A: ABC String

Time Limit: 1 second, Memory limit: 2G

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 Input 1	Sample Output 1
ABACBCAACCBB	2