G: Riverboat Tours

Tour operators are setting up a sequence of sites along a system of rivers for sight-seeing. They want the tours to begin and end at a given site. The tour may go into various tributaries (branches) but never repeat any stretch of river other than obviously having to return by the same route, noting that river systems do not have any loops. You task is to help the operators verify that a planned sequence of sites is feasible, meaning that it is possible to make the tour using the sequence with no skipping or backtracking. So if one passes site A while going upstream, site A will be listed in the sequence at the proper location AND appear a second time during the return trip at the proper location. In fact, it would appear a third time if the site is at the confluence of two streams and the tour continues up the second stream. The only reason a site would be listed only once is if it is at the extreme end of a branch so that the boat reaches it and immediately turns around. Note that there is no reason to continue a tour beyond the farthest site on branch, so there will always be such a single-listed site for each branch. At each confluence of two branches such that the tour takes both branches upstream, there will always be a site.

Input

There may be multiple prospective tours to verify. The first line will contain an integer indicating that number. Each tour will be described using just one line. The line will contain a string of upper-case letters, each of which represents a site on the proposed tour. The string represents the sequence to be verified.

Output

For each case, present the tour number followed by either *valid* or *oops* according to if the sequence is valid.

Sample Input

```
2
ABCBEFEBA
ABCBEFEBAA
```

Sample Output

```
Tour 1: valid
Tour 2: oops
```