Problem A. Different Consecutive Characters

Time limit	500 ms
Code length Limit	50000 B
OS	Linux

Chef has a binary string S of length N. Chef can perform the following operation on S:

- Insert any character (0 or 1) at any position in S.

Find the minimum number of operations Chef needs to perform so that no two consecutive characters are same in S.

Input Format

- The first line contains a single integer T — the number of test cases. Then the test cases follow.
- The first line of each test case contains an integer N — the length of the binary string S.
- The second line of each test case contains a binary string S of length N containing 0s and 1s only.

Output Format

For each test case, output on a new line the minimum number of operations Chef needs to perform so that no two consecutive characters are same in S.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 1000$

Sample 1

Input	Output
3 2 11 4 0101 5 00100	1 0 2

Test case 1: We can perform the following operations: $11 \rightarrow 1\underline{0}1$.

Test case 2: We do not need to perform any operations.

Test case 3: We can perform the following operations: $00100 \rightarrow 0\underline{1}0100 \rightarrow 0101\underline{0}10$.

Problem B. Apaxiaaaaaaaaaaans!

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

The ancient and mysterious Apaxian civilization, which we most certainly did not make up, continues to confound the researchers at the Oriental Institute. It turns out that the Apaxians had a peculiar naming system: the more letters in your name, the higher your status in society. So, in Apaxian society, `robert` was probably a lowly servant, and `robertapalaxiamethostenes` was likely a High Priest or Minister. Even more than that, Apaxians valued the number of adjacent letters that were the same in a name. So, while `robert` continues to be an unimpressive name, `roooooooooobert` probably elicited cheers and applause wherever he went.

Unfortunately, this makes the task of reading Apaxian scrolls very cumbersome, especially when you consider that a particularly famous Apaxian queen had ten thousand consecutive `a`'s in her name. Legend has it that she was already two years old by the time the Royal Herald finished announcing her birth.

To make the Oriental Institute's life easier, the Department of Computer Science has offered to convert the Apaxian scrolls into a more readable format. Specifically, we will be taking Apaxian names and replacing all consecutive runs of the same letter by a single instance of such letter.

So, for example, the compact version of `roooobert` would be `robert`, where the four consecutive `o`'s have been replaced with a single `o`. Similarly, the compact version of `rrrooobbbert` would also be `robert`. On the other hand, the compact version of `robert` is still `robert`.

Input

The input contains a single name. Each name contains only lowercase letters (`a`–`z`), no whitespace, a minimum length of 1 character, and a maximum length of 250 characters.

Output

The output contains the compact version of the name: any time the same letter appears two or more times in sequence, it must be replaced by a single instance of that letter.

Sample 1

Input	Output
robert	robert

Sample 2

Input	Output
rooobert	robert

Sample 3

Input	Output
roooooobertapalaxxxxios	robertapalaxios

Problem C. Normal Problem

Time limit 1000 ms
Mem limit 262144 kB

A string consisting of only characters 'p', 'q', and 'w' is painted on a glass window of a store. Ship walks past the store, standing directly in front of the glass window, and observes string a. Ship then heads inside the store, looks directly at the same glass window, and observes string b.

Ship gives you string a. Your job is to find and output b.

Input

The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases.

The only line of each test case contains a string a ($1 \leq |a| \leq 100$) — the string Ship observes from outside the store. It is guaranteed that a only contains characters 'p', 'q', and 'w'.

Output

For each test case, output string b, the string Ship observes from inside the store, on a new line.

Examples

Input	Output
5 qwq ppppp pppwwwqqq wqpqwpqwwqp pqpqpqpq	pwp qqqqq pppwwwqqq qpwwpqwpqpw pqpqpqpq

Problem D. Aaah!

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

Jon Marius shouted too much at the recent Justin Bieber concert, and now needs to go to the doctor because of his sore throat. The doctor's instructions are to say "aaah". Unfortunately, the doctors sometimes need Jon Marius to say "aaah" for a while, which Jon Marius has never been good at. Each doctor requires a certain *level* of "aah" – some require "aaaaaah", while others can actually diagnose his throat with just an "h". (They often diagnose wrongly, but that is beyond the scope of this problem.) Since Jon Marius does not want to go to a doctor and have his time wasted, he wants to compare how long he manages to hold the "aaah" with the doctor's requirements. (After all, who wants to be all like "aaah" when the doctor wants you to go "aaaaaah"?)

Each day Jon Marius calls up a different doctor and asks them how long his "aaah" has to be. Find out if Jon Marius would waste his time going to the given doctor.

