Binomial String Expansion

One of many cases where algebra effectively applies the tools of combinatorics is binomial expansion. A trivial example is an expansion of the expression $(x+y)^4$ that can be written as

$$(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4.$$

Taking this process one step further into abstraction, we can come up with a general formula for computing the expansion coefficients such as

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k},$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Admittedly, it is rather hard to calculate binomial coefficients by hand. With this in mind, it would be a good idea to create a computer program that can expand any power of a sum of two variables into a sum of powers. To make it even more interesting, let us digress a little from the usual mathematical notation. For the sake of this problem, we will assume that a variable can be an entire lower-case word. If the computer is to do the work anyway, why stick with just single-letter variables?

Input

The number of test cases is given by T, such that $T \leq 100$, on the first line of the input. After that, T test cases follow. Each test case is provided on a single line formatted as `(a+b)^k`. In this format, `a` and `b` stand for variables. For this problem, we will just consider strings representing variables consisting of only `a-z` letters. The exponent, given by k, where $1 \leq k \leq 50$, is a power that you need to raise the sum. The power operation is denoted by the standard "xor" character. You can safely assume that there are no lines longer than 100 characters.

Output

For each test case, output a single line formatted as `Case N: S`. Where N is the test number (starting from 1) and S is an expanded expression. For more details, see sample test cases provided below.

Input sample

```
4
(a+b)^1
(something+nothing)^2
(sample+case)^3
(last+one)^4
```

Output sample

```
Case 1:  a+b
Case 2:  something^2+2*something*nothing+nothing^2
Case 3:  sample^3+3*sample^2*case+3*sample*case^2+case^3
Case 4:  last^4+4*last^3*one+6*last^2*one^2+4*last*one^3+one^4
```