G - Trading Card Game

Source file name: `game.c`, `game.cpp`, *or* `game.java`
Author(s): Federico Arboleda and Alejandro Sotelo

Little Ricky is obsessed over the new trading card game, *Sorcery: The Meeting*. He just cannot stop talking about it! He is so obsessed, in fact, that he spends most of his monthly allowance in buying *Sorcery: The Meeting* trading cards, in the hopes of getting all of them.

Of course, buying every single opened card would be too expensive for poor little Ricky, who has to buy everything on his mother's allowance. Instead, he decides to save up during some months and then buy as many unopened cards as he can, hoping he can get them all. Fortunately for Ricky, unopened cards are sold individually!

Being as obsessed as he is, he knows exactly how many cards are in circulation and that, unlike in some other trading card games, there is exactly the same chance to find each card in any single buy. Sadly, Ricky is not very good at math and so he cannot even begin to comprehend what he's going to find in such a big buy. He has, though, put aside some of his allowance this month to ask for your help in calculating the odds of making a good buy (in exchange for the price of a couple of *Sorcery: The Meeting* cards, of course).

If Ricky tells you there are N *Sorcery: The Meeting* trading cards in circulation, all of them equally likely, and he has saved enough to buy m of them at the same time, what is the probability that he will get exactly k different cards?

Being as obsessed as he is, he knows that floating-point numbers would necessarily incur a loss of precision, which he will not tolerate. Therefore, he wants this information as a fraction in lowest terms.

Input

The input consists of several test cases. Each case is a line with three blank-separated integers, N, m and k, which represent, respectively, the total number of cards in circulation, the number of cards Ricky is going to buy, and the number of different cards he expects to get ($1 \leq N \leq 100$, $0 \leq m \leq 100$ and $0 \leq k \leq 100$).

The end of the input is given by $N = m = k = 0$, which should not be processed as a test case.

The input must be read from standard input.

Output

For every test case print a line of the form "`p/q`", where $\frac{p}{q}$ is a fraction in lowest terms representing the probability that Ricky will get exactly k different cards under the described conditions. A probability of 0 should be represented as "`0/1`" and a probability of 1 should be represented as "`1/1`".

The output must be written to standard output.

Sample Input	**Sample Output**

10 1 1
10 2 2
10 12 11
10 4 2
10 4 1
10 4 0
0 0 0
	1/1
	9/10
	0/1
	63/1000
	1/1000
	0/1