

NOI '00 P2 - Program Parser

Time limit: 0.6s **Memory limit:** 16M

National Olympiad in Informatics, China, 2000

The *Tiny Basm* programming language (abbreviated the TB programming language) expressed in [Backus-Naur Form](#) (BNF) is:

```
<program> ::= <statement> [newline] { <statement> [newline] }
<statement> ::= <line-number> [space] <statement-body>
<statement-body> ::= <increment-statement> | <output-statement> | <branch-statement> |
<condition-statement> | <end-statement>
<increment-statement> ::= <variable> + <integer>
<output-statement> ::= <variable> ?
<branch-statement> ::= GO [space] <line-number>
<condition-statement> ::= IF [space] <variable> = <integer> [space] <branch-statement>
<end-statement> ::= END
<variable> ::= <letter>
<line-number> ::= <integer>
<integer> ::= <digit> { <digit> }
<letter> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
```

Note: `::=` indicates a definition. `|` denotes *or*. An expression enclosed in curly braces `{ }` indicates that it may occur 0 or more times. `[newline]` represents an end of line character (ASCII code 10). `[space]` represents a space character (ASCII code 32).

The following are examples of statements with incorrect formats (there will not be incorrect statements in the input):

```
10 A+1.5          (Does not comply with the increment-statement definition - the value is not an
integer)
20 A ?            (Does not comply with the output-statement definition - there is an extra
added space)
30 IF A=B GO 10    (Does not comply with the conditional-statement definition - <variable> =
<variable> is invalid)
```

Execution of TB programs:

- The process starts executing at the smallest line number; until a condition-statement is encountered, execution will be in ascending order of line number.
- All variables are initialized to 0 before the program begins execution.
- An increment-statement takes the variable in the statement and adds it to the integer, finally storing it back into the variable.
- An output-statement takes the value in the specified variable and displays it to the screen.
- When executing condition-statements, if and only if the variable and the integer immediately following it are equivalent, the branch-statement following it will be executed. All integers in branch-statements will be at most 4 digits long.

- After a branch-statement has been executed, the program will jump to the line with the line number following `GO`.
- After an end-statement has been executed, the entire program will terminate.
- Assume that the system will be able to process integers of any size, and overflows will not occur.

Please write a program that, given a program P in the TB programming language, finds the number of statements that will be executed upon running the program (whether or not a conditional-statement successfully branches, it will count as one executed statement).

Input Specification

The input will contain one program P in the TB programming language, with no more than 100 total lines of statements. No statement within P will exceed a length of 22 characters.

Within P 's branch-statements, there will always be a statement corresponding to the line number following `GO`.

P may contain many different end-statements.

Within P , the statement with the largest line number will always be an end-statement.

No line number in P will exceed 3 000.

The input will not necessarily give the lines of P in increasing order of line number.

Output Specification

The output should contain one line with one integer. If the program can properly terminate, output the number of executed statements. If the program cannot properly terminate, output `-1`.

Note: It is guaranteed that the answer will never exceed 5 million.

Sample Input

```
10 A+1
20 IF A=5 GO 60
60 END
30 A+2
40 A?
50 GO 20
```

Sample Output

```
11
```

Explanation

The order in which the lines are executed are: 10 → 20 → 30 → 40 → 50 → 20 → 30 → 40 → 50 → 20 → 60, for a total of 11 executed statements.