# One more symbol

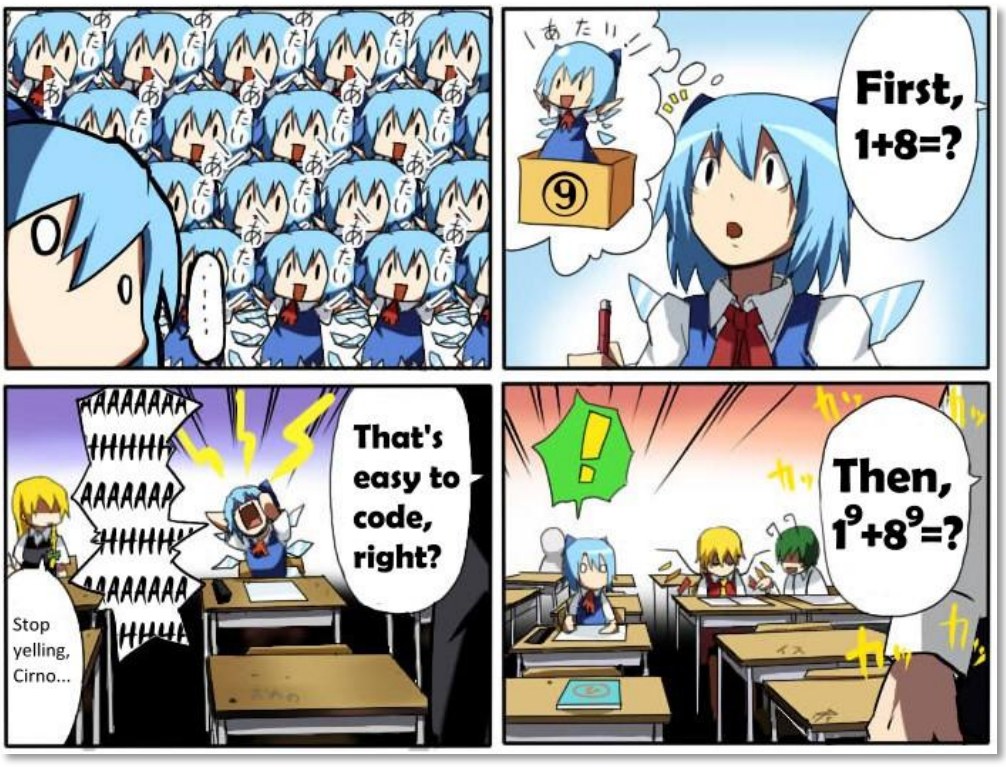
## Problem Statement

After last tutorial, Cirno has learnt how to implement infix calculation with + and *. And she thought she has mastered infix calculation, she feels she is the strongest. But teacher Kamishirasawa knows that Cirno only barely understand the material. So teacher Kamishirasawa gives a problem to test the class:

“Everyone knows how to implement + * calculation following arithmetic order. What if I add one more symbol, ^, which mean ‘power’, do you all know how to implement? Relax; I will reduce the problem into an easier version. Listen carefully.

1. For all the formula given, all numbers are single digit only, 0 to 9.
2. For ‘^’ (power), 2^3 means $2^3$ = (8)
3. There is NO consecutive ^, so you will NOT see 2^2^2.
4. + and * are not part of the power. i.e. 2^3*4 = $2^3$ *4 = 32 , NOT $2^{3*4}$
5. All the intermediate step and the result will not exceed signed 32bit integer.
6. I think everyone knows x^0 = 1, right?
7. Of course, you need to follow arithmetic order. For example: 1+2^3*4 = 33 = 4*2^3+1. You can check with the calculator in Windows.

So I guess it’s an easy, right? Please write a piece of code to handle this task.”



Obviously, Cirno doesn’t know how to solve it. She is afraid teacher Kamishirasawa will give her a headbutt if she didn’t answer it right. She asks you to teach her how to solve that. So you decide to write a program to teach her.

## Input

For each testcase, it consists of one single line string, which is the formula. The length of the formula is less than or equal to 100. The formula is always valid, and it contains digits(0-9), +, * and ^ only.

## Output

For each given data set, print the answer of that formula.

## Sample

| Input | Output |
| --- | --- |
| 1+1 | 2 |
| 1*2 | 2 |
| 2^3 | 8 |
| 1+2*3 | 7 |
| 1^2+3 | 4 |
| 0^0 | 1 |