Google Kickstart '17 Round C Problem A - Ambiguous Cipher

Time limit: 2.0s **Memory limit:** 64M

Susie and Calvin are classmates. Calvin would like to be able to pass notes to Susie in class without their teacher or other classmates knowing what they are talking about, just in case the notes fall into the wrong hands. Calvin has devised a system to encrypt his messages.

Calvin only passes one word to Susie each time, and that word consists of only uppercase letters, because Calvin is so excited to talk to Susie. Each word is encrypted as follows:

- Calvin assigns a number to each letter based on the letter's position in the alphabet, where `A` = 0, `B` = 1, ..., `Z` = 25.
- For every letter in the word, Calvin determines the encrypted value of the letter by summing the values of the 1 or 2 letter(s) that are adjacent to that letter in the word. He takes that sum modulo 26, and this is the new value of the letter. Calvin then converts the value back to an uppercase letter based on positions in the alphabet, as before.
- The encrypted word is determined by encrypting every letter in the word using this method. Each letter's encryption is based only on the letters from the original unencrypted message, and not on any letters that have already been encrypted.

Let's take a look at one of the notes Calvin is writing for Susie. Since Calvin is always hungry, he wants to let Susie know that he wants to eat again. Calvin encrypts the word `SOUP` as follows:

- `S` = 18, `O` = 14, `U` = 20, and `P` = 15.
- Calvin encrypts each letter based on the values of its neighbor(s):
  - First letter: $14 \bmod 26 = 14$.
  - Second letter: $(18 + 20) \bmod 26 = 12$.
  - Third letter: $(14 + 15) \bmod 26 = 3$.
  - Fourth letter: $20 \bmod 26 = 20$.
- The values 14 12 3 20 correspond to the letters `OMDU`, and this is the encrypted word that Calvin will write on the note for Susie.

It is guaranteed that Calvin will not send Susie any words that cannot be decrypted at all. For example, Calvin would not send Susie the word `APE`, since it does not have any valid decryptions. (That is, there is no word that Calvin could have encrypted to `APE`.)

However, Calvin's system is not perfect, and some of the words he sends Susie can actually be decrypted to multiple words, creating ambiguity! For example, `BCB` can be decrypted to `ABC` or `CBA`, among other possibilities.

Susie pulled another all-nighter yesterday to finish school projects, and she is too tired to decrypt Calvin's messages. She needs your help!

Input Specification

The first line of the input gives the number of test cases, T. T test cases follow. Each case is a single line that contains a string W of uppercase letters: an encrypted word that Calvin has sent.

Output Specification

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the decrypted word, or `AMBIGUOUS` if it is impossible to uniquely determine the decrypted word.

Limits

For all subtasks,

- $1 \leq T \leq 100$.
- W consists of only uppercase English letters.
- W is decryptable to one or more words. (That is, W is the result of an encryption of some word.)
- W does not decrypt to the word `AMBIGUOUS`. (You will only output that when the decryption is ambiguous.)

For 7 of the 19 available marks,

$2 \leq$ the length of $W \leq 4$.

For the remaining 12 of the 19 available marks,

$2 \leq$ the length of $W \leq 50$.

Sample Input

```
3
OMDU
BCB
AOAAAN
```

Sample Output

```
Case #1: SOUP
Case #2: AMBIGUOUS
Case #3: BANANA
```

Note that the last sample case would not appear in the Small dataset.

Sample Cases #1 & #2 were explained in the problem statement.

In Sample Case #3, `BANANA` is the only word that encrypts to `AOAAAN`.