Task palindrome count

⏳1 sec. 💾 128 MB

An ancient inscription discovered in Bulgaria is believed to contain a coded message. Scientists found that the message can be obtained by removing some of the symbols (or none) and the rest form a *palindromic* sequence. A sequence of symbols is *palindromic* if it can be read the same backwards and forwards. The scientists will look at all possible messages.

Help them by finding the number of *palindromic* messages that can be obtained by erasing some (or none) of the symbols of the inscription. As this number can be very large, you have to find it modulo $10^9 + 7$. For simplicity, we will denote the symbols with small Latin letters.

Implementation details

You should implement the function `count_palindromes`:

```
int count_palindromes (std::string s)
```

It will be called once for each test with s – a string, representing the inscription. The function should return the count of *palindromic* messages that can be obtained modulo $10^9 + 7$.

Constraints

- $1 \le N \le 9\ 000$, where N is the size of the string;
- s consists of small Latin letters.

Subtasks

Subtask	Points	Required subtasks	N	Other constraints
0	0	—	—	The example.
1	20	0	≤ 20	-
2	40	$0 - 1$	≤ 500	-
3	40	$0 - 2$	$\le 9\ 000$	-

Example

Consider the following call:

```
count_palindromes("acbac")
```

In this case there are 11 *palindromic* messages that can be obtained: **a**~~cbac~~, ~~a~~**c**~~bac~~, ~~ac~~**b**~~ac~~, ~~acb~~**a**~~c~~, ~~acba~~**c**, **a**~~cb~~**a**~~c~~, ~~a~~**c**~~ba~~**c**, **ac**~~b~~**a**~~c~~, **a**~~c~~**ba**~~c~~, ~~a~~**cb**~~a~~**c**, and ~~a~~**c**~~b~~**ac**. Therefore, the call should return 11.

Sample grader

The input format is the following:

- line 1: a string – the value of s.

The output format is the following:

- line 1: one integer – the return value of the call.