Problem J. Jingles of a String

Input file:	`jingles.in`
Output file:	`jingles.out`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a string s a *jingle* $J(L, R)$ of its substring $s[L..R]$ is the set of characters that appear among its characters. For example, if $s =$ "`abacaba`", then $s[3..5] =$ "`aca`" and $J(3, 5) = \{a, c\}$.

You are given a string s. Find all possible jingles of its non-empty substrings and for each possible jingle find the longest substring of s with such jingle.

In the example above, there are 6 sets that appear as jingles of substrings of s. For example, for a set $\{a, b\}$ there are six substrings that has such jingle: $s[1..2]$, $s[1..3]$, $s[2..3]$, $s[5..6]$, $s[5..7]$, $s[6..7]$. Among them $s[1..3]$ (as well as $s[5..7]$) has length 3, this is the longest substring of s with such jingle.

Since printing all jingles would make output file too large, you have to output the sum

$$v(s) = \sum_{J \in \mathbb{J}} S(J)L(J).$$

Here $\mathbb{J}$ is the set of all jingles of the string, $S(J)$ is the number of characters in J, $L(J)$ is the length of longest substring of s that has J as its jingle.

Input

The input file contains multiple test cases.

The first line of the input file contains t — the number of test cases.

Each of the following t lines contains a string s that consists of at most 100 000 lowercase characters.

The sum of lengths of strings for all test cases in the input file doesn't exceed 100 000.

Output

For each test case first output two integers: d — the number of different sets that appear as jingles of some substrings of s and $v(s)$ — the sum described in the problem statement.

Examples

jingles.in	jingles.out
2 abacaba abbcccdddd	6 36 10 125