Problem A
Fuzzyland

Input File: A.DAT
Program Source File: A.PAS or A.C or A.CPP

On his journey around the world, a traveler has found himself in Fuzzyland. Fuzzyland has a set of crossroads connected by one-way roads. At each crossroad, none to several roads marked as "North," "East," "South," or "West" start from that crossroad. When the traveler comes to a crossroad, he chooses one of the roads that are marked as noted in his traveling plan, and goes to the next crossroad by this road. Because the signs are fuzzy and not consistent, several strange cases may occur:

- Only the roads that start from the crossroads are marked; those that end in the crossroads are not. The traveler can go from crossroads only by the marked (one-way starting) roads.
- None to several different roads may be marked by the same sign.
- If the traveler comes from the crossroad A to the crossroad B by the road marked as "North" at A, it does not mean that he can go back from B to A by a road marked "South;" there may exist a road from B to A marked by another sign, or there may be no marked roads from B to A.
- A marked road from the crossroad A may lead to the same crossroad.

For example, the following table defines a road map in Fuzzyland.

Crossroad	Marked roads that go from the crossroad to the crossroad#:			
	North	East	South	West
1	2, 3		1	
2		4		3
3	5	2, 4		
4				5
5	6		1	
6				

In this example, the traveler at the crossroad 1 can chose one of the North roads and get either to the crossroad 2 or to the crossroad 3, or go South and get to the crossroads 1; no other options are possible at the crossroad 1. Or, from the crossroad 5, he can go either North to the crossroad 6, or South to the crossroad 1.

The traveler has got several traveling plans. Each plan is a sequence of letters 'N', 'E', 'S', and 'W' that denote the direction that the traveler should choose at the crossroad he is currently being at. A plan is considered correct if there exists a path of exactly the same length as the sequence, from the starting to the destination crossroad that follows the plan; otherwise the plan is incorrect. Note that there may be none, one, or several such paths for the same plan. For the given example, if the traveler starts from the crossroad 1 and wants to get to the crossroad 6, the sequence "SSSNEEWN" is correct because there is the path: 1(start), 1, 1, 1, 3, 2, 4, 5, 6(destination) of the length 8 that leads to the destination. The sequence "NEWNS" in that case is not correct; the traveler can follow the paths: 1(start), 2, 4, 5, 6, and then he cannot go South (note that this path is not correct because it is not of the same length as the sequence), or: 1(start), 3, 2, 3, 5, 1, (the ending crossroad is not the desired destination), or: 1(start), 3, 4, 5, 6 (the same as for the first option).

Your program should find out which of the given plans are, and which are not correct for the given road map, starting, and destination crossroads.

The program input file contains several data sets. Each data set represents one road map with the starting and destination crossroads, and several given plans. There is exactly one empty line between successive data sets.

Each data set begins with one line containing an integer that represents the number of crossroads. Zero denotes the end of the input file. The crossroads are numbered starting from 1. Then follows a line with two integers separated by blanks that define the starting and the destination crossroads. Then follows a set of lines, four lines for each crossroad. These four lines define the adjacent crossroads (separated by blanks) that can be directly reached from the given crossroad by the roads marked with: North (first line), East (second line), South (third line), and West (fourth line). If a line contains a single 0, that means there are no roads marked with this sign starting from the given crossroad.

Then follows a line with a single integer NP that denotes the number of given plans. The next NP lines contain the sequences (one sequence per line). The plan consists of the letters 'N', 'E', 'S', or 'W'. The maximum number of crossroads is 30, and the maximum plan length is 80.

For each given data set, the program prints a line with the data set header (the sentence "Fuzzyland #:"), and then one line for each given plan with the answer 'YES' if the sequence is correct, or 'NO' otherwise.

Input Sample

```
6
1 6
2 3
0
1
0
0
4
0
3
5
2 4
0
0
0
0
0
5
6
0
1
0
0
0
0
0
3
SSSNEEWN
NEWNS
SNEWSNEWN

7
1 7
2
0
6
0
0
3
0
5
2
0
0
4
0
4 5
6
0
0
0
4
5 6
0
7
0
2
0
0
0
0
0
3
NEWENSWSSN
SSWN
SNN

0
```

Output for the Input Sample

```
Fuzzyland #1:
YES
NO
YES

Fuzzyland #2:
YES
NO
NO
```