Problem J
Repetitive String Invention
Time Limit: 2 Seconds, Memory Limit: 2G

Lulu has a string consisting of lowercase English letters. She would like to make her string *Repetitive*. A repetitive string has an even number of characters, and the first half of the string exactly matches the second half of the string. For example, "**lulu**", "**abcabc**" and "**xx**" are repetitive strings, while "**xyx**" and "**abac**" are not.

To get a repetitive string, Lulu can take two non-overlapping, non-empty substrings from her string and concatenate them together. The substrings must be concatenated in the order that they appear in her string.

She's wondering, what is the number of ways she can choose two substrings to make a repetitive string? Two ways are different if at least one of the substrings Lulu uses comes from a different part of her string.

Consider the string "**aaaa**".

- There are six ways for Lulu to form the repetitive string "**aa**": by matching each "**a**" with each subsequent "**a**" (1+2, 1+3, 1+4, 2+3, 2+4, 3+4).
- There are also three ways for her to form "**aaaa**": "**a**"+"**aaa**", "**aa**"+"**aa**" and "**aaa**"+"**a**".

So there are nine ways for Lulu to form a repetitive string by concatenating non-overlapping, non-empty substrings of "**aaaa**" in order.

Input

The single line of input contains a single string s ($1 \leq |s| \leq 800, s \in \{\mathbf{a}-\mathbf{z}\}^*$). This is Lulu's string.

Output

Output a single integer, which is the number of ways Lulu can concatenate two non-overlapping, non-empty substrings from her string in order to get a repetitive string.

Sample Input 1	Sample Output 1
aaaa	9

Sample Input 2	Sample Output 2
axabxbcxcdxd	22