Input

The input consists of two lines. The first line is the "aaah" Jon Marius is able to say that day. The second line is the "aah" the doctor wants to hear. Only lowercase 'a' and 'h' will be used in the input, and each line will contain between 0 and 999 'a's, inclusive, followed by a single 'h'.

Output

Output "go" if Jon Marius can go to that doctor, and output "no" otherwise.

Sample 1

Input	Output
aaah aaaaah	no

Sample 2

Input	Output
aaah ah	go

Problem E. Anti-Palindrome

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

A palindrome is a sequence of two or more characters that reads the same both forward and backward. Palindromes can appear inside a longer text, such as “Happy days are here again.” in which “pp”, “ehe” and “ere” are some of the palindromes (ignoring spaces, punctuation, and case). An anti-palindrome is a sequence of two or more characters in which there are no palindromes. You are given text that may contain, or itself form, a palindrome. Your task is to determine if there are any palindromes in the text, in which case declare "Palindrome", or if there are none, declare "Anti-palindrome". The text may contain spaces, as well as alphabetic and other non-alphabetic characters. All non-alphabetic characters should be ignored, and uppercase and lowercase letters are considered the same when detecting palindromes.

Input

The input consists of one line of text as specified above. The input line will not be longer than 80 characters, and contains at least one alphabetic character.

Output

Print “Palindrome" if the text contains or is itself a palindrome, or “Anti-palindrome" if there are no palindromes in the text.

Sample 1

Input	Output
`Happy days are here again.`	`Palindrome`

Sample 2

Input	Output
`It is game day.`	`Palindrome`

Sample 3

Input	Output
`It was game day.`	`Anti-palindrome`

Problem F. Alphabet

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

A string of lowercase letters is called alphabetical if some of the letters can be deleted so that the only letters that remain are the letters from 'a' to 'z' in order. Given a string s, determine the minimum number of letters to add anywhere in the string to make it alphabetical.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The only line of input contains a string s ($1 \leq |s| \leq 50$) which contains only lowercase letters.

Output

Output a single integer, which is the smallest number of letters needed to add to s to make it alphabetical.

Sample 1

Input	Output
xyzabcdefghijklmnopqrstuvw	3

Sample 2

Input	Output
aiemckgobjfndlhp	20

Problem G. ABC String

Time limit	1000 ms
Mem limit	2097152 kB
OS	Linux

You're given a string consisting of the characters `A`, `B`, and `C`. The string contains the same count of `A`, `B`, and `C` characters.

A string is *beautiful* if

- Its length is divisible by 3.
- The string can be split evenly into contiguous substrings of size 3, where each substring has one `A`, one `B`, and one `C`, in any order.

For example: `ABCCBA` is a beautiful string, but `ABCAB` and `CCBAAB` are not beautiful.

Given a string, you want to partition it into subsequences (not necessarily contiguous) such that each subsequence is a beautiful string.

For example, for the string `ABACBCAACCBB`, we can do the following:

```
AB   CA C B
  ACB  A C B
```

This partitions the string into two subsequences `ABCACB` and `ACBACB`, both of which are beautiful strings.

For the given string, find the minimum number of subsequences you can partition it into such that each subsequence is beautiful. It can be proven that there is always at least one such partition for all possible inputs that satisfy the input constraints.

Input

The first line of input contains a string s ($3 \leq |s| \leq 3 \cdot 10^5$). $|s|$ is divisible by 3. s contains an equal number of characters `A`, `B`, and `C`.

Output

Output a single integer, which is the minimum subsequences that s can be partitioned into so each subsequence is a beautiful string.

Sample 1

Input	Output
ABACBCAACCBB	2

Problem H. Singularity Cup P2 - Reverse Substring Partitioning

Time limit 1000 ms
Mem limit 262144 kB

You are given a string S of length N, only consisting of the lowercase letters `a-z`.

Let us define an RSP (Reverse Substring Partition) as an operation where you partition S into two non-empty contiguous substrings, reverse both substrings, and merge the two letters that touch in the middle of the partitions into one.

In order to perform an RSP, it is required that during the final step, the last letter of the leftmost partition is the same as the first letter of the rightmost partition so they can be merged. Otherwise, this operation cannot be performed. More formally, you may perform an RSP by partitioning S into two non-empty substrings A and B where $A + B = S$ if and only if the last letter of A when reversed is equal to the first letter of B when reversed.

After performing any number of RSPs, what is the minimum possible length of S?

Constraints

$1 \leq N \leq 10^6$

S only contains lowercase letters from the English alphabet.

Input Specification

The first line of input contains an integer N.

The next line of input contains N lowercase letters representing S.

Output Specification

Output a single integer, the smallest possible resulting length of S.

Input	Output
4 noon	2