Problem

Hanoi Tower Puzzle

Time limit: 8 seconds

Little Cyan Fish has three pegs on his desk, labeled `A`, `B`, and `C`. He also has n disks of distinct sizes, numbered 1 through n from small to large. Initially, all disks are stacked on peg `A`, with smaller disks above larger disks.

Little Cyan Fish wants to transform this initial configuration into a given target configuration. Unlike the classical Tower of Hanoi, this puzzle allows the following special type of move.

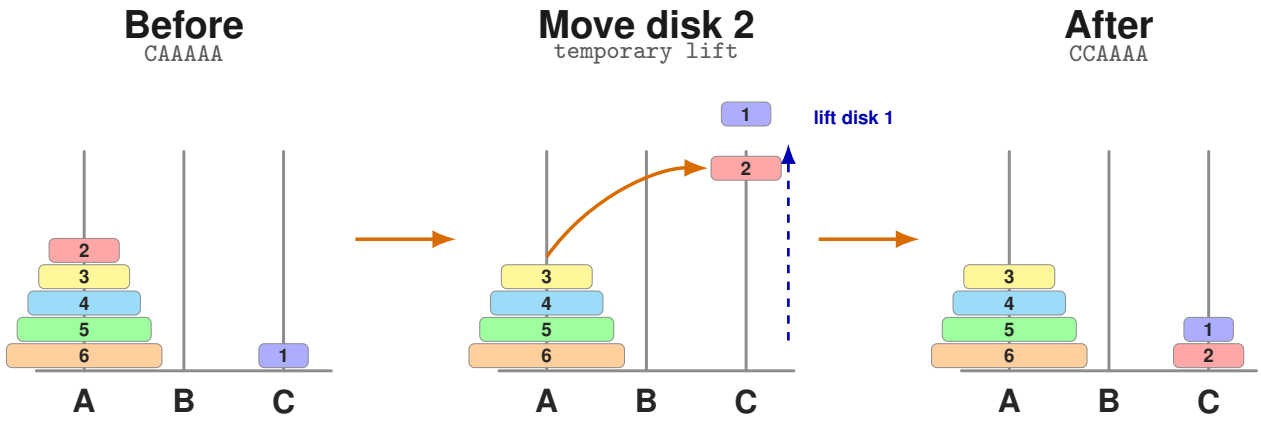
In one move, Little Cyan Fish performs the following operations:

1. Choose a non-empty source peg and remove its top disk;
2. Choose a destination peg that is different from the source peg;
3. On the destination peg, temporarily lift all disks that are smaller than the disk being moved, preserving their original top-to-bottom order;
4. Place the moved disk onto the destination peg;
5. Put the lifted disks back onto the destination peg, preserving their original top-to-bottom order.

Equivalently, a move takes the top disk from a source peg and inserts it into a different destination peg, below all smaller disks and above all larger disks on that destination peg. The relative order of the disks originally on the destination peg does not change.

A configuration can be represented by a string. For a configuration string t, the i-th character of t denotes the peg containing disk i, where disk i is the i-th smallest disk. Since disks on each peg are always ordered with smaller disks above larger disks, such a string uniquely determines the entire configuration.

For example, the following figure illustrates one move from the first sample.



One move in sample 1: `CAAAAA` → `CCAAAA`.

The target configuration is specified by a string s. The i-th character of s is the peg on which disk i must be located in the final configuration.

Help Little Cyan Fish compute the minimum number of moves required to reach the target configuration from the initial configuration.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \leq T$), indicating the number of test cases.

For each test case, the first line of the input contains a non-empty string s consisting only of characters `A`, `B`, and `C`. The length of s is the number of disks in the corresponding test case, and the i-th character of s specifies the peg on which disk i must be located in the final configuration.

It is guaranteed that the sum of $|s|$ over all test cases does not exceed 3×10^6.

Output

For each test case, output a single line containing a single integer, denoting the minimum number of moves required to reach the target configuration from the initial configuration.

Sample Input 1	Sample Output 1
4 BCABCA BCBAB AAAAAAAAABC BACBACBACBACBACBACBACBAC	7 8 20 88

Explanation of Sample 1: For the first test case, one optimal strategy is

$$\texttt{AAAAAA} \to \texttt{CAAAAA} \to \texttt{CCAAAA} \to \texttt{CCBAAA} \to \texttt{CCBBAA} \to \texttt{CCBBCA} \to \texttt{CCABCA} \to \texttt{BCABCA}.$$

For the second test case, one optimal strategy is

$$\texttt{AAAAA} \to \texttt{BAAAA} \to \texttt{BCAAA} \to \texttt{BCBAA} \to \texttt{BCBCA} \to \texttt{BCBCB} \to \texttt{BABCB} \to \texttt{BABAB} \to \texttt{BCBAB}.$$