AMERICAN COMPUTER SCIENCE LEAGUE

2018-2019 — Contest #4

Senior Division - Prefix Evaluation

PROBLEM: Evaluate a prefix expression. The operands in the expression are integers between -1,000 and 1,000, exclusive. The operators are the unary operator absolute value (|); the binary operators addition (+), subtraction (-), and multiplication (*); and the trinary operators "switcher" (@) and "max" (>). The @ operator of *a*, *b*, and *c* returns *b* when *a* is positive; otherwise, it returns *c*. The > operator returns the largest of its 3 operands.

Each line of data is valid prefix expression with at least one space separating all operands and operators.

Example 1: `* + 4 5 - 3 -1` simplifies to `* 9 4`, which has a value of 36.

Example 2: `@ - 8 9 82 46` simplifies to `@ -1 82 46`, which has a value of 46.

Example 3: `@ | - -8 10 82 46` simplifies to `@ | -18 82 46`,
which simplifies to `@ 18 82 46`, which has a value of 82.

Example 4: `+ > 8 * 2 7 9 6` simplifies to `+ > 8 14 9 6`,
which simplifies to simplifies to `+ 14 6`, which has a value of 20.

INPUT: Five lines of data. Each line is a string, <= 128 characters, representing a valid prefix expression with operands and operators as described above. At least one space will separate operands and operators.

OUTPUT: Evaluate each prefix expression and print the answer.

SAMPLE INPUT (http://www.datafiles.acsl.org/2019/contest3/sr-sample-input.txt):

```
* + 4 5 - 3 -1
@ - 8 9 82 46
@ | - -8 10 82 46
+ > 8 * 2 7 9 6
| * @ - 1 6 34 12 > - 990 1000 * -2 3 + -51 49
```

SAMPLE OUTPUT:

#1. 36
#2. 46
#3. 82
#4. 20
#5. 24