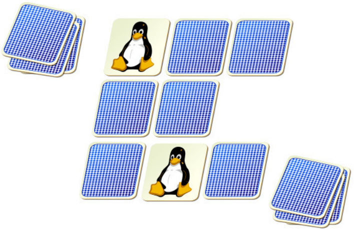
A Adversarial Memory



Charlie is playing a game of *memory* (also known as *concentration*) on his own. The game consists of $2n$ cards, where each of the numbers from 1 to n is written on exactly two cards. The cards are upside down on the table. In a turn, Charlie turns over one card, looks at it, and then turns over another card. If the cards have the same number, they are removed from the game. Otherwise, they are turned back over and placed back. The goal of the game is to remove all the cards from the game in as few moves as possible.

You are a magician, and hence you are able to change the numbers on upside down cards seamlessly. If Charlie turns over the same card twice, you need to make sure that he sees the same number both times, or else Charlie would notice something is wrong. You also need to make sure that for each number there will be exactly two cards on which Charlie will see that number. Your goal is to force Charlie to need at least $2n - 1$ turns to finish the game.

More formally, there are $2n$ indices from 1 to $2n$. In a turn, Charlie chooses an index i and turns over the card at index i. You can then decide what number Charlie will see when he turns over the card. Then Charlie will choose a different index j and turn over the card at index j. You can then decide what number Charlie will see when he turns over that card. The only restrictions are that Charlie must always see the same number when he turns over the same card, and that for each number there will be exactly two cards on which Charlie will see that number. Note that Charlie will never choose the index of a card that is already out of the game.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

The interactor first sends a line containing the integer n ($1 \leq n \leq 10\,000$), the number of different values on the cards.

Following this, $2n - 1$ turns will be played. Each turn consists of two parts:

- The interactor first sends a line containing an integer i ($1 \leq i \leq 2n$), denoting the index of the first card Charlie turns over. Your program must respond with the number on card i.
- The interactor then sends a line containing an integer j ($1 \leq j \leq 2n$), denoting the index of the second card Charlie turns over. Your program must respond with the number on card j.

For each turn, it is guaranteed that i and j are distinct indices of cards that are still in the game.

Your submission should exit after it has answered $2n-1$ turns (i.e. printed $4n-2$ numbers). Reading more input will result in a time limit exceeded and printing more output will result in a wrong answer.

Make sure you flush the buffer after each write.

A testing tool is provided to help you develop your solution.

Sample Interaction 1

Read	Write
`1` `1`	
	`1`
`2`	
	`1`

Sample Interaction 2

Read	Write

`3`
`1`	

	`1`
`2`	
	`2`
`3`	
	`2`
`2`	
	`2`
`4`	
	`3`
`5`	
	`1`
`5`	
	`1`
`1`	
	`1`
`4`	
	`3`
`6`	
	`3`