|  |  |  |
| --- | --- | --- |
| **G** | **Techboy and his love for**  **string** | Time Limit: 0.5 sec |

Techboy is a very popular contestant in AUST programming community. Now and then he tries his best to set interesting problem for his junior contestants. Whenever he hasn’t find enough time for this, he tries to find out interesting online problem with different story for his junior contestants. As he recently joined a software firm, he doesn’t have that much free time that he used to have when he was a student. So for this Intra AUST programming contest he decided to set a easy but not so common problem for his juniors.

For a given string, Techboy’s junior need to print the length of longest repeating subsequence such that the two subsequence don’t have same string character at same position, i.e., any i’th character in the two subsequences shouldn’t have the same index in the original string.

Say if the given string is “aabb” then the longest repeating subsequence is “ab” , so answer is 2 and if the given string is “aab” then the longest repeating subsequence is “a” and so answer is 1. The two subsequence are ‘a’ (first) and ‘a’ (second). Note that ‘b’ cannot be considered as part of subsequence as it would be at same index in both.

**Input**

At first gives you an integer **T (T<=100)**, is the number of test cases. For each test case there is given a string **S** of lowercase English characters **(‘a’ - ‘z’)** (you can assume that length of **S** is less than or equal to **1000**)

**Output**

For every test case, print case number first and then print the longest repeating subsequence length of that string.

**Sample I/O**

|  |  |
| --- | --- |
| Input | Output |
| 2  aabb  aab | Case 1: 2  Case 2: 1 |