

Alice in Amsterdam, I mean Wonderland

This is a fact that not many people know, but Alice lived in Amsterdam. Yes, there you go. And as many kids of the time, she used to go for some medicinal mushrooms to the drug store (the pharmacies, the coffee shops didn't exist yet). Usually, between the regular mushrooms, a magical one could be found, producing as expected, deep and vivid hallucinations. During one of those hallucinations, Alice was transported to a "wonderland", where many weird things happened. One thing was particularly dazzling for her: even though everything looked pretty familiar, the distance between monuments of the city was sometimes negative!!! Although, zero distance between two different monuments means a direct path doesn't exist. A loop from a given monument right back to it can be of length zero (with means that it can be reached instantly like in real life) or negative, like for regular paths. Alice also thought that she saw some positive distances for loops, but we should treat those cases as zero distance.

Now, as a very smart girl as she is, she figured out a way to find the shortest path between any two monuments. Unfortunately, as expected, Alice forgot it when she got sober again. She was only able to remember that, in some cases, she could get stuck in a cycle path with negative distance. In such cases, there will always be a cheaper path to get to the same monument. This was one of the few things that had perfect sense for her: Your shortest path will be shorter if you take that cycle again and again, to infinity. Alice, has been trying to figure optimal distances all over again, but she can't. She doesn't want to trip again, she has been clean for longer than a year (good for her!!). Would you be so kind to help her?

Given a list of monuments in a city, and their relative distances, find the shortest paths between some pairs of monuments.

Input

Each case, starts with one line containing **N**, the number of monuments in the test case ($1 \leq N \leq 100$). Next **N** lines will each contain one string **K** and **N** integers **K_j**, separated by single spaces. **K** is a name of a monument and will consist of at most 20 alphanumeric characters. Each integer **K_j** ($0 \leq j < N$) in line **i** describes the distance from monument **i** to **j** ($-2^{30} \leq K_j \leq 2^{30}$). Next line will contain a single integer **Q** ($1 \leq Q \leq N^2$). It will be followed by **Q** lines, each with a pair of integers (**U**, **V**), indicating the start and destination monument for the path that is queried ($0 \leq U, V < N$).

End of the input is indicated by a test case with **N** = 0 and should not be processed.

Output

For each test case, print a line "Case #**tc**:" (without quotes), where **tc** is the case number, starting from 1. Next **Q** lines should describe query results. If the optimal distance can be infinitely small, print only "NEGATIVE CYCLE". In other cases, start the line with "**start_name-destination_name**" followed by the actual result. If the destination can't be reached, print "NOT REACHABLE", otherwise print the integer distance.

Example

Input:

2
Nieuwkerk -1 1
Oudekerk 1 0
4
0 0
0 1
1 1
1 0
3
Nieuwkerk 0 -5 0
Oudekerk 10 0 0
Pierdeck -100 -100 0
9
0 0
0 1
0 2
1 0
1 1
1 2
2 0
2 1
2 2
0

Output:

Case #1:
NEGATIVE CYCLE
NEGATIVE CYCLE
NEGATIVE CYCLE
NEGATIVE CYCLE
Case #2:
Nieuwkerk-Nieuwkerk 0
Nieuwkerk-Oudekerk -5
Nieuwkerk-Pierdeck NOT REACHABLE
Oudekerk-Nieuwkerk 10
Oudekerk-Oudekerk 0
Oudekerk-Pierdeck NOT REACHABLE
Pierdeck-Nieuwkerk -100
Pierdeck-Oudekerk -105
Pierdeck-Pierdeck 0