Problem A: Single Digit Adder

Write a program that can evaluate expressions from the following roughly BNF (Backus Naur Form) grammar:

```
expr ::=  term | expr '+' term | expr '-' term
unary_op ::= '+' term | '-' term
term ::= '(' expr ')' | '(' unary_op ')' | literal
literal ::= [0-9]
```

There will be no whitespace within an expression. All expressions will consist solely of the characters (,), +, -, and the digits 0 through 9. You may assume that all input is well-formed.

Input (from file: a.in)

The input will consist of one expression per line followed by a newline. There will be no blank lines in the file.

Output (to stdout)

For each expression, output its integer value, followed by a single newline.

Sample Input

```
1
(-2)+3
(1-(2+3))
(1-2+3)
(1-(+(2-3)))
```

Sample Output

```
1
1
-4
2
2
```