

Esquemas de codificação são frequentemente usados em situações que necessitam de criptografia ou para economia no armazenamento/transmissão de dados. Aqui, nós desenvolveremos um simples esquema de codificação que codifica tipos particulares de palavras com cinco ou menos letras minúsculas em inteiros.

Considere o alfabeto Inglês $\{a,b,c,\dots,z\}$. Usando este alfabeto, um conjunto de palavras *válidas* será formado em ordem lexicográfica. Neste conjunto de palavras *válidas*, as letras de uma palavra estão estritamente em ordem ascendente; isto é, as letras ao final de uma palavra válida sempre estarão após letras seguindo suas posições na lista de letras do alfabeto $\{a,b,c,\dots,z\}$. Por exemplo:

abc aep gwz são palavras válidas de três letras, porém:

aab are cat não são.

Para toda palavra válida, associamos um inteiro que dá a posição da palavra na lista ordenada (em ordem alfabética) de todas as palavras *válidas*. Isto é:

```
a -> 1
b -> 2
.
.
z -> 26
ab -> 27
ac -> 28
.
.
az -> 51
bc -> 52
.
.
vwxyz -> 83681
```

Seu programa deverá ler uma série de linhas de entrada. Cada linha de entrada terá uma única palavra, que terá de 1 a 5 letras. Para cada palavra lida, se a palavra for *inválida* imprima o número 0. Se a palavra lida for *válida*, imprima o índice da posição dessa palavra na lista alfabética apresentada acima.

Entrada

A entrada consiste de uma série de palavras, uma por linha. As palavras têm no mínimo uma letra e no máximo cinco letras. Apenas as letras de *a* até *z* (em letras minúsculas) aparecerão na entrada. O primeiro caractere da linha já será o primeiro caractere da palavra. A linha terminará com um “fim-de-linha” (*ENTER*).

Saída

A saída consistirá de um número inteiro, maior ou igual a zero e menor ou igual a 83681. Haverá apenas um número impresso por linha.

Exemplo de Entrada

```
z
a
cat
vwxyz
```

Exemplo de Saída

```
26
1
0
83681
```