

## 4899 The Encoding

Professor Moriarty has devised what he believes is a fiendishly clever code to communicate with his cohorts. At the first level, messages using this code are a sequence of numbers, each of which consists of between 2 and 9 digits (each digit in the range 1 through 9) in ascending and consecutive order (like 2345, or 789). To make things worse, at the second level of encoding he replaces each digit with a capital letter, and these letter sequences are then transmitted as the encoded message.

Luckily, some of the cohorts discarded a slip of paper containing a clue that enables you to decode the numeric sequences into the original messages. But you must still find some way of translating those letter sequences in the corresponding sequences of numbers. Sherlock has asked you to help.

As an example of what's needed, consider the message containing these 8 letter-sequences:

```
RE EW OIU IUYTR YTR TRE EW POIU
```

After manipulating these for a while, Sherlock found they match the digits 1 through 9 as shown in the following table. But it takes too long to do messages by hand, so a program to do the job is required.

```
1 2 3 4 5 6 7 8 9
P O I U Y T R E W
```

### Input

There will be multiple input cases to consider. The input for each case begins with a line containing an integer  $N$  which specifies the number of letter-sequences in a message.  $N$  will be no larger than 100. On the next  $N$  lines are those letter-sequences, each of which consists of between 2 and 9 uppercase alphabetic characters immediately followed by the end of line. The input for the last case is followed by a line containing a single zero.

### Output

For each case, first display the case number. If it is impossible to uniquely determine the mapping from letters to digits, display the word 'Ambiguous'. Otherwise, display the mapping from letters to digits as shown in the sample below, displaying a blank line after the output for each case.

### Sample Input

```
5
GL
OWIZXUGL
WIZX
WI
ZX
8
RE
EW
OIU
IUYTR
```

YTR  
TRE  
EW  
POIU  
0

### Sample Output

Case 1:

Ambiguous

Case 2:

1 2 3 4 5 6 7 8 9  
P O I U Y T R E W