

L Lock In

Time limit: 10s

Audrey is playing a game ranking her favourite 100 competitive programmers in a top 10 list. She already has objectively ranked all 100 of them. Now, one by one, she is given 10 distinct programmers uniformly at random, and must rank them among each other, in a relative top 10 list. Once she has made a choice, she is not allowed to move them any more, and she must stick to her choice. She cannot bear her assignments list being incorrect, so she tries to give a correct list as often as possible. Of course, she knows that it is impossible to always be right, so she is satisfied if she is correct at least 1% of the time. She is not allowed to assign the same spot to two different people. Help Audrey fill in her top 10 lists.



The podium for the best programmers.
CC BY-SA 4.0 by Santeri Viinamäki
on Wikimedia Commons

Interaction

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

Each game, exactly 10 times, the interactor will send one line with an integer n ($1 \leq n \leq 100$), the objective rank Audrey assigned to a competitive programmer. Then, your program should print one line with an integer r ($1 \leq r \leq 10$), the spot in the relative ranking Audrey should assign this programmer to.

Writing the same value of r twice in the same game will result in a wrong answer.

The 10 objective ranks given in each game are pairwise distinct.

After each game, the interactor will print one line with either “win” or “loss”, depending on whether your list is correct or not. Note that, in each game, exactly 10 programmers need to be assigned a relative rank, even when it is clear that your list can no longer be correct.

Your submission will be run on exactly one test case, in which the interactor will play exactly 10 000 games. Winning fewer than 100 of these games will result in a wrong answer. Note that the sample interaction on the next page shows only two games for illustration purposes only.

The interactor is not adaptive: for each of your submissions, the 10 objective ranks of each independent game are uniformly randomly generated, and do not depend on your queries.

Make sure you flush the buffer after each write.

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

Read	Sample Interaction 1	Write
1		1
		1
2		2
		2
3		3
		3
4		4
		4
5		5
		5
6		6
		6
7		7
		7
8		8
		8
9		9
		9
10		10
		10
win		
26		3
		3
3		1
		1
8		2
		2
50		5
		5
46		4
		4
52		6
		6
92		10
		10
95		9
		9
84		8
		8
42		7
		7
loss		