

## 6419 Matchstick Maths

For reasons known only to yourself, you have decided to write a program to solve “Matchstick maths” problems. These problems consist of a simple mathematical equation that has been expressed graphically, using matchsticks to form the digits and operators (see Figure 1). The trick of such a “Matchstick maths” equation is that the expression is given with one of the matchsticks having been moved from one possible location to another possible location, thereby creating an expression that is either badly formed (i.e., some of the digits or operators may not be represented correctly), or an expression that does not evaluate to true (e.g.,  $1+1=3$ ).

Your task is to check whether a given expression can be made to evaluate to true by moving a single matchstick from one possible location to another possible location.

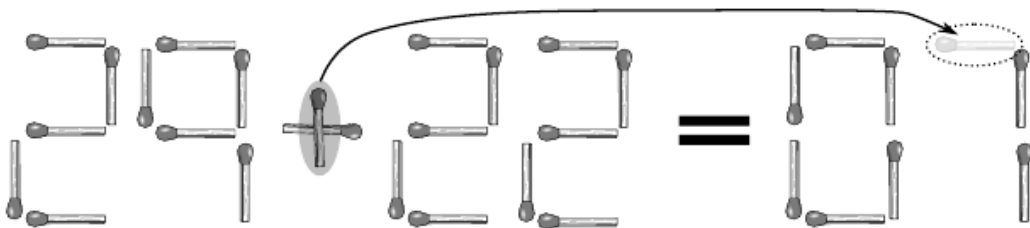


Figure 1: An example of matchstick maths:  $29 + 22 = 1$  can be turned into  $29 - 22 = 7$  by moving one matchstick.

### Input

Your input consists of an arbitrary number of records, but no more than 20. Each record contains exactly 44 integers chosen from the set  $\{0, 1\}$ . Each integer represents a single location in the equation where a matchstick may be placed, with a value of ‘1’ denoting the presence of a matchstick. Integers are separated by one or more spaces. Figure 2 shows how the possible locations are numbered, while Figure 3 provides the reference representation for each digit. Any arbitrary sequence of ‘0’ and ‘1’ values may appear in the input record, so take note that the input is not limited to the digits of Figure 3.

Individual numbers are limited to the range  $[0, 99]$ . Numbers in the range  $[0, 9]$  are always represented with a leading zero, e.g., 00 for 0, or 04 for 4.

The end of input is indicated by a line containing only the value ‘-1’.

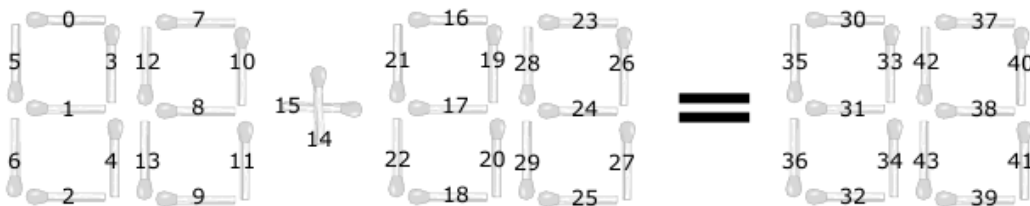


Figure 2: Numbering of match positions.

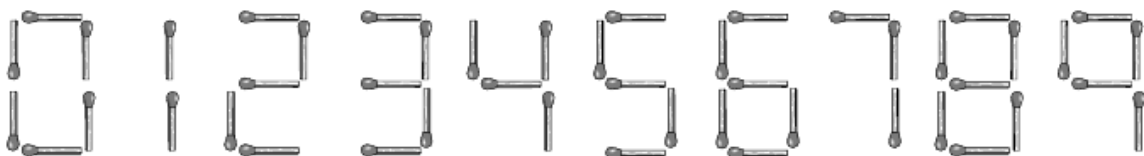


Figure 3: The representation used for each of the digits.

## Output

For each input record, output either

valid

or

invalid

with ‘valid’ being output only if a given equation can be transformed into a valid equation (left hand side of equation evaluates to the same value as the right hand side of the equation) by moving exactly one matchstick.

A valid (true) expression requires that either a plus (+) or minus (-) operator appears in positions 14 and 15 (see Figure 2), and that all the characters are well-formed (i.e., one of the characters in Figure 3).

## Sample Input

```
1 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0
1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1
-1
```

## Sample Output

valid

invalid