PROBLEM A
BEJMUL (JUMBLE)

Most of you are probably familiar with Jumble™, a daily newspaper feature, where you reconstruct scrambled words. For example, you might be given "ROYIN" and have to come up with "IRONY".

One aspect that makes a particular scrambling of letters "hard" is if it looks like a "real" word. Thus, "ROYIN" is tougher scramble than "NRYIO" because it confuses your brain into thinking that it's already a word.

Another aspect that makes a scrambling harder is whether letters are in their correct position. So, "RIONY" isn't a very good scramble, even if it might kind−of look like a real word.

Your task is to write a program that scores scrambles as "good", "fair", "poor", or "not". Your program will read pairs of words and score them. A scramble is "good" if none of its letters are in the correct place and it looks "real" (more on that later). A scramble is "poor" if it doesn't look "real" and has either the first letter in place or any two consecutive letters in place. If the word isn't scrambled at all, it is said to be "not" scrambled. Otherwise, the scramble is "fair".

How do we know if a word looks "real"? We will use an extremely crude heuristic that the word must alternate between vowels ('Y' is a vowel for these purposes) and consonants. However, certain groups of vowels and consonants are allowed:

AI AY EA EE EO IO OA OO OY YA YO YU
BL BR CH CK CL CR DR FL FR GH GL GR
KL KR KW PF PL PR SC SCH SCR SH SHR
SK SL SM SN SP SQ ST SW TH THR TR
TW WH WR

Also, all double consonants are allowed. No other combinations are allowed, so "SWR" wouldn't be good, even though both "SW" and "WR" are both OK.

Input:
The input for your program will be pairs of words, all upper−case letters, alternating until a "word" of 999 is encountered. The first word of the pair is the original word; the second is the scrambled version that you must evaluate. You may assume that the words are anagrams of one another and contain no spaces. Words will be at least 3 letters in length.

Input will be from the text file **A.in**

Output:
For each pair, you should print how well the second scores as a scramble of the first. Use the format shown in the sample.

(Sample I/O on back)

Sample I/O:

Input:	*Output:*
`SPAM`	`"MAPS" is a fair scramble of "SPAM"`
`MAPS`	`"RIONY" is a fair scramble of "IRONY"`
`IRONY`	`"ONYRI" is a good scramble of "IRONY"`
`RIONY`	`"IRONY" is not a scramble of "IRONY"`
`IRONY`	`"INOYR" is a fair scramble of "IRONY"`
`ONYRI`	`"IOYRN" is a poor scramble of "IRONY"`
`IRONY`	
`IRONY`	
`IRONY`	
`INOYR`	
`IRONY`	
`IOYRN`	
`999`	

DUCK GOLO! (That's a scramble of 'GOOD LUCK'! ☼)