

# Chop Ahoy! Revisited!

Given a non-empty string composed of digits only, we may group these digits into sub-groups (but maintaining their original order) if, for every sub-group but the last one, the sum of the digits in a sub-group is less than or equal to the sum of the digits in the sub-group immediately on its right. Needless to say, each digit will be in exactly one sub-group.

For example, the string 635 can only be grouped in one sub-group [635] or in two sub-groups as follows: [6-35] (since  $6 < 8$ .) Another example is the string 1117 which can be grouped in one sub-group [1117] or as in the following: [1-117], [1-1-17], [1-11-7], [1-1-1-7], [11-17] and [111-7] but not any more, hence the total number of possibilities is 7.

Write a program that computes the total number of possibilities of such groupings for a given string of digits.

## Input

Your program will be tested on a number of test cases. Each test case is specified on a separate line. Each line contains a single string no longer than 25, and is made of decimal digits only.

The end of the test cases is identified by a line made of the word "bye" (without the quotes.) Such line is not part of the test cases.

## Output

For each test case, write the result using the following format:

k. n

where k is the test case number (starting at 1,) and n is the result of this test case.

## Example

### Input:

635  
1117  
9876  
bye

### Output:

1. 2  
2. 7  
3. 2