**The Goal**

Binary with 0 and 1 is good, but binary with only 0, or almost, is even better! Originally, this is a concept designed by Chuck Norris to send so called *unary* messages.

Write a program that takes an incoming message as input and displays as output the message encoded using Chuck Norris’ method.

**Rules**

Here is the encoding principle:

* The input message consists of ASCII characters (7-bit)
* The encoded output message consists of blocks of 0
* A block is separated from another block by a space
* Two consecutive blocks are used to produce a series of same value bits (only 1 or 0 values):

- First block: it is always 0 or 00. If it is 0, then the series contains 1, if not, it contains 0  
- Second block: the number of 0 in this block is the number of bits in the series

**Example**

Let’s take a simple example with a message which consists of only one character: Capital C. C in binary is represented as 1000011, so with Chuck Norris’ technique this gives:

* 0 0 (the first series consists of only a single 1)
* 00 0000 (the second series consists of four 0)
* 0 00 (the third consists of two 1)

So C is coded as: 0 0 00 0000 0 00

Second example, we want to encode the message CC (i.e. the 14 bits 10000111000011) :

* 0 0 (one single 1)
* 00 0000 (four 0)
* 0 000 (three 1)
* 00 0000 (four 0)
* 0 00 (two 1)

So CC is coded as: 0 0 00 0000 0 000 00 0000 0 00

**Game Input**

Input

Line 1: the message consisting of N ASCII characters (without carriage return)

Output

The encoded message

Constraints

0 < N < 100

Example

Input

C

Output

0 0 00 0000 0 00