Good Memories

You and your friends just returned from a fantastic trip to Europe. During the trip, members of your group went on x excursions to different attractions. You decide to make a scrapbook of all the good memories, but you can't remember the sequence in which you visited each attraction during each excursion. You ask your friends to write down the sequence of attractions they remember visiting during each excursion — in order (note that your friends may not recall all the attractions visited on an excursion). Next, you want to take the notes for each individual excursion and determine whether or not you all agree on the order of visited attractions. For example, consider the following two notes:

1. `Eiffel tower,Olympus,Disneyland`
2. `Eiffel tower,Oktoberfest,Disneyland`

In this example, one of your friends forgot about visiting Olympus and the other friend forgot about visiting Oktoberfest; however, both of your friends agree on a weak order (Eiffel tower first, Disneyland last). Because a weak order exists and there are no contradictions in the sequence of events, the group is in agreement that an *order exists*.

It's also important to note that your friends may disagree on the order in which they visited attractions. For example, consider the following two notes:

1. `Disneyland,Eiffel tower`
2. `Eiffel tower,Disneyland`

In this example, your friends disagree on the order in which they visited Disneyland and the Eiffel tower. Because there is a conflict in the sequence of events, the *order is violated*.

Given the notes for x excursions, determine whether or not the group of friends that went on an excursion agrees on the sequence of attractions visited. For each excursion, print whether or not everyone agrees on the sequence of events. If the information given in the notes contains direct conflicts, print `ORDER VIOLATION` on a new line; otherwise, print `ORDER EXISTS`.

Input Format

The first line contains a single integer, x, denoting the number of excursions.
The subsequent lines describe each excursion in the following form:

1. The first line contains an integer, n, denoting the number of friends that went on the excursion.
2. Each line i of the n subsequent lines contains a single string of comma-separated attraction names specifying the sequence of attractions that friend i recalls visiting.

Constraints

- $1 \leq x \leq 20$
- $1 \leq n \leq 200$
- The total number of attractions visited during a single excursion is ≤ 1000.
- It is guaranteed that an attraction name contains a maximum of 50 characters.

Output Format

For each of the x excursions, print whether or not the group of friends who went on the excursion agrees on the sequence in which they visited the attractions. If everyone is in agreement, print `ORDER EXISTS` on a new line; otherwise, print `ORDER VIOLATION` on a new line.

Sample Input

```
2
3
Red square,Colosseum
Louvre,Red square
Louvre
3
Sacre Coeur,The Hermitage
Stonehenge,Versailles,Louvre
Louvre,Stonehenge
```

Sample Output

```
ORDER EXISTS
ORDER VIOLATION
```

Explanation

For the first excursion, the only possible sequence of visited attractions is *Louvre → Red square → Colosseum*. Because everyone is in agreement, we print *ORDER EXISTS* on a new line.

For the second excursion, your friends do not agree on the order in which you visited the *Louvre* and *Stonehenge*. Because everyone is *not* in agreement, we print *ORDER VIOLATION* on a new line.