

Problem I: Egyptian Fractions

Source file: `egypt.{c, cpp, java}`

Input file: `egypt.in`

During the Middle Kingdom of Egypt (circa 2000 BC), the Egyptians developed a novel way of writing fractions. Adding a certain symbol to the hieroglyph for an integer was understood to represent the reciprocal of that integer. This made it easy to write any fraction of the form $\frac{1}{N}$ (called a *unit fraction*)—simply add the reciprocal symbol to the hieroglyph for N .

There was no direct way to represent a fraction of the form $\frac{M}{N}$, where $M > 1$. Instead, any such fraction was written as the sum of unit fractions. For example, the fraction $\frac{3}{4}$ could be written as:

$$\frac{3}{4} = \frac{1}{2} + \frac{1}{4}$$

Notice that multiple sums may yield the same result; for example, $\frac{3}{4}$ can also be written as

$$\frac{3}{4} = \frac{1}{3} + \frac{1}{4} + \frac{1}{6}$$

Your job is to take a fraction $\frac{M}{N}$ and write it as a sum of unit fractions by way of a “greedy” search. In a greedy search you continually subtract the largest unit fraction possible until you are left with zero. For example, consider the fraction $\frac{9}{20}$. A greedy search would find the sum $\frac{1}{3} + \frac{1}{9} + \frac{1}{180}$ in three steps, like so:

$$(1.) \quad \frac{9}{20} - \frac{1}{3} = \frac{7}{60}, \quad (2.) \quad \frac{7}{60} - \frac{1}{9} = \frac{1}{180}, \quad (3.) \quad \frac{1}{180} - \frac{1}{180} = 0$$

Note that at each step we subtracted the largest possible unit fraction from whatever remained of our original fraction.

One additional restriction must be added to keep our solutions from becoming too large: Each time we subtract a unit fraction, we must be left with a fraction whose denominator is strictly less than 1,000,000. For example, consider the fraction $\frac{17}{69}$. The first two unit fractions we would subtract would be $\frac{1}{5}$ and $\frac{1}{22}$, leaving us with $\frac{7}{7590}$. At this point, the largest unit fraction we could subtract would be $\frac{1}{1085}$, leaving us with

$$\frac{7}{7590} - \frac{1}{1085} = \frac{1519}{1647030} - \frac{1518}{1647030} = \frac{1}{1647030}$$

Unfortunately, this leaves us with a fraction whose denominator is larger than 1,000,000, so we can *not* use this unit fraction in our sum. We move on to the next largest unit fraction, $\frac{1}{1086}$, which leaves us with

$$\frac{7}{7590} - \frac{1}{1086} = \frac{1267}{1373790} - \frac{1265}{1373790} = \frac{2}{1373790} = \frac{1}{686895}$$

Since the final answer reduces to a fraction with a denominator less than 1,000,000, we *can* use this unit fraction, leaving us with a final answer of $\frac{1}{5} + \frac{1}{22} + \frac{1}{1086} + \frac{1}{686895}$.

In this case we only had to skip over a single fraction. In general, however, you may have to skip over many fractions in order to find one that will work. While you are searching, you will have to perform calculations on many fractions with large denominators; make sure you do so efficiently, or your program may take too long to execute.

You should also make sure you are using a data type large enough to hold the large numbers you are working with. Even though we have restricted denominators to 1,000,000, you may have to calculate intermediate values with denominators up to 10^{12} . A 64-bit integer will be required to hold such values (`long` in Java, `long long` in C/C++).

Finally, it might be worth noting that, by its very nature, the greedy algorithm will always find some answer consisting of fractions with denominators less than 1,000,000 since, at the very least, any fraction can be represented as a sum of unit fractions with its own denominator. For example:

$$\frac{3}{999983} = \frac{1}{999983} + \frac{1}{999983} + \frac{1}{999983}$$

Input: The input will consist of a sequence of fractions; one per line. Each line will contain only two integers M and N , where $1 < M < N < 100$, representing the fraction $\frac{M}{N}$. M and N will have no common divisors greater than 1. The end of the input will be indicated by two zeros: 0 0.

Output: For each input fraction $\frac{M}{N}$ you are to output a single line containing numbers $D_1 \leq D_2 \leq D_3 \leq \dots$ such that:

$$\frac{M}{N} = \frac{1}{D_1} + \frac{1}{D_2} + \frac{1}{D_3} + \dots$$

This should be the first sum arrived at using a greedy search while enforcing the denominator bound of 1,000,000. Each number should be separated by a single space, with no leading or trailing whitespace.

Example input:	Example output:
3 4	2 4
2 7	4 28
9 20	3 9 180
17 69	5 22 1086 686895
0 0	