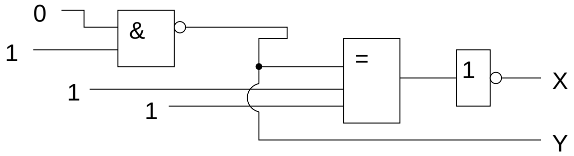
Gates of Logic

`logic.c | logic.C | logic.java | logic.p`

The Department of Computer Science and Engineering runs courses dealing not only with algorithms but also with computer hardware. One such introductory course explains basic principles of integrated circuits ("chips"), binary logic, boolean algebra, etc. As you may know, the very basic units of logical circuits are called *gates*. A gate is an element performing one simple logical operation. It can be connected to other gates using *lines*.



Logical circuits may be drawn as pictures with the gates represented as squares with inputs on the left and outputs on the right. In each square, there is a symbol that determines the gate type: Number **1** denotes an OR gate (its outputs are 0 if and only if there is no input with the value of 1), **&** is an AND gate (outputs are 1 if and only if there is no 0 input), and = is a XOR gate (outputs are 1 if and only if there is an odd number inputs that have the value of 1).

Your task is to scan such a "picture" and compute values of all named circuit outputs. The lines may split and join again but you may assume that each "value consumer" (input port of a gate or a named output) will be connected to exactly one "value source" (output port of a gate or an input value). There will be no feedback loops, i.e., there exists no cycle that would lead through the same gate twice.

Input Specification

The input contains several pictures. Each picture consists of at least one and at most 200 rows composed of the following characters:

- **Space** (" "). Empty space in the picture. Spaces are used to indent other characters to appropriate locations, because the exact position of characters is often important. Trailing spaces at the end of input rows may be present but may also be left out.
- **Dash** ("-"). Horizontal line. It connects characters on its left and right together, those characters will always exist and be able to "accept" the connection.
- **Pipe** ("|"). Vertical line, connects characters that are directly above and below. Like with the horizontal line, those characters will always accept the connection.
- **Plus sign** ("+"). Line connection or a bend. Connects characters on all four sides. All characters that are able to accept the connection are considered connected (there will always be at least two). However, there may be sides that contain a non-empty character that is not connected. For example, if a dash is present on a position directly below the plus sign, they are not considered connected.
- **Lowercase letter x** ("x"). Crossing of two lines without a connection. All four neighboring characters will accept the connection. The character above is connected to the one below and the character to the left with the one on the right, but there is no mutual connection between these two pairs.

- **Equal sign** ("=="). Represents an input or output port. It always connects characters on its left and right, at least one of these characters is the port. If there is a port on the left, it may only be a value source. If there is a port on the right, it may only be a value consumer.
- **Lowercase letter o** ("o"). Negation. There will always be a gate on the left and a port on the right of this character. It makes the particular gate output negated.
- **Hash mark** ("#"). Gate, which has always a rectangular shape with two vertical and two horizontal sides. The left vertical side may be connected to input ports, the right side to output ports (possibly negated). No two gates will touch each other's side, which means that any two vertically or horizontally neighboring hash marks are always parts of the same gate.

  The rectangle size will always be at least 3 characters in both directions, which means there is at least one character inside. All inner characters are empty (spaces), with exactly one exception. That single non-empty character denotes the gate type (note that it may have different meaning than outside the gate area) and will be a digit "one" ("1"), ampersand ("&"), or an equal sign ("=").
- **Binary digit** ("0" and "1"). Input value of the circuit. It is connected to the character on its right, which is always an equal sign.
- **Uppercase letter** ("A" through "Z"). Named output of the circuit. It accepts connection from its left, which is always an equal sign. Each letter will appear at most once, which means the number of circuit outputs is between 0 and 26, inclusive.

Each picture will be terminated by a row consisting solely of asterisk ("*") characters (at least one). The last picture will be followed by two such rows. No row in the input will be longer than 200 characters.

Output Specification

For each picture, print the values of all named outputs, sorted alphabetically. Each output row should contain three characters: output name (one uppercase letter), equals sign, and a binary value (zero or one). Print one empty line after each test case.

Sample Input

```
     0=+
       |
       | #######
       +=#     #
         # &   #o=--+
1=------=#     #    |
         #     #    |
         #######    +--=### ###
                    |   #=#=#1#o==X
       1=-----------x--=# # ###
               1=---x--=###
                    +------------=Y
***********************************
1=A
***
*
```

Output for Sample Input

```
X=0
Y=1

A=1
```