

7690 As Easy as CAB

We all know how to alphabetize a list when you know the alphabet: One word may be a prefix of another longer word, in which case the shorter word always comes before the longer word. With any other two words there must be a first place in the words where their letters differ. Then the order of the words is determined by the lexicographical order of these first differing letters.

How about the reverse problem: Can you find the lexicographic order of the alphabet from an ordered list of words? Suppose an alphabet exists where the following list of word strings is given in lexicographical order:

```
cab
cda
ccc
badca
```

It is clear that *c* comes before *b* in the underlying alphabet because *cab* comes before *badca*. Similarly, we know *a* comes before *d*, because *cab* < *cda*, *a* comes before *c* because *cab* < *ccc*, and *d* comes before *c* because *cda* < *ccc*. The only ordering of the 4 alphabet characters that is possible is *adcb*.

However, it may be that a list contains inconsistencies that make it impossible to be ordered under any proposed alphabet. For example, in the following list it must be that *a* comes before *b* in the alphabet since *abc* < *bca*, yet it also must be that *b* comes before *a* in the alphabet since *bca* < *aca*.

```
abc
bca
cab
aca
```

Finally, some lists may not provide enough clues to derive a unique alphabet order, such as the following:

```
dea
cfb
```

In this list, *d* comes before *c* but we don't know about the relative positions of any of the other letters, so we are unable to uniquely discern the order of the alphabet characters.

Input

The input file contains several test cases, each of them as described below.

The first line of input will contain *L* and *N*, separated by a space, where *L* is a lowercase character $b \leq L \leq z$ representing the highest character in the traditional alphabet that appears in the derived alphabet, and *N* is an integer $1 \leq N \leq 1000$ that is equal to the number of strings in the list. Each of the next *N* lines will contain a single nonempty string of length at most 1 000, consisting only of characters in the derived alphabet. No two strings will be the same.

Output

For each test case, write to the output one single line that must follow the description below.

If the input is consistent with a unique ordering of the alphabet, output a string that designates that ordered alphabet. If the data is inconsistent with any ordering, output 'IMPOSSIBLE'. If the data is consistent with multiple orderings, output 'AMBIGUOUS'.

Sample Input

```
d 4
cab
cda
ccc
badca
c 4
abc
bca
cab
aca
f 2
dea
cfb
e 5
ebbce
dbe
adcd
bc
cd
```

Sample Output

```
adcb
IMPOSSIBLE
AMBIGUOUS
edabc
```