

## Problem I. I18n

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         256 megabytes

$i18n$  compression for a string  $s_1 \dots s_n$  ( $n \geq 4$ ) is a string  $s_1 + to\_string(n-2) + s_n$ , where  $to\_string(k)$  is a string containing decimal representation of number  $k$ . For example,  $i18n(abxcd) = a3d$ .

Your task is to minimize the length of given text using  $i18n$  compression for some words (the remaining words will not change). You should compress words in such way that for every compressed word  $x$  there is exactly one uncompressed word  $y$  before  $x$  such as  $i18n(y) = i18n(x)$  and  $y$  must be equal to  $x$ .

### Input

The only line of input contains space separated words consisting of lowercase Latin letters. The length of input file doesn't exceed  $10^5$  characters.

### Output

Print one line with space separated compressed text from input.

### Examples

standard input	standard output
internationalization bla bla bla abcd acbd bcda abcd internationalization internationalization	internationalization bla bla bla abcd acbd bcda abcd i18n i18n
a bb ccc dddd ccc bb a	a bb ccc dddd ccc bb a

### Note

In the first test:

- *bla* is not compressed because it's length is less than 4,
- *acbd* is not compressed because there is *abcd* before it,  $i18n(abcd) = i18n(acdb)$ , but  $abcd \neq acbd$ ,
- the second one *abcd* is not compressed because there are two uncompressed words with the same  $i18n$  before it: *abcd* and *acbd*.