Problem J3: Punchy

Problem Description
In the early days of computing, instructions had to be "punched" onto rectangular cards, one instruction per card. This card deck was then fed into a card reader so the program could be read and executed. Students put elastic bands around their card deck, and, often, carried their cards in a box for fear that they would become rearranged, and thus, their program would be incorrect.

Poor Bill though...he left his cards right near a window and the wind blew his neat deck of cards all over the place, and thus his program is out of order! Bill decides to pick up the cards in some random order and then execute the program.

Write a program to read and execute the commands in Bill's "new" program.

Input Specification
The programming language that Bill is using has only two variables (A and B) and seven different types of instructions.

Initially, the variables A and B contain the value 0.

There is one instruction per line. An instruction is an integer in the range $1...7$, possibly followed by a variable name, which in turn is possibly followed by either a number or a variable.

In all instructions below, the variable X or Y may refer to either A or B. The specific instructions are:

- $1\ X\ n$ means set the variable X to the integer value n;
- $2\ X$ means output the value of variable X to the screen;
- $3\ X\ Y$ means calculate $X + Y$ and store the value in variable X;
- $4\ X\ Y$ means calculate $X * Y$ and store the value in variable X;
- $5\ X\ Y$ means calculate $X - Y$ and store the value in variable X;
- $6\ X\ Y$ means calculate the quotient of X/Y and store the value in variable X as an integer, discarding the remainder.
- 7 means stop execution of this program.

You may assume that all division instructions do not cause a division by zero, and that all other operations (including instruction 1) do not cause the computed/stored value to be larger than 10,000 or smaller than -10,000.

(To clarify division of negative numbers, $-3/2$ and $3/-2$ both have quotient -1 and $-3/-2$ has quotient 1.)

Output Specification

Your program should output the value of the indicated variables, one integer per line, until the "stop" instruction has been read in, at which time your program should stop execution.

Sample Input (with output shown in *italics*)

```
1 A 3
1 B 4
2 B
4
2 A
3
3 A B
2 A
7
5 A A
2 A
0
2 B
4
7
```