ACM Intercollegiate Programming Contest
Pacific NW Region
1999

Problem G
String Matching

It's easy to tell if two words are identical – just check the letters. But how do you tell if two words are almost identical? And how close is "almost"?

There are lots of techniques for approximate word matching. One is to determine the best substring match, which is the number of common letters when the words are compared letter–by–letter. The key to this approach is that the words can overlap in any way. For example, consider the words CAPILLARY and MARSUPIAL. One way to compare them is to overlay them:

```
CAPILLARY
MARSUPIAL
```

There is only one common letter (A). Better is the following overlay:

```
CAPILLARY
     MARSUPIAL
```

with two common letters (A and R), but the best is:

```
   CAPILLARY
MARSUPIAL
```

Which has three common letters (P, I and L).

The approximation measure appx(word1, word2) for two words is given by:

$$\frac{\text{common letters} * 2}{\text{length(word1)} + \text{length(word2)}}$$

Thus, for this example, appx(CAPILLARY, MARSUPIAL) = 6 / (9 + 9) = 1/3. Obviously, for any word W appx(W, W) = 1, which is a nice property, while words with no common letters have an appx value of 0.

Input:
The input for your program will be a series of words, two per line, until the end–of–file flag of –1. Using the above technique, you are to calculate appx() for the pair of words on the line and print the result. For example:

```
CAR CART
TURKEY CHICKEN
MONEY POVERTY
ROUGH PESKY
A A
-1
```

The words will all be uppercase.

Output:
Print the value for appx() for each pair as a reduced fraction, like this:

```
appx(CAR,CART) = 6/7
appx(TURKEY,CHICKEN) = 4/13
appx(MONEY,POVERTY) = 1/3
appx(ROUGH,PESKY) = 0
appx(A,A) = 1
```

Fractions reducing to zero or one should have no denominator.