I. The Secret Message

Program:	`secret.(cpp|java|py)`
Input:	`secret.in`
Balloon Color:	`Purple`

Description

You have crafted a new way of sharing secret messages with your friends. With your method, the sender sends a string of a mix of alphabet and numeric characters, and the receiver can reconstruct the message without having to know a predefined key. The way this works is that the receiver would use the numeric characters to re-order the alphabet characters, in a rather simple way. As the string is read, digits would appear. If this digit is even, the previous alphabet characters up to the previous digit (or up to the beginning of the text if it's the first digit) would be added to the beginning (left) of the message. If the digit is odd, they would be added to the end (right) of the message. For example, if the secret is "lo8el4Wor7H2ld3", then the message would be "HelloWorld".

Given the secret, write a program to display the message.

Input

The input starts with a number T ($1 \leq T \leq 1{,}000$) that represents the number of test cases in the file. Each test case is on a line that contains a string representing the secret. The maximum number of characters in the string is 100,000, and the minimum is 2 characters. The string always begins with an letter and ends with a digit. The string only contains alphabets and digits and has no blank spaces. Consecutive digits are possible.

Output

The output for each test case is in this form:

```
k. M
```

where ***k*** represents the test case number (starting at 1), and ***M*** is the decoded message.

Sample Input / Output

secret.in

```
2
lo8el4Wor7H2ld3
tM1re4es98sa3Sec0ge7
```

OUTPUT

```
1. HelloWorld
2. SecretMessage
```