H
Assembly line

The last worker in a production line at the factory of Automated Composed Machinery is worried. She knows that her job hangs in the balance unless her productivity increases. Her work consists of assembling a set of pieces in a given sequence, but the time spent on assembling pieces a and b and then c may not the same as that on assembling pieces b and c, and then assembling a with the resulting component. Only two consecutive pieces may be assembled at a time, and once they are assembled they behave as another piece in terms of the time needed for further assembly.

In order to aid her, you need to find the optimal way to assemble all components. The input to your program will be a set of symbols representing (types of) pieces, and a so-called assembly table representing the time it takes to assemble them, as well as the type of the resulting component. For instance, we may have two symbols $\{a, b\}$, and the following table:

	a	b
a	3-b	5-b
b	6-a	2-b

This means, for example, that two pieces of type a and a may assembled in 3 minutes, and the result is a component of type b, in that the time required to assemble it again with another piece of, say, type a is 6 minutes, and so on. Note that the table is not symmetric, i.e. assembling b and a may be more time-consuming than a and b.

For a sequence of components labelled aba, the two possible solutions are:

- $(ab)a = ba = a$ with time $time(ab) + time(ba) = 5 + 6 = 11$.
- $a(ba) = aa = b$ with time $time(ba) + time(aa) = 6 + 3 = 9$.

So the result for this case would be a piece of type b in 9 minutes (denoted 9-b).

Input

The input consists of several test cases. Each test case begins with a line containing a natural number k ($1 \leq k \leq 26$), followed by a line with k symbols (characters in `[a-z]`) separated by spaces. The following k lines contain the assembly table: the i-th line has k pairs of the form *time*-*result*, where *time* is an integer between 0 and 1 000 000 inclusive, and *result* a symbol belonging to the preceding set. The j-th pair in the i-th line represents the time to compose pieces of types represented by the i-th and j-th symbols, along with the type of the resulting piece. After the table, a line with an integer n indicates the number of lines that follow, each line being a string of at most 200 symbols. Each of these lines is a sequence of components that need to be assembled together in the right order.

The input will finish with a line containing `0`, which should not be processed.

Output

For each test case, print n lines, each with an integer *time* and a symbol *result* in the format *time*-*result*. Each line represents the minimum time and the type of the resulting piece for the corresponding case in the input. In case of a tie among several possible results with the same minimum time, choose from among those the piece whose type letter appears first in the line that contained the k symbols at the beginning of the test case. (For example, if that line was $a\ c\ b$ and both c and b can be obtained with minimum cost 5, print 5-c).

There must be an empty line between the output of different test cases.

Sample Input

```
2
a b
3-b 5-b
6-a 2-b
2
aba
bba
2
m e
5-e 4-m
3-e 4-m
1
eme
0
```

Sample Output

```
9-b
8-a

7-m
```