AMERICAN COMPUTER SCIENCE LEAGUE

2018-2019 **Contest #4**

Intermediate Division - Prefix Evaluation

PROBLEM: Evaluate a prefix expression. The operands in the expression are single digit whole numbers. The operators are binary addition (+), subtraction (-), and multiplication (*); the trinary operator "switcher" (@); and the trinary operator "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 a valid prefix expression with operands and operators separated by at least one space.

Example 1: `* + 4 5 - 3 1` simplifies to `* 9 2`, which has a value of 18.

Example 2: `@ - 8 9 7 6` simplifies to `@ -1 7 6`, which has a value of 6.

Example 3: `+ > 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/contest4/int-sample-input.txt):

```
* + 4 5 - 3 1
@ - 8 9 7 6
+ > 8 * 2 7 9 6
- @ - 3 5 7 * 2 4 > * 4 6 * 3 7 * 9 3
* 7 - + 4 6 > 0 - 2 3 1
```

SAMPLE OUTPUT:

```
#1.  18
#2.  6
#3.  20
#4.   -19
#5   63
```