Border Jump 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are given a string S consisting of lower-cased English letters. You need to perform some operations on S until it becomes empty. Each time you can perform one of the following three operations:

1. Delete the first character of S.
2. Delete the last character of S.
3. Choose a good substring S' of S and replace S with S'.

A non-empty string S' is called a good substring of string S if and only if $S' \neq S$, S' is a prefix of S, and the reverse of S' is a suffix of S. The reverse of a string $p_1 p_2 \cdots p_k$ of length k is another string $p_k p_{k-1} \cdots p_1$ of length k.

What's the maximum number of type 3 operations can you perform?

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first and only line contains a string S ($1 \leq |S| \leq 10^5$) consisting of lower-cased English letters.

It is guaranteed that the sum of $|S|$ over all test cases does not exceed 2×10^5.

Output

For each test case, output one line containing one integer indicating the maximum number of type 3 operations you can perform.

Example

standard input	standard output
3 aaaa abbaabba xy	3 4 0

Note

For the first sample test case: `aaaa` $\xrightarrow{\text{op. 3}}$ `aaa` $\xrightarrow{\text{op. 3}}$ `aa` $\xrightarrow{\text{op. 3}}$ `a` $\xrightarrow{\text{op. 2}}$ $\varnothing$.

For the second sample test case: `abbaabba` $\xrightarrow{\text{op. 3}}$ `abbaabb` $\xrightarrow{\text{op. 1}}$ `bbaabb` $\xrightarrow{\text{op. 3}}$ `bbaab` $\xrightarrow{\text{op. 1}}$ `baab` $\xrightarrow{\text{op. 3}}$ `baa` $\xrightarrow{\text{op. 1}}$ `aa` $\xrightarrow{\text{op. 3}}$ `a` $\xrightarrow{\text{op. 1}}$ $\varnothing$.