

As it was mentioned in the previous task, any DNA sequence consists of four bases: adenine (A), cytosine (C), guanine (G) and thymine (T) and can be written as a string constructed from characters A, C, G or T. As strings, DNA sequences can be ordered alphabetically. In order to reduce the amount of memory, a DNA sequence can be described by its length and index in the ordered (index starts from 0) set of all possible DNA sequences of some specific length. So you are asked to write a program that will encode any given DNA sequence in a pair of numbers.

Let's take for example all DNA sequences of length 2. They form the ordered set:

$$\{ AA; AC; AG; AT; CA; CC; CG; CT; GA; GC; GG; GT; TA; TC; TG; TT \}$$

So the sequence "CC" can be described by the pair (2;5) (2 is a length and 5 is an index in the set), "AG" by (2,2), "TG" by (2,14) and so on.

Input

The number of tests T ($T \leq 100$) is given on the first line. Each of the next T lines contains a DNA sequence S of maximal length of 30 characters.

Output

For each test case, output a single line 'Case T : ($A : B$)'. Where T is the test case number (starting from 1) and ($A : B$) is a pair describing the DNA sequence by a given approach.

Sample Input

```
3
AC
ATA
TAGCAGCAGCAGCGAA
```

Sample Output

```
Case 1: (2:1)
Case 2: (3:12)
Case 3: (16:3374617184)
```