

## 3843 Expression

Expressions are formalized in Lambda Calculus. Two expressions are said to be equivalent, if for each combination of constants/values for variables, they have the same output.

Define a binary operation ( $*$ ) on a set  $S$ . The set  $S$  contains letters from  $\{a, b, \dots, p\}$ . The operation  $x * y$ ,  $x, y \in S$ , is defined by an operation table, where values of operands  $x$  and  $y$  appear in lexicographic order. The table is illustrated below for an operation on  $S = \{a, b, c\}$ . The table indicates that  $a * a = b$ ,  $b * a = a$ ,  $a * b = c$ , and so on.

$x*y$	a	b	c
a	b	c	c
b	a	a	b
c	a	b	c

Consider a string  $s = "s_1s_2 \dots s_n"$  of letters in  $S$  and/or variables denoted by  $x$  and/or  $y$ . Each variable may assume a value equal to a letter in  $S$ . Define the  $LR$  expression of  $s$ , where letters in  $s$  are processed from left to right,

$$LR(s) = (\dots (((s_1 * s_2) * s_3) * s_4) * \dots s_n).$$

Similarly, define the  $RL$  expression of  $s$ , where letters are processed from right to left,

$$RL(s) = (s_1 \dots * (s_{n-3} * (s_{n-2} * (s_{n-1} * s_n)))) \dots).$$

Given the set  $S$ , the operation table defining the operation  $*$  and a number of strings each containing not more than two variables, you are required to write a program that determines the equivalence of  $LR$  and  $RL$  expressions for each string.

### Input

The input may contain multiple test cases. For each case there are three input lines.

The first line gives an integer  $n$  and a string. The integer  $n$  represents the total number of letters in  $S$  and the string identifies the letters in  $S$ , in lexicographic order.

The second line gives  $n$  strings each of length  $n$ . The strings define, in order, the entries in the operation table appearing in rows  $1, 2, \dots, n$ .

The third line gives an integer  $m$  and a sequence of  $m$  strings. Each string in the sequence is composed of letters in  $S$  and/or at most two variables  $x, y$ . Assume that  $n$  and  $m$  are less than 10.

A line, containing a zero '0' as the first character, follows the last test case.

### Output

For each string in the third input line of a test case there is one output line. The line gives the total number  $k$  of equivalent  $LR$  and  $RL$  expressions in the given string and identifies the  $k$  strings in lexicographic order.

**Sample Input**

```
3 abc
bcc aab abc
5 abc bbb xx axby xay
2 ab
aa ab
1 axaxbbx
0
```

**Sample Output**

```
1 abc
0
3 aa bb cc
3 aabb aabc abbc
3 caa cab cac
2 aaaabba ababbbb
```