

# Look and Say

## Background

1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, 31131211131221, 13211311123113112211, 11131221133112132113212221, 3113112221232112111312211312113211, ...

This is called the look-and-say sequence. Each successive number is generated by looking at the previous number and saying the size of each group of repeated digits.

### From Wikipedia:

To generate a member of the sequence from the previous member, read off the digits of the previous member, counting the number of digits in groups of the same digit. For example:

- 1 is read off as "one 1" or 11.
- 11 is read off as "two 1s" or 21.
- 21 is read off as "one 2, then one 1" or 1211.
- 1211 is read off as "one 1, then one 2, then two 1s" or 111221.
- 111221 is read off as "three 1s, then two 2s, then one 1" or 312211.

## Description

Luke and Sally really like the look-and-say sequence. Luke likes to look at the numbers, and Sally likes to say them. Because the numbers in this sequence grow very long very quickly, Sally has proposed an alternate sequence that keeps the numbers shorter. Her idea is that instead of looking at groups of repeated digits, you look at the total number of each digit in the entire number, and say those. For example:

- 1 is read off as "one 1" or 11.
- 11 is read off as "two 1s" or 21.
- 21 is read off as "one 1, and one 2" or 1112.
- 1112 is read off as "three 1s, and one 2" or 3112.
- 3112 is read off as "two 1s, one 2, and one 3" or 211213.

Because Luke still prefers the classic look-and-say sequence, they decide to compromise. If the number is EVEN, then they will use the classic method to generate the next number, and if the number is ODD, they will use Sally's new method.

Because this compromise complicates things, they have asked you to write a program to help them keep everything straight. Given an initial number **N**, your program should generate the next number in the sequence according to the Luke-and-Sally compromise.

## Input

The first line contains a single positive integer **T**, representing the number of test cases. The next **T** lines each contain a single integer **N**.

## Limits

$1 \leq T \leq 100$   
 $0 \leq N \leq 10^{100}$

## Output

For each test case, output a single line containing the next number in the sequence.

### Input

```
2
13211311123113112211
13211311123113112212
```

### Output

```
1214243
1113122113311213211321221112
```