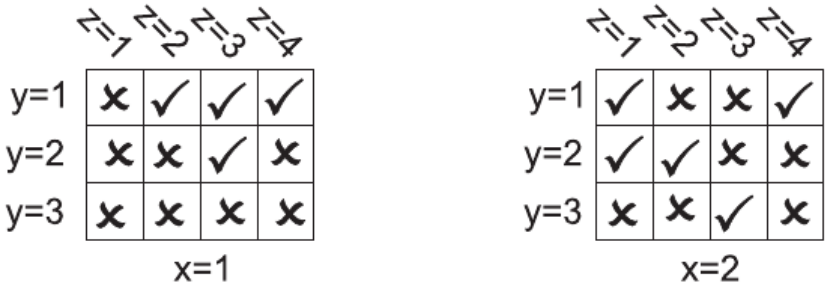
ACM-ICPC Live Archive

3250 One for all. All for there exists.

Given a boolean function of N variables, write a program to determine the validity of queries involving universal (for-all) and existential (there-exists) quantifications.

Consider, for example, the following boolean function of three variables x, y, and z:



Here's a sample of queries to answer:

- $\forall x\ \forall y\ \forall z$

  answer is false, since not every combination is true.

- $\exists y\ \forall z\ \exists x$

  Answer is true. For $y = 1$, we have a value for x giving a true result for every value of z: $< z = 1, x = 2 >$ and $< z = 2, x = 1 >$ and $< z = 3, x = 1 >$ and $< z = 4, x = 1|2 >$

- $\exists x\ \forall z\ \exists y$

  Answer is true for $x = 2$.

Input

Your program will be tested on a number of test cases. Each test case is specified using $Q + 3$ lines.

The first two lines of a test case specify the boolean function while the remaining lines specify the queries for that test case. The first line includes $N + 1$ positive numbers. The first number is N which is the number of variables. The remaining N numbers are $D_1, D_2, \ldots, D_N$ where D_i is the number of values for v_i (the i-th variable.) The second line is made of $D_1 * D_2 * \cdots * D_N$ '`T`' or '`F`' characters (true or false) specifying the values of the function, in row major order.

The third line includes a single positive value Q representing the number of queries. Each query is specified on a separate line using N pairs, one for each variable (all queries will involve all variables.) Each pair is described using two parts: The first part represents the quantification and is one of two letters: '`A`' (for all) and '`E`' (there exists.) The second part is a number representing the variable number. For example, the query '`A2E1`' stands for $\forall v_2\ \exists v_1$

Consecutive test cases are separated by a single blank line. The end of test cases is specified using a dummy test case with $N = 0$.

The number of variables is less than 10. The number of values for any variable will not exceed 10. However, $D_1 * D_2 * \cdots * D_N$ will not exceed 1,000,000.

The first test case in the Sample I/O section represents the example used in the problem description.

Output

For each test case, print the following line:

`Test case #` k

where k is the test case number (starting at 1). For each query, print the following line:

`  Query #` k `is` r`.`

where k is the query number (starting at 1) and r is the result of the query ('`true`' or '`false`'.)

Sample Input

```
3 2 3 4
FTTTFFTFFFFFTFFTTTFFFFTF
3
A1A2A3
E2A3E1
E1A3E2

2 2 2
TFTF
6
A2A1
A2E1
A1E2
E1A2
E2A1
E1E2

0
```

Sample Output

```
Test case # 1
  Query # 1 is false.
  Query # 2 is true.
  Query # 3 is true.

Test case # 2
  Query # 1 is false.
  Query # 2 is false.
  Query # 3 is true.
  Query # 4 is false.
  Query # 5 is true.
  Query # 6 is true.
```