

## Problem K. Keep Distance

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

You are participating in an infamous TV-show. Show participants are usually asked to do ridiculous stuff, and you're not an exception.

Right now, there are several colored cubes arranged in a single row in front of you. Your quest goes as follows. The show host announces her favorite color among those present on the cubes. Then, you have to rearrange the cubes as fast as possible in such a way that the row becomes *beautiful* with regard to the host's favorite color.

Let's number the cubes in the row with  $1, 2, \dots$  from left to right. Suppose there are  $k$  cubes of color  $c$  in the row, namely, at positions  $p_1 < p_2 < \dots < p_k$ . The row is called *beautiful* with regard to color  $c$  if the neighboring cubes of this color are placed at equal distances from each other, that is, if the following equation is satisfied:  $p_2 - p_1 = p_3 - p_2 = \dots = p_k - p_{k-1}$ . Note that if  $k \leq 2$ , the row is beautiful with regard to color  $c$  as well.

During the rearrangement of the cubes, you're only allowed to perform the following operation any number of times: take out any two cubes of different colors and exchange their positions in the row. Note that these two cubes do not have to be adjacent.

You already see the row of the cubes but you don't know the favorite color of the host. Now that there's still some time before the ultimate quest, you decided to calculate the minimum number of cube swaps required to make the row beautiful with regard to every present color.

### Input

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 2500$ ) — the number of test cases.

Each of the next  $t$  lines contains a non-empty string consisting of lowercase English letters, representing the colors of the cubes in the row in order from left to right. Here, "a" stands for amaranth, "b" stands for bittersweet, "c" stands for crimson, and so on.

The total length of the input strings doesn't exceed 250 000.

### Output

For each input row, output a sequence of integers. For every color in order from "a" to "z" present in the row, output the minimum number of swaps required to make the row beautiful with regard to this color.

### Example

standard input	standard output
1 vrrrcvrvvr	0 1 2

### Note

In the example test case, the row is already beautiful with regard to crimson ("c"). To make the row beautiful with regard to red ("r"), it's enough to swap the first and the third cubes — the row will look like "rvrvcvrvvr". To make the row beautiful with regard to vermilion ("v"), it's enough, for example, to swap the first and the fifth cubes, and then swap the second and the seventh cubes — the row will look like "crrrvvvvvr".