Longest Common Prefix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given a string s of length n.

For each position i with $1 \le i \le n$, let t_i be the string obtained from s by deleting the i-th character of s (and keeping the relative order of all other characters).

For two strings x and y, let $\text{lcp}(x, y)$ denote the length of their longest common prefix, that is, the largest integer $\ell \ge 0$ such that the first ℓ characters of x and y are equal (if $x_1 = y_1, x_2 = y_2, \ldots, x_\ell = y_\ell$).

Your task is to choose two **different** positions i and j such that the value $\text{lcp}(t_i, t_j)$ is as large as possible, and compute this maximum possible value.

Formally, you need to find

$$\max_{1 \le i < j \le n} \text{lcp}(t_i, t_j).$$

You do not need to output i or j, only the maximum value of $\text{lcp}(t_i, t_j)$.

Input

The first line contains a single integer T ($1 \le T \le 2 \cdot 10^5$), representing the number of test cases.

Each test case consists of a single line containing the string s ($2 \le |s| \le 5 \cdot 10^5$). It is guaranteed that s only contains lowercase English letters and the sum of $|s|$ does not exceed $5 \cdot 10^5$).

Output

For each test case, output a single integer on a separate line, representing the maximum possible value of $\text{lcp}(t_i, t_j)$ over all pairs of indices $i \ne j$.

Example

standard input	standard output
5 abac abbbb cbacb babcc aa	2 4 3 4 1