down to the lowlands they gradually join and finally the single resulting river flows to the sea. Moreover, the Trabantian villages are exactly at the rivers' springs, junctions and at the mouth of the largest river. Please note that more than 2 rivers can join at a junction. However, the rivers always form a tree (with villages as vertices).

The pricing policy of the Boatherds is very simple: each segment of each river between two villages is assigned a price (the price is same in both directions), so if a tourist requests a journey between any two villages, the ticket office clerks just add the prices of the segments along the only path between the villages.

One day, a very strange tourist appeared. She told the clerks that she returns to her country on the next day and she wants to spend all the remaining money on a boat trip, so they should find a route with exactly this cost. Being just poor (ahem) businessmen, they have asked the Abacus Calculator Makers for help.

You are given a description of the river network with costs of river segments and a sequence of integers $x_1, \ldots, x_k$. For each x_i, you should determine if there is a pair of cities (a, b) in the river network such that the cost of the trip between a and b is exactly x_i.

Input Specification

The input consists of several instances. Each instance is described by (in the following order):

- A single line containing a single integer: the number of villages N ($1 \leq N \leq 10\,000$).
- N lines describing the villages. The i-th of these lines ($1 \leq i \leq N$) describes the village with number i. It contains space separated integers d_1, c_1, d_2, c_2, $\ldots$, d_{k_i}, c_{k_i}, 0. The d_j's are numbers of villages from which the rivers flow directly to the village i (with no other villages in between), each c_j is the price of the journey between villages i and d_j. Moreover, $2 \leq d_j \leq N$ and $0 \leq c_j \leq 1\,000$. Village 1 always corresponds to the mouth of the largest river, therefore no d_i can ever be equal to 1.
- $M \leq 100$ lines describing the queries. The i-th of these lines corresponds to the i-th query and contains a single integer x_i ($1 \leq x_i \leq 10\,000\,000$).
- The instance is finished by a single line containing the number 0.

The whole input is ended by a single line containing the number 0.

Output Specification

For each instance you should produce a sequence of M lines (where M is the number of queries in the particular instance). The i-th of these lines contains the word "AYE" if there exists a pair of cities in the river network which is connected by a path of cost x_i, or the word "NAY" otherwise.

Output for each instance must be followed by a single line containing just the dot character.

Sample Input

```
6
2 5 3 7 4 1 0
0
5 2 6 3 0
0
0
0
1
8
13
14
0
0
```

Output for Sample Input

```
AYE
AYE
NAY
AYE
.
```