# CompressionText

**Input:** `standard input`
**Output:** `standard output`

You are part of a team that is working on a piece of software to handle text compression. Your team has designed the compression algorithm as follows:

To compress a given string of text, two strings, each 3 characters in length, will be chosen as compression keys. The strings may contain any combination of capital letters and/or spaces. Then, a compressed string will be generated from the original such that replacing "`*1`" in the compressed string with the first string and replacing "`*2`" with the second string will recreate the original text.

For example, if the two compression keys are "`FOO`" and "`BAR`", then the compressed string "`*2X*1`" would decompress to "`BARXFOO`", and "`*1*1 *2`" would decompress to "`FOOFOO BAR`".

You have been tasked with writing a proof of concept implementation to test the effectiveness of this compression scheme. Your goal is to optimally select the two compression keys, and generate the compressed text to be as short as possible. You are to print the length of the shortest possible text.

## Input

On the first line one positive number: the number of testcases. After that per testcase:

- One line with a string *original* specifying the original string.

Constraints:

- *original* will contain between 1 and 50 characters, inclusive.
- Each character of *original* will be an uppercase letter ('`A`'-'`Z`') or a space ('` `').

## Output

Per testcase print one line with the length of the shortest possible text.

## Sample Input

```
5
BARXFOO
FOOFOO BAR
ABCDEFGHIJKLMNOPQRSTUVWXYABCDEFGHIJKLMNOPQRSTUVWXY
QWERTYUIOPASDFGHJKLZXCVBNM
BBAAAABBBBAAABABABBBABABAAABABABAAABBAABABBBABAAAB
```

## Sample Output

```
5
7
46
24
40
```