

Typically, in a base b number system the digit set is $\{0, 1, \dots, b - 1\}$.

However, it is possible to allow a digit set with negative digits

$$\{-a, -(a - 1), \dots, -1, 0, 1, \dots, (a - 1), a\}$$

with appropriately chosen a . In the sequel, the negative digits will be written as $'d$ when d is a positive digit, such that a negative digit occupies two characters in a recording of a number. For example, with $b = 10$ and $a = 6$ we have the following digits

$$\{'6, '5, '4, '3, '2, '1, 0, 1, 2, 3, 4, 5, 6\}$$

The resulting number system is called *signed-digit* system and in order to make the parameters explicit we call it $SD(b, a)$ system, in the case above $SD(10, 6)$. Computing a number from its representation in an SD system is the same as in usual number systems, just some digits have negative values.

Using two digits in $SD(10, 6)$ we can record all the numbers in the range $'6'6.66$, i.e. in the usual decimal notation the range is -66.66 , but with the new digits we do not need a sign for the entire number. The SD systems are redundant in that a number can have more than one representation. For example, in $SD(10, 6)$ the decimal number 4 has at least two representations: 4 and $1'6$.

The redundancy of $SD(b, a)$ number system allows to design faster algorithms for addition. If the following conditions are satisfied

$$\left\lceil \frac{(b + 1)}{2} \right\rceil \leq a \leq b - 1$$

we can choose one of the available representations of each number in such a way that we can design an algorithm for addition which runs in constant time as the carry propagation can be eliminated. However, we will not be concerned with carry propagation here.

Your task is simple. Given a number n in usual decimal notation fitting into a 32-bit integer, a positive $b \leq 10$ and a satisfying the conditions stated above, you are asked to convert n into its $SD(b, a)$ representation where the negative digits are recorded as explained above. If n has more than one representation in $SD(b, a)$, then anyone will do.

Input

The input file contains a sequence of lines, each line contains three numbers: n , b and a . The last line of input has $b = 0$ and should not be processed.

Output

For each line of input produce one line of output containing a representation of n in $SD(b, a)$.

Sample Input

```
98 10 6
-89 10 6
456789 5 3
2147483647 6 4
0 10 9
0 0 0
```

Sample Output

```
10'2
'111
11'111'113'1
10'13032010'131
0
```

