

Nowadays, there are smartphone applications that instantly translate text and even solve math problems if you just point your phone’s camera at them. Your job is to implement a much simpler functionality reminiscent of the past — add two integers written down as ASCII art.

An ASCII art is a matrix of characters, exactly 7 rows high, with each individual character either a dot (.) or the lowercase letter ‘x’.

An expression of the form $a + b$ is given, where both a and b are positive integers. The expression is converted into ASCII art by writing all the expression characters (the digits of a and b as well as the ‘+’ sign) as 7×5 matrices, and concatenating the matrices together with a single column of dot characters between consecutive individual matrices. The exact matrices corresponding to the digits and the ‘+’ sign are as follows:

| | | | | | | | | | | |
|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| xxxxxx |x | xxxxxx | xxxxxx | x...x | xxxxxx | xxxxxx | xxxxxx | xxxxxx | xxxxxx | |
| x...x |x |x |x | x...x | x.... | x.... |x | x...x | x...x | ..x.. |
| x...x |x |x |x | x...x | x.... | x.... |x | x...x | x...x | ..x.. |
| x...x |x | xxxxxx | xxxxxx | xxxxxx | xxxxxx | xxxxxx |x | xxxxxx | xxxxxx | xxxxxx |
| x...x |x | x.... |x |x |x | x...x |x | x...x |x | ..x.. |
| x...x |x | x.... |x |x |x | x...x |x | x...x |x | ..x.. |
| xxxxxx |x | xxxxxx | xxxxxx |x | xxxxxx | xxxxxx |x | xxxxxx | xxxxxx | |

Given an ASCII art for an expression of the form $a + b$, find the result of the addition and write it out in the ASCII art form.

Input

The input file contains several test cases, each of them as described below.

Input consists of exactly 7 lines and contains the ASCII art for an expression of the form $a + b$, where both a and b are positive integers consisting of at most 9 decimal digits and written without leading zeros.

Output

For each test case, output 7 lines containing ASCII art corresponding to the result of the addition, without leading zeros.

Sample Input

....x.xxxxx.xxxxx.x...x.xxxxx.xxxxx.xxxxx.....xxxxx.xxxxx.xxxxx
....x.....x.....x.x...x.x.....x.....x...x...x...x.x...x.x...x
....x.....x.....x.x...x.x.....x.....x...x...x...x.x...x.x...x
....x.xxxxx.xxxxx.xxxxx.xxxxx.xxxxx....x.xxxxx.xxxxx.xxxxx.x...x
....x.x.....x.....x.....x.x...x...x...x...x...x...x.x...x
....x.x.....x.....x.....x.x...x...x...x...x...x...x.x...x
....x.xxxxx.xxxxx....x.xxxxx.xxxxx....x.....xxxxx.xxxxx.xxxxx

Sample Output

....x.xxxxx.xxxxx.xxxxx.x...x.xxxxx.xxxxx
....x.....x.....x.x...x.x.....x
....x.....x.....x.x...x.x.....x
....x.xxxxx.xxxxx.xxxxx.xxxxx....x
....x.x.....x.....x.....x...x...x
....x.x.....x.....x.....x...x...x
....x.xxxxx.xxxxx....x.xxxxx....x