

Tomato Automata are small cool programs. You give them an infinite sequence of ones and zeros, and they give you a sequence of numbers. They are widely used in the Stanescu Operating System (SOS). They are written in Tomato Programming Language that is very simple. Here is its specification, tutorial and reference:

1. Tomato is a very simple but powerful language.
2. Lines in a Tomato program are numerated with integers from 1 to N ($N \leq 100000$) in the order they appear in the input.
3. There is exactly one command on a line.
4. The execution starts from the first line.
5. There are exactly five Tomato commands — `ifgo`, `jump`, `pass`, `loop`, and `die`.
6. When executed, each command prints its line number (into the output sequence).
7. The `ifgo` command has one argument - the line number of another instruction. It reads one bit from the input stream. If the bit is one, the execution jumps to the line with the given as argument line number. Otherwise the execution continues with the next line.
8. The `jump` command has one argument - the line number of another instruction. When executed, the execution jumps to the line with the given line number.
9. The `pass` command has no arguments. It does nothing (except printing its line number like all other commands). Then the execution continues with the next line.
10. The `die` command has no arguments. It terminates the execution of the program (printing its line number before that). The `die` command can not be used inside a loop.
11. The `loop` command is the only one with two arguments. It may be used to construct loops. The first argument is the starting line number $\langle line \rangle$ (less than the line number of the `loop` command), and the second is a positive integer $\langle count \rangle$. When executed, it loops from the start line a $\langle count \rangle - 1$ number of times (because it is already executed once). When the loop is executed the given number of times, the execution continues with the next line.
12. The `jump` and `ifgo` commands must be used only with line numbers in the scope of the innermost loop containing them (they can not jump outside of the innermost loop or inside loops nested in the innermost loop that does not contain the command).
13. The `loop` command can not be used to create overlapping loops. Nested loops must be strictly nested (they can not use the same starting line).
14. When the last line of the program is executed, the execution continues from the first except when the last line is `die` command.
15. White spaces may occur freely before or after the command name and their arguments.
16. The maximal length of a line in Tomato Programming Language is 80 characters including spaces.

Stanescu has lots of Tomato programs. He is interested in maximal length of output sequence that specific program can generate, where the length is the number of printed line numbers. Obviously, it is not possible to test each possible input sequence (of ones and zeros), so he needs a program that computes this.

Input

The input contains several programs, separated with an empty line. Each of them is a correct Tomato program.

Output

For each given program your solution has to print on a separate line the maximal length of the output sequence the program could generate. Print 'infinity' if there is no maximal length for the 9 output sequence. When finite, the maximal length will not exceed 10^9 .

Sample Input

```
Ifgo 2
loop 1 3
die

ifgo 2
ifgo 3
pass
die

    pass
        ifgo 4
        jump 5
        ifgo 3
    loop 2 2
    pass
loop 1 2
die
```

Sample Output

```
7
4
23
```