

UCF Local Contest — September 5, 2015

Balanced Strings

filename: balance
(*Difficulty Level:* Medium)

Being an individual obsessed with balancing and trying to find strings that meet that criteria, you have begun a new endeavor to study the curious structure of what we will call balanced strings.

Balanced strings are strings that maintain an equal ratio of consonants to vowels in all of their substrings. What? You don't know the difference between a consonant and a vowel? Vowels are the characters 'a', 'e', 'i', 'o', 'u' and sometimes 'y'. Actually, you don't like the character 'y' because this makes the problem much harder. What was the difficulty level of this problem again? Oh yeah Medium! We can't have a problem about consonants and vowels that makes 'y' sometimes a vowel! That would make the problem a Hard and too many hard problems is a very bad thing. Being obsessed with balance, you have decided that 'y' will be included with the vowels. That way, there are 6 vowels and 20 consonants. A nice balanced even number of both! Oh! I almost forgot! A consonant is any letter that is not a vowel. Also to simplify things (this is a medium problem after all!), we will consider strings composed of lowercase letters only.

Now you are ready to understand balanced strings! A balanced string is a string that has an equal number of consonants and vowels in all of its substrings. Well... not all of its substrings. Just the substrings of even length! For example, the string "orooji" has the following set of even-length substrings: {"or", "ro", "oo", "oj", "ji", "oroo", "rooj", "ooji", "orooji"}. In that set the following substrings are unbalanced: {"oo", "oroo", "ooji", "orooji"}. That is, the substrings do not contain an equal number of vowels and consonants. So, the string "orooji" is not balanced. But a string like "arup" is balanced. The requisite even-length substrings are: {"ar", "ru", "up", "arup"} and all these substrings are balanced (have the same number of vowels and consonants), thus the string "arup" is balanced. Note that a balanced string may contain an odd number of characters, e.g., the string "ali" is balanced since all its even-length substrings ({"al", "li"}) are balanced.

Now you want to take words and erase some of the letters and replace them with new letters to form balanced strings. Since this is a medium problem, we've taken the liberty of erasing some of the letters for you and replaced them with '?' characters. See! The problem is already halfway solved!

The Problem:

Given a string of lowercase letters and '?' characters, count the number of ways to replace all the '?' with lowercase letters such that the string results in a balanced string. Two ways are considered different if there exists some i such that the i^{th} character in one string differs from the i^{th} character of the other string. Note that all the '?' do not have to be replaced by the same letter.

The Input:

The first input line contains a positive integer, n , indicating the number of strings to balance. The strings are on the following n input lines, one string per line. Each string contains only lowercase letters and/or '?' characters. Assume each input string has at least one character and at most 100 characters.

The Output:

For each string, output "String # d : v " where v is the number of ways to replace the questions marks with lower case letters. It is guaranteed that v will fit in a signed 64-bit integer for the strings we provide. Leave a blank line after the output for each string.

Sample Input:

```
7
orooji
al?
a?i
g?ha
a?u?
????????????????
arup
```

Sample Output:

```
String #1: 0

String #2: 6

String #3: 20

String #4: 6

String #5: 400

String #6: 1117952409600000000

String #7: 1
```

Note: The first and last strings in Sample Input do not have any '?'. The first input is not balanced so there is no way of replacing all '?' to make it balanced, thus the output is zero. However, the last input is balanced so there is one way of replacing all '?' to make it balanced, thus the output is 1.