# Problem G: Magic Calculator

Time Limit: 1 Sec  Memory Limit: 128 MB
Submit: 171  Solved: 27
[Submit (submitpage.php?cid=1056&pid=6&langmask=0)][Status (problemstatus.php?id=1268)][Web Board (bbs.php?pid=1268&cid=1056)]

## Description

IceRuler has a magic calculator. It can calculate the answers for expressions, with constants only containing integers in the range 0-9 and operators only appearing in the following table.

| Priority Level | Operator | Name | Combination Law |
|---|---|---|---|
| 1 | () | Bracket | |
| 2 | + | Positive Sign | Right to left |
| | - | Negative Sign | |
| | ~ | Bitwise Not | |
| 3 | * | Multiplication | Left to Right |
| 4 | + | Addition | Left to Right |
| | - | Subtraction | Left to Right |
| 5 | & | Bitwise And | Left to Right |
| 6 | ^ | Bitwise Xor | Left to Right |
| 7 | \| | Bitwise Or | Left to Right |

   Operators will be operated following their priority levels. For example, in 1+2&3, addition will be calculated first, then bitwise and. For 1-+2,  positive sign will be calculated first, then subtraction.
   Operators with same priority level will be operated following the combination law. For example, in 1+2-3, the combination law of  addition and subtraction is "Left to Right", addition should be calculated first, then subtraction. In -~1 , bitwisenot will be calculated first, then negative sign.
   The expressions uses 32-bits signed integer type in the calculate process. You do not need to consider type conversion and overflow processing. For example, the result of ~1 should be -2, and the result of multiplying 2 by 32 times should be 0.
   Please write a program to simulate the IceRuler's calculator. Given expressions, you are required to return the results of this calculator.

## Input

Integer T in one line means testcases (1≤T≤1000000). For each case , you are given an expression in one line, and you should print its result in one line. The length of each expression is not exceed 50.

## Output

You should print result of each expression in one line.

## Sample Input

```
3
3&(2|1)
+-~3
2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2
```

## Sample Output

```
3
4
0
```

## HINT