H. Continued Fractions

The (simple) continued fraction representation of a real number r is an expression obtained by an iterative process of representing r as the sum of its integer part and the reciprocal of another number, then writing this other number as the sum of its integer part and another reciprocal, and so on. In other words, a continued fraction representation of r is of the form

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \dots}}}$$

where a_0, a_1, a_2, … are integers and a_1, a_2, … > 0. We call the a_i-values *partial quotients*. For example, in the continued fraction representation of 5.4, the partial quotients are a_0 =5, a_1 = 2, a_2 = 2. This representation of a real number has several applications in theory and practice. If r is a rational number, the partial quotients are eventually all zero, so we only need to consider a finite number of partial quotients.

Given two rational numbers in continued fraction representation, your task is to perform the four elementary arithmetic operations on these numbers and display the result in continued fraction representation.

Input:

There may be multiple cases to consider. Process until an end-of-file is detected. Each test case consists of three lines. The first line contains two integers n_1 and n_2, where $1 \le n_i \le 9$ is the number of partial quotients of rational number r_i for $1 \le i \le 2$. The second line contains the partial quotients of r_1 and the third line contains the partial quotients of r_2. The absolute values of the quotients are not more than 10 and you may assume that $r_1 > r_2 > 0$.

Output:

Display the partial quotients of the continued fraction representations of $r_1 + r_2$, $r_1 - r_2$, $r_1 \times r_2$, and r_1 / r_2, in order, each in a line. Do not print any trailing zero partial quotients. Follow this format exactly: "Case", one space, the case number, a colon and no trailing space on the first line for a case, and the answers on separate lines, again with no trailing spaces.

Sample Input	Sample Output
4 3 5 1 1 2 5 2 2	Case 1: 11 0 5 30 4 6 1 27