## Problem 1 – Basic BASIC

### Do you know Elio? Elio is a young man who hates almost everything. One of the things he hates the most is all old programming languages. Recently he found an old diskette with strange code on it. And because he is very curious he wants to execute the code and view the result. He hates old programming languages, so he asks you to write a C# program to execute that code and find its output result. Since Elio is a really good-natured boy, he deserves your help.

That old language actually was a dialect of the well-known programming language BASIC. We will call it Basic BASIC (BB). So what are the important things about the BB language?

First of all, every single line of the BB code starts with an integer number – the unique identifier of the line. Every line has unique identifier bigger than all its preceding lines. One BB code line can contain only one command. After executing the command (and if no **GOTO** executed), then the code execution continues from the line with the smallest unique identifier which is bigger than the unique identifier of the current line. The code execution starts from the line with the smallest unique identifier.

The second important thing about the BB is that you can only use 5 variables. These variables are named **V**, **W**, **X**, **Y** and **Z**. They can only have integer values between -2 000 000 000 and 2 000 000 000, inclusive. The default value of these 5 variables is 0. In other words, if no value is preliminarily assigned to them their value will be 0.

Finally, the most important thing in the BB language is the commands. In BB there are 5 types of commands listed below:

1. Changing the value of a variable. You can assign values to any of these variables by simply using the assign (**=**) command. BB also has 2 arithmetic operations available – addition (**+**) and subtraction (**-**). Here are four examples for assigning values to variables: **V=-5**, **X=Y**, **W=X-Y**, **Z=Z+1**. Only one arithmetic operation is allowed with a single assigned command. Note that these lines are invalid: **V=X+Y+Z**, **W=1-4+3**, **X=-1-X**, **X=-2--3**.
2. Conditional command execution (**IF** <condition > **THEN** <command>). The conditions in the **IF** operator can only be comparisons: bigger than (**>**), less than (**<**) and equal to (**=**) between *variable and variable*, *variable and number* or *number and number*. The **IF** statement executes the command after the **THEN** statement if and only if the condition is true. For example the following BB code will assign value 5 to the **X** variable only if the **Y** variable is greater than the **Z** variable: **IF Y > Z THEN X = 5**. If the condition is false, then the code execution continues exactly from the next line ( the line after the **IF** command).
3. Unconditional branching (**GOTO**). The **GOTO** operator unconditionally jumps to a command line with the specified unique identifier. That line will always exist. For example the command **GOTO 10** will continue the code execution from a command with the unique identifier 10. **GOTO** “parameter” will be always a number, thus commands like **GOTO X** will be invalid.
4. Manipulating the output (**CLS** and **PRINT**).
   1. If the BB code reaches the **CLS** (clean screen) command, it automatically clears the content printed so far and starts printing from the beginning.
   2. The **PRINT** command gets the value of a variable and writes it in a single line in the output. For example **PRINT Z** will print the value of the **Z** variable on the output and then will write a new line.
5. Command for stopping the code execution – **STOP**. The **STOP** command immediately stops the execution of the BB code.

Note that the BB language **ignores all spaces** because they do not affect the semantics of the language.

### Input

The input data should be read from the console.

The input will consist of valid code written in Basic BASIC (BB), always ending with a line containing the word “**RUN**” (see the examples below).

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output data should be printed on the console.

You must write on the console the output result from executing the BB code given in the input.

### Constraints

* The given BB code will always be uppercased and valid in the syntax described above.
* Line identifiers will always be between 0 and 10 000, inclusive.
* It is guaranteed that the code will always reach its **STOP** command (or the end of the code) after no more than 1 000 000 command executions.
* None of the 5 variables (**V**, **W**, **X**, **Y** and **Z**) will have values smaller than -2 000 000 000 or bigger than 2 000 000 000 in any part of the code execution.
* Allowed working time for your program: 0.8 seconds. Allowed memory: 16 MB.

### Examples

|  |  |
| --- | --- |
| **Input example** | **Output example** |
| 5 X=-1  6 IF X=-1 THEN X=0  7 PRINT X  8 CLS  10 PRINT X  20 X=X+1  30 IF X < 4 THEN GOTO 10  40 STOP  50 PRINT X  RUN | 0  1  2  3 |
| 0 X = 1  1 Y = 2  2 Z = Y - X  5 PRINT X  6 PRINT Z  10 X = X +1  20 IF X = Y THEN GOTO 2  RUN | 1  1  2  0 |