Problem 3: Behavior Patterns

Overview
Many online services now try to verify a user’s identity by comparing the pattern (including timing) of keystrokes as various phrases are typed. If a user’s pattern of keystrokes differs from a known standard – for that user – then the current user is assumed to be an impersonator.

A similar analysis can be made of commands given to text-based operating systems or of mouse movements and menu selections within a graphical user interface.

In all of these cases, the actions are encoded using some system and then verified as part of the security process.

Problem
In this problem, you will first be given an encoding of an actual user’s actions. Each action will be encoded by a single character, and – within an encoding – the same character will always be used for that action. Then you will be given a standard encoding for a known user performing a similar task. Unfortunately, the standard encoding may encode the various actions using different characters. It is your job to detect cases which are clearly impersonation.

Two sequences will be considered equivalent if one can be transformed into another through (simultaneous) substitution of one letter for another via a one-to-one function. [As an immediate consequence of this, two sequences are not equivalent if they have different lengths.]

Thus, the sequences “AABACBADB” and “XXAXQAXMA” are equivalent (A→X, B→A, C→Q, D→M). Similarly, “PROG” is equivalent to “GRAM” (P→G, R→R, O→A, G→M). Conversely, “ORANGE” is not equivalent to “BANANA” (because both R, N, and E would all have to map to A.)

Input
Each line of the input (except the last) will consist of two encoded sequences separated by a single space. Each encoding will consist of between 1 and 128 upper or lower case sensitive letters. (This implicitly limits the user to 52 different actions.) The final line of the input will have only the string “END” – which will never be an encoding itself.

-over-
Output

Each line of input, other than the last, generates a single line of output indicating whether there is a user mismatch or a possible match, formatted as in the examples.

Example

Input

AABACBADB XXAXQAXMA
PROG GRAM
ORANGE BANANA
Aa BB
abc ABCD
Success mDlleYY
END

Output

Possible match
Possible match
Mismatch
Mismatch
Mismatch
Possible match