**XX. Lombdos**

# Program Name: lombdos.java Input File: lombdos.dat

Ral is lazy, and waited until the night before 7 lakes to review lambdas for the written test. Sadly, his teammate hates him so he gave him a faulty review guide. Instead of a lambda review guide he was given a review guide for the offbrand version “lombdos”. All lombdos follow a few simple rules. Help Ral evaluate some lombdos so he can figure out how they work.

* All lombdos operate on 32 bit signed numbers (java ints)
* All lomdos take exactly two inputs, and return 1 output
  + All inputs are uppercase letters
* The name of a lombdos is a sequence of special characters
  + A lombdos name will never be named +,-,\*, or /
* A lombdos can be made of variables, arithmatic operations and other lombdos
  + Division is always Integer division
  + There will never be division by zero
  + Lombdos definitions will only include lombdos that are defined before it
  + Lombdos definitions will never include themself
* All lombdos and arithmatic operations are evaluated from left to right (normal operator precedence is ignored)

**Input**

The first line consists of an integer n, the number of lombdos. The next n lines consist of lombdos definitions, one per line. Each lombdos will be of the format “Name = P1, P2 -> EXPR”. There will be spaces between all elements of the expresion. Next will be an integer m, the number of lombdos calls. The next m lines will consist of a integer, one lombdos, and another integer. The lombdos and its inputs will be seperated by spaces.

**Output**

Output the numeric result of each lombdos call.

**Example Input File**

4

\*\* = A, B -> A \* B

\*+ = X, Y -> X \* Y + Y

& = N, M -> N \*\* M / M

&& = E, R -> E & R \*+ E \*+ R

4

3 \*\* 5

3 \*+ 4

3 & 7

2 && 3

**Example Output to Screen**

15

16

3

21