F. String Compression

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ivan wants to write a letter to his friend. The letter is a string s consisting of lowercase Latin letters.

Unfortunately, when Ivan started writing the letter, he realised that it is very long and writing the whole letter may take extremely long time. So he wants to write the *compressed version* of string s instead of the string itself.

The *compressed version* of string s is a sequence of strings $c_1, s_1, c_2, s_2, \ldots, c_k, s_k$, where c_i is the decimal representation of number a_i (without any leading zeroes) and s_i is some string consisting of lowercase Latin letters. If Ivan writes string s_1 exactly a_1 times, then string s_2 exactly a_2 times, and so on, the result will be string s.

The length of a *compressed version* is $|c_1| + |s_1| + |c_2| + |s_2| \ldots |c_k| + |s_k|$. Among all *compressed versions* Ivan wants to choose a version such that its length is minimum possible. Help Ivan to determine minimum possible length.

Input

The only line of input contains one string s consisting of lowercase Latin letters ($1 \le |s| \le 8000$).

Output

Output one integer number — the minimum possible length of a *compressed version* of s.

Examples

input
aaaaaaaaaa
output
3

input
abcab
output
6

input
cczababababab
output
7

Note

In the first example Ivan will choose this compressed version: c_1 is `10`, s_1 is `a`.

In the second example Ivan will choose this compressed version: c_1 is `1`, s_1 is `abcab`.

In the third example Ivan will choose this compressed version: c_1 is `2`, s_1 is `c`, c_2 is `1`, s_2 is `z`, c_3 is `4`, s_3 is `ab`.