Problem 3: Mayan Math

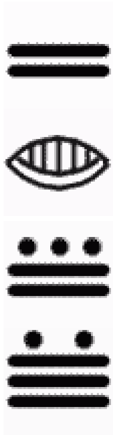
Source filename:	`mayan.(cpp|java)`
Input filename:	`mayan.in`
Output filename:	`mayan.out`

The **Maya** were noted for their fully developed written language, as well as their art, architecture, mathematical and astronomical systems. At its peak, it was one of the most densely populated and culturally dynamic societies in the world.

In common with the other Mesoamerican civilizations, the Maya used a base-20 (vigesimal) numbering system. They used a system of bars and dots as "shorthand" for counting. A dot stood for one, a bar stood for five and the zero was represented by a “shell” glyph. Figure 1 lists all of the glyphs used in the Mayan number system.

Like our numbering system, they used place values to expand this system to allow the expression of very large values. To represent large values, their glyphs were arranged vertically – with the least significant glyph at the bottom.

For example, here is the Mayan expression for the number 80,277:



($10*20^3 + 0*20^2 + 13*20 + 17 = 80277$)

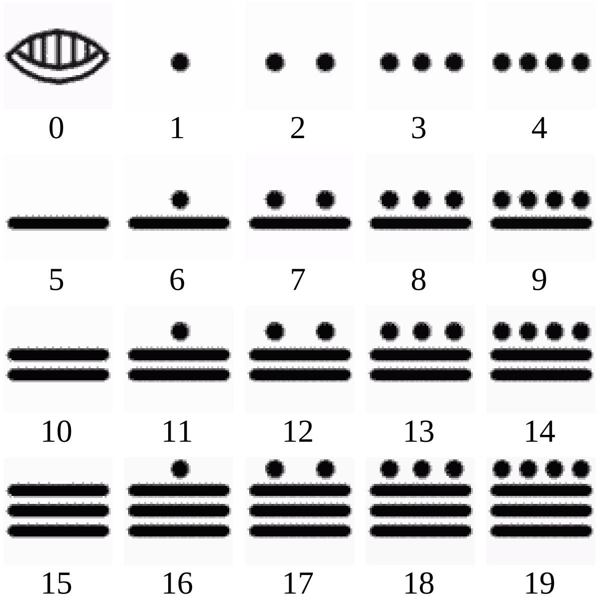
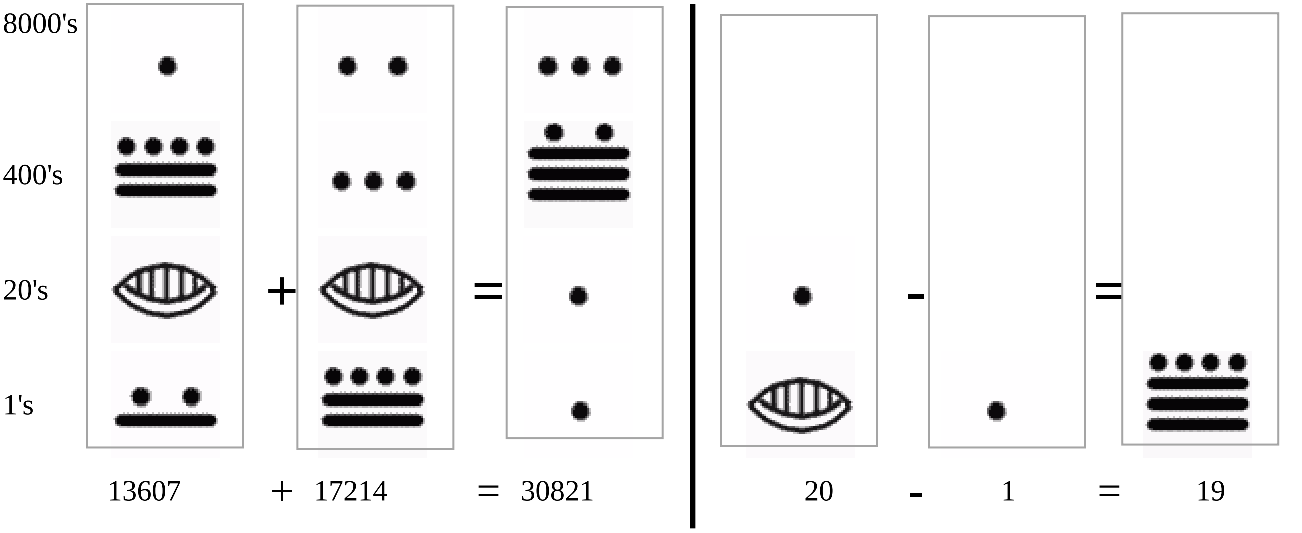


Figure 1

It was very easy to add and subtract using this number system. Here are two examples:



As you can see, adding is just a matter of adding up dots and bars! A group of 5 dots should be replaced by a bar. A group of 4 bars should be replaced by a dot in the next highest glyph. The reverse is done when subtracting.

Objective

Write a program that reads Mayan numbers and performs arithmetic operations - producing Mayan numbers as output.

Input File Specifications (`mayan.in`)

The input file contains one or more test cases. The end of the input file is marked by a line that contains a single zero, ‘0’.

Each test case is represented by several lines that contain 2 Mayan numbers separated by an arithmetic operator (either addition or subtraction). Each Mayan number is made using n glyphs ($1 \leq n \leq 4$) each separated from the others by a single blank line. However, the only blank lines are those found between the glyphs; there are no blank lines before the most significant glyph or after the least significant.

Each individual glyph is made using from 1 to 4 lines that represent the dots and lines (or shell) as listed in figure 1. Asterisks ‘`*`’ are used to represent the dots, lines are represented by five dashes ‘-----’ (ASCII=45) and the shell is represented by the at-sign ‘`@`’. The line after the least significant glyph of a Mayan number contains a single pound sign ‘`#`’ to mark the end of the Mayan number. Between each pair of Mayan numbers is either a plus sign ‘`+`’ or a minus sign ‘`-`‘ (ASCII=45).

For this problem, all arithmetic operations will result in non-negative values.

Output Specifications (`mayan.out`)

For each test case, read the pair of Mayan numbers and perform the indicated operation. Output the answer using the same format that is described above to represent a Mayan number, using 1 to 5 glyphs and the line after the least significant glyph that contains single pound sign ‘`#`’. The only blank lines in the output should be the ones that are used to separate the glyphs of a particular Mayan number.

Example Input File

```
*
          ← <Blank Line>
****
-----
-----
          ← <Blank Line>
@
          ← <Blank Line>
**
-----
#
+
**
          ← <Blank Line>
***
          ← <Blank Line>
@

****
-----
-----
#
*
          ← <Blank Line>
@
#
-
*
#
0
```

Corresponding Output

```
***
          ← <Blank Line>
**
-----
-----
-----
          ← <Blank Line>
*
          ← <Blank Line>
*
#
****
-----
-----
-----
#
```