Challenge 3 - Lost in Lost

The writers of the Lost series have a serious problem. Due to its immense popularity, they have written several hundred seasons. Almost every chapter has some *flashbacks* or *flashforwards* that display scenes of other chapters of other seasons, that may have been written, or may not have.

In order to write a book of the complete series, they want to have all scenes arranged chronologically, but they are lost in their scripts (what a shame!). They have hired you to solve this.

Input

First a number **N** of scripts to process is given. Then, **N** scripts follow, each one in a different line.

Each script consists of a sequence of described scenes. Each scene is delimited by a dot (**.**), a less-than (**<**) or a greater-than (**>**) symbol:

- Scenes beginning with the dot are chronologically ordered.
- If the scene begins with a **<** symbol, then it's a flashback during the previous (chronologically ordered) scene. That is, it happened before the previous scene.
- If the scene begin with a **>** sign, then it's a flashforward during the previous (chronologically ordered) scene. That is, it happened after the previous scene.

Example: .A>B<C is C,A,B

There may not be any possible chronological order for a given script, or there may be more than one if the scriptwriters have made enough of a mess.

Output

For each script, if there is only one possible chronological order, output the list of scenes described, ordered chronologically and separated by commas.

If the script has a finite number (greater than 1) of possible chronological orders, return 'valid'.

If no chronological order is possible, or there is more than one beginning or end scene, or there is neither beginning nor end scene, then return 'invalid'.

Sample input

```
3
.john gets into the plane<john sees paul at the airport.john tells mia they are going
to die>the plane crashes
.john gets into the plane<john sees paul at the airport.john tells mia they are going
to die>the plane crashes.the plane takes off.mia sees through the window an isolated
island<the plane crashes
.john gets into the plane<john sees paul at the airport.john tells mia they are going
```

```
to die>the plane crashes.john sees paul at the airport
```

Sample output

```
john sees paul at the airport,john gets into the plane,john tells mia they are going
to die,the plane crashes
valid
invalid
```

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

```
Challenge token: drCNPACjH9pg1LUfQrwS
```

To test your program

```
./test_challenge drCNPACjH9pg1LUfQrwS path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To submit your program to the challenge

```
./submit_challenge drCNPACjH9pg1LUfQrwS path/source_pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

Important: In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **you can only submit once!** Once your solution is submitted you won't be able to amend it to fix issues or make it faster, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the info section.

Go ahead

I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

I'm stuck! :(

Be sure you follow the Tuenti Engineering twitter for updates and possible hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	Skip this challenge :((You have 4 skips)

Refresh status

Tweet about this! #TuentiChallenge3

Share Follow @Tuentieng