RPN Calculator
Problem ID: rpn

Your goal is to write a postfix ("Reverse Polish Notation") calculator. Unlike the probably more familiar "infix" notation, in postfix notation the numbers are given first, then the desired operation. The following example is from Wikipedia. The infix notation

$$((15/(7-(1+1)))*3)-(2+(1+1))$$

can be written as

$$15 \quad 7 \quad 1 \quad 1 \quad + \quad - \quad / \quad 3 \quad * \quad 2 \quad 1 \quad 1 \quad + \quad + \quad -$$

It can be evaluated by what Wikipedia calls the "left to right" algorithm as follows:

```
15 7 1 1 + - / 3 * 2 1 1 + + - =
15 7     2 - / 3 * 2 1 1 + + - =
15         5 / 3 * 2 1 1 + + - =
             3 3 * 2 1 1 + + - =
                 9 2 1 1 + + - =
                 9 2     2 + - =
                 9         4 - =
                             5
```

Your program will read and process lines of input one at a time, until it reads `quit` on a line by itself. Each line of input will either be an integer, or a string representing an operation. If the input is a number, your program should save it on a stack. If the input is an arithmetic operation, your program should remove the two most recently saved numbers, and replace them with the result of the operation. If before the operation, the most recently added number (i.e. the top of the stack) is b, and the second most recently added number is a, you should replace them with

operation	replacement
$+$	$a+b$
$-$	$a-b$
$/$	$\lfloor a/b \rfloor$
$*$	$a*b$
^	a^b

The non-arithmetic operations should be implemented as follows:

dup	copy the top element of the stack
print	print the top element of the stack
pop	remove the top element of the stack
swap	interchange the top two elements on the stack
quit	Stop reading and processing input

Input

Each line of input will either be an integer, or a string $+, -, *, /,$ ^, `dup`, `pop`, `print`, `quit`, or `swap`. You may assume that there will always be sufficient arguments for each operation (i.e. there is no "stack underflow"), and that no division by zero is attempted.

Output

The output should be the output from the print operations in the input (if any).

Sample Input 1

```
15
7
1
1
+
-
/
3
*
2
1
1
+
+
-
print
quit
```

Sample Output 1

```
5
```

Sample Input 2

```
17
3
2
*
print
dup
dup
pop
+
3
/
print
swap
-
13
+
2
^
print
quit
```

Sample Output 2

```
6
4
0
```

Sample Input 3

```
2
128
^
3
/
-2
*
print
quit
```

Sample Output 3

```
-226854911280625642308916404954512140970
```