



# Non-Overlapping Palindromes

Time limit: 7500 ms  
Memory limit: 256 MB

Alice often likes to play with **palindromic** strings. Given a string  $S$ , she wants to find two non-empty palindromic **substrings** that are not overlapping. What is the maximum sum of lengths of these two palindromic substrings?

## Standard input

The input begins with a single integer  $T$  on the first line, the number of test cases.

Each of the next  $T$  lines gives one test case with a single string  $S$ .

## Standard output

For each test case, output a single line with the maximum sum of lengths.

## Constraints and notes

- $1 \leq T \leq 10$
- $S$  contains between 2 and  $10^5$  lowercase English letters.
- A string is palindromic if we can obtain the same string by reversing it. For example, `abcba`, `abba`, `a` are palindromic, and `abc` is not palindromic.

### Input

```
3
xabcbayabbaz
abcbaabc
abcba
```

### Output

```
9
7
4
```

### Explanation

`xabcbayabbaz` contains substrings `abcba` and `abba` that are not overlapping. Their length sum is  $5 + 4 = 9$ .

`abcbaabc` contains substrings `a` and `cbaabc` that are not overlapping. Their length sum is  $1 + 6 = 7$ .