UCF "Practice" Local Contest — August 27, 2011

Ternary Logic (filename: maybe)

Miles is an expert at Boolean logic, which uses the two values TRUE and FALSE and the fundamental operators NOT, AND, and OR. However, to understand his friend Maisy's way of thinking, he realized that he needs to understand her slightly more sophisticated form of logic, which has the additional value MAYBE. Maisy's ternary logic system has the same operators, and they yield the same results when applied to any combination of TRUE and FALSE values. So, Miles only needs to learn the following:

```
NOT operator:
     NOT MAYBE = MAYBE

AND operator:
     FALSE AND MAYBE = FALSE
     TRUE  AND MAYBE = MAYBE
     MAYBE AND MAYBE = MAYBE

OR operator:
     TRUE  OR MAYBE = TRUE
     FALSE OR MAYBE = MAYBE
     MAYBE OR MAYBE = MAYBE
```

Miles has noticed in some situations that Maisy says "maybe" quite often, and she seems to ignore conditions that Miles thinks should change her mind. In order to predict those situations, he develops expressions using her ternary logic, and starts to evaluate all possible outcomes of each expression in order to count the number of MAYBE results. Note that different results are possible for a given expression since any variable in the expression can take on the value TRUE, FALSE or MAYBE. Note, however, that when the same variable occurs more than once in the same expression, all copies of that variable take on the same value (i.e., all TRUE, all FALSE, all MAYBE).

The Problem:

Write a program to help Miles count the number of MAYBE results that are possible from a ternary logic expression.

The Input:

The first input line will contain a single integer n. The next n lines will each contain 1 to 40 characters which form a single logical expression. Each input expression will consist of the following characters only:

```
0 (zero) represents FALSE
1 (one) represents MAYBE
2 (two) represents TRUE
A, B, C, D, E, F, G (uppercase letters) are variables which can
     take on any ternary value
& (ampersand) represents AND
+ (plus) represents OR
! (exclamation) represents NOT, and always appears in front of
     what it operates on
```

Each expression is guaranteed to be well-formed. Precedence of operations is NOT, then AND, then OR. The associativity for each operator is left-to-right. There are no parentheses nor other grouping of sub-expressions within the expression.

The Output:

Print a heading for each input expression followed by the expression. Then print the number of MAYBE results that are possible from the expression. Leave a blank line after the output for each data set. Follow the format illustrated in Sample Output.

(Sample Input/Output on the next page)

Sample Input:

```
5
A&B
A+B+C
A&!B+B&!A
!A&A
2+A&!1
```

Sample Output:

```
Expression #1: A&B
3 MAYBE result(s) possible

Expression #2: A+B+C
7 MAYBE result(s) possible

Expression #3: A&!B+B&!A
5 MAYBE result(s) possible

Expression #4: !A&A
1 MAYBE result(s) possible

Expression #5: 2+A&!1
0 MAYBE result(s) possible
```