2781 Powerful Calculator

Today, facing the rapid development of business, ACM (Association of Calculator Management) recognizes that more powerful calculator should be studied, developed and appeared in future market shortly. ACM now invites you attending such amazing research and development work.

In most business application, the top three useful calculation operators are Addition (+), Subtraction (-) and Multiplication (*) between two given integers. Normally, you may think it is just a piece of cake. However, since some integers for calculation in business application may be very big, such as the GDP of the whole world, the calculator becomes harder to develop.

For example, if we have two integers 20000000000000000 and 4000000000000000, the exact results of the addition, subtraction and multiplication are:

```
20000000000000000 + 4000000000000000 = 24000000000000000
20000000000000000 - 4000000000000000 = 16000000000000000
20000000000000000 * 4000000000000000 = 80000000000000000000000000000000000
```

Note: ACM prefers the exact format of the results rather than the float format or scientific remark format. For instance, we need "24000000000000000" rather than $2.4 * 10^{16}$.

As a programmer in ACM, your current task is to develop a program to obtain the exact results of the addition $(a + b)$, subtraction $(a - b)$ and multiplication $(a * b)$ between two given integers a and b.

Input

The input file consists of several test cases. Each case has two separate lines where the first line gives the integer a and the second gives b ($|a| < 10^{200}$ and $|b| < 10^{200}$).

When both a and b are equal to zero, the input file ends.

Output

For each test case in the input file, output three separate lines showing the exact results of addition $(a + b)$, subtraction $(a - b)$ and multiplication $(a * b)$ of that case, one result per line.

Leaving a blank line between two successive test cases in the output.

Sample Input

```
20000000000000000
4000000000000000
0
0
```

Sample Output

```
24000000000000000
16000000000000000
80000000000000000000000000000000000
```