Problem D. johnny

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 2 points *2 minute timeout*	The contest is finished.
large 30 points *10 minute timeout*	The contest is finished.

Don't know what distributed problems are about? See our guide.

Problem

You and Johnny play a very simple card game. Both players have a deck of cards. Both draw a card at random from their deck, and the player with the better card wins. "Better" is a complex concept, but for any two cards, you (and Johnny) know which one is better, with no ties allowed. Note that this is not necessarily transitive: if card A is better than B, and B is better than C, it is possible that C is better than A.

Johnny is very, very unhappy when he loses. So, you would like to make sure that he will win. You have N cards, and you want to split them into two non-empty decks (with no cards left over), one for you and one for Johnny, so that whatever card you draw from your deck and whatever card Johnny draws from his, Johnny will win.

If it is possible to do this in multiple ways, you want Johnny's deck to be as large as possible (as long as your deck is not empty).

Input

The input library will be called "johnny"; see the sample inputs below for examples in your language. It will define two methods: NumberOfCards(), which will return the number of cards, and IsBetter(i, j), which will return true if card i is better than card j, for $0 \le i, j <$ NumberOfCards(). If called with $i=j$, it will return false. One call to IsBetter will take approximately 0.03 microseconds.

Output

If it is possible to split the cards so that Johnny will always win, output one number: the largest possible size of Johnny's deck. If it is impossible to split the cards in such a way, output the word `IMPOSSIBLE`.

Limits

Each node will have access to 256MB of RAM, and a time limit of 3 seconds.

Small input

Your solution will run on 10 nodes.
$2 \le$ NumberOfCards() ≤ 1000.

Large input

Your solution will run on 100 nodes.
$2 \le$ NumberOfCards() $\le 2 \times 10^4$.

Sample

Input	Output
See sample input files below.	For sample input 1: 1 For sample input 2: IMPOSSIBLE For sample input 3: 4

Sample input libraries:
Sample input for test 1: johnny.h [CPP] johnny.java [Java]
Sample input for test 2: johnny.h [CPP] johnny.java [Java]
Sample input for test 3: johnny.h [CPP] johnny.java [Java]