Problem 7: Off Base

Integers in bases larger than 10 are usually represented using letters to represent the digits which have values larger than 9. For example, hexadecimal (base 16) integers are written using the digits 0 through 9 (to represent themselves) and the letters A through F to represent the digits with values 10 through 15, respectively. In a similar manner, the remaining letters in the alphabet could be used to represent the digits with values 16 through 35. This allows for easy display of values in any base from 2 through 36.

In rummaging through a collection of files on an old reel of magnetic tape, a computer archeologist came across a file containing a sequence of what appeared to be arithmetic formulas. The expressions were of the form

number-1 operator number-2 = number-3

where *number-1*, *number-2*, and *number-3* are formed from the digits 0-9 and the letters A-Z, and *operator* is one of '+' or '-' – presumably meaning addition or subtraction. The archeologist would like to know if these expressions really do represent valid expressions, and if so, in what base the numbers were written. The assumption is made that the "digits" A through Z (upper case only) do represent digits with values 10 through 35, none of the numbers are negative, and none of the numbers contain more than 50 digits.

You volunteer to help. At the outset, you know that examining a single number is insufficient to determine its base. For example, the number 77 could be written using any base greater than 7. If, however, you should see the expression

77 + 22 = 99

then you can easily tell than 10 is the smallest base that could have been used. On the other hand, if the expression was

77 + 22 = 121

then you can determine that the numbers were written in base 8. Your problem is to write a program that will assist the archeologist in determining the smallest base used to represent the numbers in each expression, and to also identify those expressions that couldn't have been expressed using a number base in the range 2 to 36.

Input

There will be multiple cases. For each case there will be a single line containing an expression. Blanks may appear before or after any of the numbers, the operator, or the equal sign, but they are not required. An empty line (that is, one containing only zero or more blanks) will follow the expression for the last case.

Output

For each case, display the case number (starting with 1) and the smallest base that could have been used if the expression is correct. If the expression could not be correct using any of the possible bases, then display the case number and the statement "expression is invalid."

Sample Input

```
77 + 22 = 99
115 + 26 = 143
2K3 - 1A1 = 1J2
2N + M = 3I
```

This line is blank

Expected Output

```
Case 1: minimum base is 10
Case 2: minimum base is 8
Case 3: expression is invalid.
Case 4: minimum base is 27
```