Problem I
The Imperial Problem

Source file name: `imperial.c, imperial.cpp` *or* `imperial.java`

Since the times of the Mighty Roman Empire, it has been customary to have signals by the roads informing the distance to the greatest city on Earth (that is Rome), the location of a landmark along the road, and in more recent times, the maximum allowed speed. Making all these road signs can be a problem even for someone as mighty as the famous Emperor Tiberius: when he committed the construction of the *Via Solis*, he personally supervised the design of every road sign. Little did he know that he was making an infantile mistake because of a basic misunderstanding of how Roman numerals work.

The Roman number system is based on seven *basic* capital letters, where '`I`'=1, '`V`'=5, '`X`'=10, '`L`'=50, '`C`'=100, '`D`'=500, and '`M`'=1000. Like in the case of Arabic numerals, the decimal digits of a number to be represented with a Roman numeral are translated into Roman notation from left to right, starting with the most significant and finishing with the least significant digit. For example, the number 1111 is written as '`MCXI`', and 1055 is written as '`MLV`'. As the second example shows, the digit 0 does not appear.

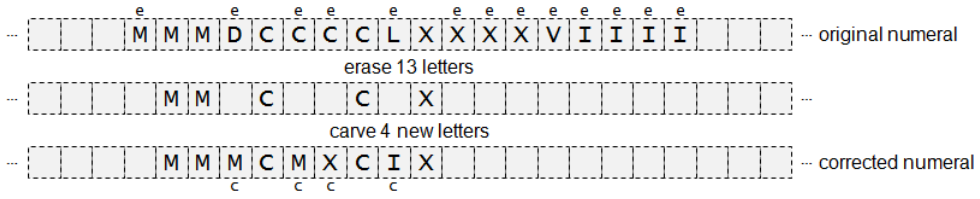
To translate a digit not defined by one of the basic capital letters, a system based on *repetition* and *addition* is used. For example, '`III`'=3, '`XXXII`'=32, and '`MMVIII`'=2008. Only the letters '`I`', '`X`', '`C`', and '`M`' can be repeated, and it does not make sense to repeat a letter more than four times because there is a shorter version for that (i.e., the expressions '`L`' and '`LX`' *must* be used instead of '`XXXXX`' and '`XXXXXX`', respectively).

The *final rule* states that no symbol can appear more than three times in a row, which happens whenever one of the decimal digits is a 4 or a 9. In those cases, a subtraction system *must* be used by combining the problematic symbol with a greater one. For example, 44 should not be written as '`XXXXIIII`' but as '`XLIV`' ((50−10)+(5−1) = 44). Note that '`I`' can only be placed at most once before '`V`' or '`X`', '`X`' can only be placed at most once before '`L`' or '`C`', and '`C`' can only be placed at most once before '`D`' or '`M`'. Because of this, the largest number that can be written using standard Roman numerals is '`MMMCMXCIX`' (3000+(1000−100)+(100−10)+(10−1) = 3999) and one should refrain from writing it as '`IMMMM`'.

The problems for Emperor Tiberius started when he designed all the road signs forgetting the final rule (e.g., instead of writing '`XLIV`', he wrote '`XXXXIIII`'). Many unnecessary letters were carved into the stone signs and the empire run into large overhead costs. For the sake of keeping his name among the greatest emperors of Rome, Tiberius wishes to amend his error. He has hired you to help him fixing his mistake. You must write a program that, given a Roman numeral, written forgetting the final rule (as Tiberius would write it), reports two integers e and c where e is the number of letters that must be *erased* and c is the number of letters that must be *carved* to transform that numeral into that one written using all the stated rules. If there are multiple solutions, your program must print the answer that minimizes the value of

e+c. If still there are two or more solutions minimizing e+c, the one that minimizes the value of e must be printed.

For example, considering a sign with the Roman numeral '`MMMDCCCCLXXXXVIIII`' (3999), it is easy to see that the minimum number of operations needed to fix it is e+c=17: e=13 erased letters plus c=4 new carved letters. The following artifact shows both the original Roman numeral and the corrected one. Erased letters have been marked with an '`e`' above the original numeral and the new carved letters with a '`c`' under the corrected one.



Note that it is not allowed to leave blank spaces inside a Roman numeral, and that every letter uses the same space when carved (i.e., all letters are written using a fixed-width font).

Input

The input consists of several lines, each one containing one Roman numeral written forgetting the final rule. No blanks are found on any of these lines. Input ends with a line containing a single asterisk ('`*`').

The input must be read from the file imperial.in.

Output

For each Roman numeral given in the input, output a single line with two non-negative integer numbers e and c, separated by one blank, where e is the number of erased letters and c is the number of new carved letters that solve the problem.

The output must be written to standard output.

Sample input	**Output for the sample input**
`IIII`	`3 1`
`CCCII`	`0 0`
`CCCCIII`	`3 1`
`CCCCXXXX`	`6 2`
`XXXXVIIII`	`7 2`
`MMMDCCCCLXXXXVIIII`	`13 4`
`*`	