Given a string of characters, we can permute the individual characters to make new strings. If we can impose an ordering on the characters (say alphabetic sequence), then the strings themselves can be ordered and any given permutation can be given a unique number designating its position in that ordering. For example the string 'acab' gives rise to the following 12 distinct permutations:

aabc	1	acab	5	bcaa	9
aacb	2	acba	6	caab	10
abac	3	baac	7	caba	11
abca	4	baca	8	cbaa	12

Thus the string 'acab' can be characterised in this sequence as 5.

Write a program that will read in a string and determine its position in the ordered sequence of permutations of its constituent characters. Note that numbers of permutations can get very large; however we guarantee that no string will be given whose position is more than $2^{31} - 1 = 2,147,483,647$.

Input

Input will consist of a series of lines, each line containing one string. Each string will consist of up to 30 lower case letters, not necessarily distinct. The file will be terminated by a line consisting of a single '#'.

Output

Output will consist of a series of lines, one for each line of the input. Each line will consist of the position of the string in its sequence, right justified in a field of width 10.

Sample Input

```
bacaa
abc
cba
#
```

Sample Output

```
        15
         1
         6
```