

# Problem J

## Joust Sort

We are going to organise the letters of the word using a novel algorithm based on the natural instincts of the hoarse chestnut tree: a *grand tournament* between its constituent letters.

We will pick several pairs of letters and have them face off. The result of a match between these letters determines their relative order. For example, if  $a < b$ , then all occurrences of  $a$  in a well-organised word should come before all occurrences of  $b$  in that word. If there is no case for  $a < b$  or  $b < a$ , then the individual letters  $a$  and  $b$  may intersperse among each other in any order.

Given a word, and several such matches, determine a fair rearrangement of the word to meet all of the constraints given. It is possible that multiple such rearrangements exists, or that none exist at all.

### Input

- One line containing the number of orders to follow,  $n$  ( $1 \leq n \leq 700$ ).
- $n$  further lines, each containing a distinct ordering of two lowercase letters  $a$  and  $b$  separated by a  $<$  or  $>$  character and spaces.
- One final line containing the word to improve,  $s$  ( $1 \leq |s| \leq 100,000$ ).

### Output

Output a sorted version of the word, or IMPOSSIBLE if the word cannot be sorted according to all of the rules at once.

In case there are multiple answers, you may output any one of them.

#### Sample Input 1

```
3
m > i
n < i
i > o
minion
```

#### Sample Output 1

```
noniim
```

#### Sample Input 2

```
1
b < n
banana
```

#### Sample Output 2

```
banana
```

**Sample Input 3**

```
6
b < a
a > b
n < b
a > b
n < b
a > b
bananaman
```

**Sample Output 3**

```
nnnbaaama
```

**Sample Input 4**

```
2
a < b
b < a
unsolvable
```

**Sample Output 4**

```
IMPOSSIBLE
```