Competitive Programming SS24

Submit until end of contest

Problem: joust (1.0 second timelimit)

Note: This is a problem that is harder to solve than usual. Solve the other problems first before spending too much time on this one.

Your king expressed a longing for a tasteful joust. As his loyal servant, you will do best to schedule one this afternoon. His highness is not blessed with a lot of patience. During a tournament he is most pleased if after each battle the victor is bested by an even stronger competitor. This convinces him of the ever increasing finesse of his underlings. To honor his unprecedented glory, the schedule must reflect the royal wishes. You must find a sequence of knights that involves all great heroes and where each knight will fight the winner of the last match. The problem is, that only the crown prince — the kingdoms most accomplished expert regarding bets and gambling — in his everlasting youth knows for each pair of knights who will prevail. Unfortunately he is not a boy of patience either. He grants you a generous number of inquiries; eleven times the number of knights!

Interaction First, you are given the number of knights $2 \leq n \leq 1000$. Then you are left talking with his highness' son. Query for a match by printing `? a b`, where $1 \leq a, b \leq n$ represent two knights. He will answer with the most noble of responses; namely `0` if `a` wins and `1` if `b` wins[1]. But do not test his patience! If annoyed, he will only respond with a disrespectful `WRONG-ANSWER`. I highly advise you stop asking then.

Be aware, that proud knights cannot simply be ranked in an absolute ordering (winning is not transitive). When you have a schedule, print one line starting with `!` followed by n space separated knight ids.

It can be shown that a valid schedule always exists. However, your audience will be adversarial. That is, the prince will see to it that everything you have not explicitly queried will turn out against you.

Example

```
> 3
< ? 1 2
> 1
< ? 1 3
> 0
< ! 1 3 2
```

Note that the outcome of the first match is irrelevant for the king.

[1] read his answer from `stdin`, and do not forget to `flush` your queries