

Problem D: Probability Paradox

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

Input file: `paradox.in`

This problem considers repeated tosses of a fair coin. Each outcome, H or T, has probability $1/2$. Any specific sequence of tosses of the same length, like HHH or THH, has the same probability (for example, $1/8$ for a sequence of length 3).

Now consider the following two-player game. Each player chooses a distinct pattern of possible coin flips having a common fixed length. For example, the first player might predict HHH while the second predicts THH. The game then begins with both players observing the same coin as it is repeatedly flipped, until one of them witnesses their pattern. For example, if the sequence of observed flips begins HHTHTTHH... the second player in our example wins the game, having just witnessed the pattern THH.

The question that interests us is, given the two players' patterns, how likely is it that the first player wins the game? Because we are flipping a fair coin, many would assume that the patterns are irrelevant and that each player has probability $1/2$ of winning the game. However, this is not the case, leading to what is known as the **Probability Paradox**.

For some patterns, it will be that the first player wins with probability precisely $1/2$. For example, by symmetry, the patterns TT and HH have equal chances of occurring first.

However, consider again our original example in which the first player chooses HHH and the second player chooses THH. For this particular match-up, the only way that the first player can win is if each of the first three tosses are H. For if the *earliest* HHH were to come somewhere other than at the beginning of the game, the pattern could be represented as

...?HHH

where "..." means a possibly empty earliest part of the sequence, and "?" refers to the toss immediately before the HHH. The "?" can not refer to an H, as in

...HHHH

because there would have been an earlier HHH that ended the game, the underlined part of ...HHHH. Yet if the preceding toss were a T, as in

...THHH.

then the second player would have already won, having observed pattern THH at the underlined ...THHH. Therefore, when considering pattern HHH vs. THH, the first player wins if and only if the first three flips are H, an event that happens with probability $1/8$.

As one more example, if the first player chooses TTH and the second chooses THH, the first player will win with probability $2/3$. Your job is to write a program that computes such a probability.

Input: The input will contain one or more datasets, each on a single line. Each dataset will consist of two equal-length yet distinct patterns using only characters H and T. The common pattern length will be in the range from 1 to 9, inclusive. There is a space between the two patterns. The input ends with a line containing only "\$".

Output: For each data set, output a single line with the *exact* probability that the first sequence precedes the second in a random sequence of fair coin tosses. The probability must be stated as a rational number, reduced to lowest terms, with a "/" between the numerator and denominator. Because each player has a nonzero chance of winning, this probability will always be strictly between 0 and 1.

Warning: The numerator and denominators for all of the final probabilities can be expressed as 32-bit integers. However, depending on your approach, you may need 64-bit integers (type **long** in Java or **long long** in C++) for some intermediate calculations, and even then you must be careful.

Example input:	Example output:
TT HH	1/2
HHH THH	1/8
TTH THH	2/3
THHTH HTHTT	15/28
HTTHHHTHT THHHTHTHH	259/452
\$	