

# String Equations

**Input:** stringeq.in

**Output:** stringeq.out

We all understand equations such as:

$$3 + 8 = 4 + 7$$

But what happens if we look at equations with strings instead of numbers? What would addition and equality mean?

Given two strings  $x$  and  $y$ , we define  $x + y$  to be the concatenation of the two strings. We also define  $x = y$  to mean that  $x$  is an anagram of  $y$ . That is, the characters in  $x$  can be permuted to form  $y$ .

You are given  $n$  distinct nonempty strings, each containing at most 10 lowercase characters. You may also assume that at most 10 distinct characters appear in all the strings. You need to determine if you can choose strings to put on both sides of an equation such that the "sums" on each side are "equal" (by our definitions above). You may use each string on either side 0 or more times, but no string may be used on both sides.

## Input

The input consists of a number of cases. Each case starts with a line containing the integer  $n$  ( $2 \leq n \leq 100$ ). The next  $n$  lines contain the  $n$  strings. The input is terminated with  $n = 0$ .

## Output

For each case, print either "yes" or "no" on one line indicating whether it is possible to form an equation as described above. If it is possible, print on each of the next  $n$  lines how many times each string is used, with the strings listed in the same order as the input. On each line, print the string, followed by a space, followed by the letter "L", "R", or "N" indicating whether the string appears on the left side, the right side, or neither side in the equation. Finally, this is followed by a space and an integer indicating how many times the string appears in the equation. Each numeric output should fit in a 64-bit integer.

If there are multiple solutions, any solution is acceptable.

## Sample input

```
2
hello
world
7
i
am
lord
voldemort
tom
marvolo
riddle
0
```

## Output for sample input

```
no
yes
i L 1
am L 1
lord L 1
voldemort L 1
tom R 1
marvolo R 1
riddle R 1
```