K: `kewl` Texting

Source file name: `kewl.c`, `kewl.cpp`, `kewl.java`, *or* `kewl.py`
Author: Ana Echavarría

Alicia and Roberto are good friends and often text each other. They would like to speed up their texting by designing a mechanism that suggests what word they may want to text next. That way, they can select the suggested word instead of having to type it directly, saving some time.

Being computer science students who just learned about language models, they came up with a solution. For each occurring word w in their past text messages, they have selected a word w' as the suggestion to be shown after typing w. The criteria for selecting w' are the following:

- The word w' is the word that comes after w the most times in the past text messages.
- If two or more words come after w the same number of times in the past texts, then w' is the word among them that appeared the most times overall in the past texts.
- If again, two or more words appeared the same number of times in the past texts, then w' is the first one of them in lexicographical order.
- They also want to suggest words that start or end a text message. Therefore, the start and end of texts (called here {*start*} and {*end*}, respectively) are also considered words. When they sort words lexicographically, {*start*} would come before {*end*} and they would both come before any other word in the texts.

For example, consider the following text messages (one per line):

1. `what a nice day`
2. `this is the nice restaurant he talked about`
3. `we want nice weather and hope it is a nice day`

There are three words that follow the word `nice`, namely, `day`, `weather`, and `restaurant`. However, `day` comes after `nice` 2 times, while `restaurant` and `weather` do so only once. Based on their model, the word suggested after typing `nice` is the word `day`. Similarly, the word `is` is followed once by both `a` and `the`, but `a` is more frequent in the texts overall (it appears 2 times while `the` appears only once); therefore, the suggestion for `is` is the word `a`. The suggestion for `day` is {*end*} since, in most cases, it is used to end a text. On the other hand, the words `what`, `this`, and `we` all come at the start of sentences and they all appear only once in the texts: hence, `this` (the first one in lexicographical order of the three) is the suggestion for {*start*}, i.e., for the beginning of a text message.

Alicia and Roberto would like to know what the sentence generated by always picking the word suggested by their language model would be. Note that such a text message either ends after a finite number of suggestions or never ends. For the example above, the sentence resulting by always picking the model's suggestion is

```
this is a nice day
```

This is because the suggested starting word is `this`, the word suggested after `this` is `is` and so on, until the word `day` is reached (whose suggestion is {*end*}, that is, the end of the text).

Your task is, for a given history of text messages between Alicia and Roberto, to compute the sentence generated by always picking the word suggested by the language model or identify if such a process never terminates.

Input

The input consist of several test cases. Each test case begins with a line containing a single integer *n* defining the number of text messages. Each of the next *n* lines represents a text message containing at least one word. Every word in the input consists only of lowercase English letters. There is no limit on the number of text messages. However, the entire set of text messages has at most 10^5 words in total and each word will have at most 20 characters.

The input must be read from standard input.

Output

For each test case, output a single line containing the sentence that would result by always selecting the word suggested by the language model or `INFINITE` if this method would result in an infinite text message. Use a single blank to separate each pair of consecutive words in the output.

The output must be written to standard output.

Sample Input	**Sample Output**

`3`
`what a nice day`
`this is the nice restaurant he talked about`
`we want nice weather and hope it is a nice day`
`3`
`a rabbit is white`
`the house is big and tall`
`we have a big house with a door`
`1`
`a rose is a rose`
	`this is a nice day`
	`INFINITE`
	`a rose`