

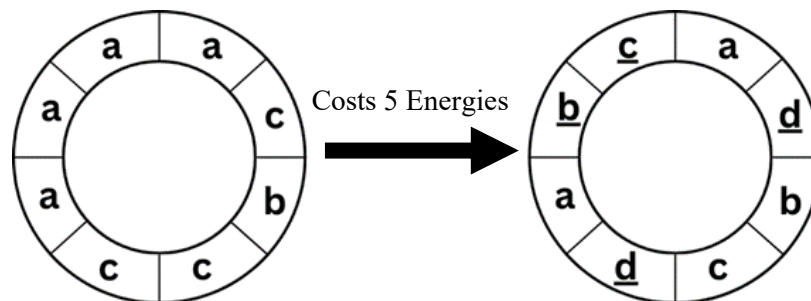
A	Ancient Password	
	Time Limit (seconds)	2
	Memory Limit (MB)	512

Long ago, in a forgotten temple, explorers discovered a stone wheel covered with **26 symbols** that matched the letters of the Latin alphabet from **a to z**. The symbols around the wheel were a mysterious password. According to legend, if the password was set correctly on the wheel, the door would open to reveal treasures.

The symbol was strange. It could not be turned in any direction you wanted.

- Each symbol could only be **rotated forward to the next letter** in the alphabet:
 $a \rightarrow b \rightarrow c \rightarrow \dots \rightarrow z \rightarrow a$
- Every step forward **costs 1 energy**, and if the symbol stays the same cost = 0

You could not jump directly to another letter. **If you wanted to change a letter further away, you had to move step by step, paying the cost each time.**



The password, however, had strict magical rules:

- No pattern of two similar letters next to each other**, like aa or cc.
- No pattern of three letters forming a mirror**, the password could not contain something like aba, cac, or nan. The temple magic **considered these palindromes impure**.
- The wheel is circular, so the **first and last letters are also considered neighbors**, so the **rules apply across the boundary too!**

Your task is **to help the explorers calculate the minimum cost** of rotating the wheel's current symbols into a valid password that satisfies the magical rules.

INPUT

A single string of length n ($3 \leq n \leq 300$), consisting of lowercase letters a–z.

OUTPUT

An integer: the minimum cost needed to transform the word into a valid password.

Input	Output
aaa	3
abyon	0
ababa	8
zzzzzz	6
zayazay	1
abcab	6
acbcaaaa	5

NOTE

abcab $\rightarrow$ abcde (cost = 6)

a $\rightarrow$ b $\rightarrow$ c $\rightarrow$ d = 3

b $\rightarrow$ c $\rightarrow$ d $\rightarrow$ e = 3