

5051 Conditional Statements

Bonjol is given a new task in his company. He is given a piece of code written in Pascal programming language for controlling a set of lights, and his task is to refactor the code and minimize its size while keeping the program logic exactly the same as before. The code is made up of several conditional statements. Each conditional statement is a line of the form

```
if  $\langle variable \rangle \langle comparison - operator \rangle \langle comparison - value \rangle$  then turnOn (  $\langle light - number \rangle$  );
```

where

- $\langle variable \rangle$ is a string of at most 255 lowercase English letters which is the name of a Pascal integer variable;
- $\langle comparison - operator \rangle$ is either “<”, “>”, or “=”;
- $\langle comparison - value \rangle$ is a 32-bit integer constant to which the $\langle variable \rangle$ is compared;
- $\langle light - number \rangle$ is another 32-bit integer constant which shows the number of the light which should be turned on if the condition “ $\langle variable \rangle \langle comparison - operator \rangle \langle comparison - value \rangle$ ” holds. (Nothing happens if the light is already turned on.)

Here is an example of such a code:

```
if a < 3 then turnOn( 5 );  
if bcq > -43 then turnOn( -117 );  
if cc = 0 then turnOn( 200 );
```

The only code modification which Bonjol is allowed is to delete a complete line. He wants to delete as many lines as possible such that the modified program remains completely equivalent to its original version. You are to help him and calculate the maximum number of lines which could be deleted.

Input

The input contains several test cases. Each test case starts with a line containing an integer n ($1 \leq n \leq 500$) which is the number of lines in the code. Each of the next n lines has a conditional statement. In each conditional statement, there is a single space after ‘if’, the $\langle variable \rangle$, the $\langle comparison - operator \rangle$, the $\langle comparison - value \rangle$, ‘then’, ‘turnOn(’, and the $\langle light - number \rangle$. The input terminates with a line containing ‘0’ which should not be processed as a test case.

Output

Write the rest of the i -th case, on the i -th line of output. You should just write one integer indicating the maximum number of lines that Bonjol can delete.

Sample Input

```
7
if aa < 4 then turnOn( 3 );
if aa > 7 then turnOn( 3 );
if aa = 7 then turnOn( 3 );
if aa = -1 then turnOn( 3 );
if aa < 7 then turnOn( 3 );
if b = 6 then turnOn( 4 );
if b = 6 then turnOn( 3 );
4
if pq = 6 then turnOn( 7 );
if pq = 6 then turnOn( 7 );
if pq = 6 then turnOn( 1 );
if pq = 2 then turnOn( 1 );
4
if pq = 6 then turnOn( 8 );
if pq = 6 then turnOn( 8 );
if pq = 6 then turnOn( 9 );
if pq < 8 then turnOn( 9 );
0
```

Sample Output

```
3
1
2
```