G Decompression

Problem

Consider the following scheme for permuting a string of letters: first we create all different rotations of the string, then we sort them in lexicographical order, and finally we take the last letter of each rotation and concatenate them to form a new string [3].

For the string `"LEIDEN."`, for example, the seven rotations in lexicographical order are (where the period `'.'` comes before any letter):

```
.LEIDEN
DEN.LEI
EIDEN.L
EN.LEID
IDEN.LE
LEIDEN.
N.LEIDE
```

so this results in the string `"NILDE.E"`.

At first glance this permutation doesn't seem useful at all, but it has an interesting property. If there are a lot of equal substrings in the original string (which might happen in case of a real language), then a lot of equal consecutive letters occur after the permutation. Therefore the resulting string is very suitable for *block compression*, where blocks of equals letters are replaced by that letter followed by a number which specifies how often that letter occurs. If the letter only occurs once, no number is added. For example the string `"AAABCC"` would be replaced by `"A3BC2"`.

Your task is now to decompress this final string to the original string. Note, however, that permuting the string is not entirely reversible: you can only obtain the original string up to a rotation (i.e., each rotation would lead to the same permuted string). To overcome this, the original string will consist of uppercase letters followed by a single period (`'.'`). This will define the initial rotation.

Input

The first line of the input contains a single number: the number of test cases to follow. Each test case has the following format:

- One line with the compressed string. This string will consist of uppercase letters, numbers and a single period, and will be a valid block compressed string. The length of the original string that resulted in the compressed one will be no more than $1,000,000$ characters.

Output

For every test case in the input, the output should contain a single line with the decompressed string.

Example

Input	Output

`3`
`NILDE.E`
`N13.E13`
`LRSGORTNOMOMAIOSROC2.GAPTE2NS`
	`LEIDEN.`
	`ENENENENENENENENENENENENEN.`
	`PROGRAMMINGCONTESTSARESOCOOL.`

[3] This transformation is called the Burrows-Wheeler transform.