10437 Playing With Fraction

Sometimes it may be found that some numbers if represented in normal integer or floating point value will not be precise enough for some calculation. For example some value like 10/3 if stored as integer the value stored will be 3. Again if stored in floating point value it will be 3.3333 (plus some extra digits of precision). Multiplying it with 3 again will give 9 and 9.9999 respectively. If we could keep the number as 10/3 , as we do in our hand calculation (fractional arithmetic) then the problem can be solved. The object oriented programming capability of newer computer languages gave us the power of doing such things as other normal variables like integer.

Today's scientific calculators even can do such calculations using fractional numbers. This problem also requires you to evaluate some expressions containing, fractional numbers. The expression may contain arbitrarily deep brackets. The expression may have operators '`+`', '`-`', '`*`' and '`/`' representing corresponding operations. The operators '`*`' and '`/`' have same precedence and have greater precedence then '`+`' and '`-`' which also have same precedence among themselves. Operators having same precedence are evaluated from left to right. Moreover brackets may override the precedence. No invalid expression will be present. You have to process each such expression.

Input

Each line contains an expression. Arbitrary number of spaces may be present in the expression. A fractional number is represented by '$N|D$' where N and D are positive 32 bit integers. The integer D may not be present. Each line will not contain more then 500 characters. Input is terminated by end of file.

Output

For each line of expression process the evaluated number reduced to their normal form. A normal form of such number may be defined as having no common divisor in numerator and denominator. If the number is invalid (i.e. denominator is 0) print '`INVALID`.

Don't print the denominator if it is unnecessary.

Sample Input

```
3|2+3
(5|3-2|3)*1|3
```

Sample Output

```
9|2
1|3
```