Problem L
Lateral Damage
Time limit: 2 seconds

You are playing *Battleships* in a large ocean with large ships. More precisely, there is a large square grid of size at most 100×100 and inside it are up to 10 of the largest type of ship in Battleships – the aircraft carrier – which has a length of five tiles, placed either horizontally or vertically. The ships do not overlap, but they are allowed to be adjacent to each other. See Figure L.1 for an example.

Unfortunately, your opponent appears to bend the rules to their liking. It looks like they do not always determine the placement of their ships before you start shooting. You are not impressed by their attempt at cheating, and decide to try and win the game anyway.



The original *Battleships* game, before the upgrade to a 100×100 grid.
CC BY-NC 3.0 by Pavel Ševela on Wikimedia Commons

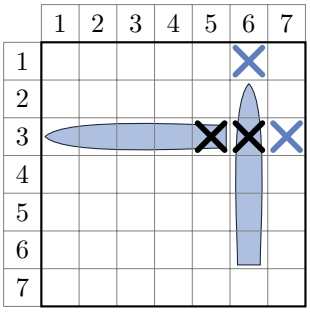


Figure L.1: Illustration of Sample Interaction 1 after the first four shots were fired.

Your goal is to locate and sink all your opponent's aircraft carriers in at most 2500 shots, that is, you must hit each of the five tiles of all their ships.

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:

The interactor first sends one line with two integers n and k ($5 \leq n \leq 100$, $1 \leq k \leq 10$), the size of the grid and the number of ships. It is guaranteed that it is possible to place k aircraft carriers in the grid without overlap.

Then, your program needs to start firing shots. Each shot is fired by printing one line of the form "x y" ($1 \leq x, y \leq n$), indicating you shoot at position (x, y). The interactor will respond with "`hit`" if the shot was a hit, "`sunk`" if the shot caused an aircraft carrier to sink, and "`miss`" otherwise. If you have shot the same location before, the response will be "`miss`".

Once you sink the last aircraft carrier, the interaction will stop and your program must exit.

The interactor is adaptive: the positions of the ships may be determined during the interaction, and may depend on where you decide to shoot.

Make sure you flush the buffer after each write.

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

Firing more than 2500 shots will result in a wrong answer.

Read	Sample Interaction 1	Write
`7 2`		
		`6 1`
`miss`		
		`6 3`
`hit`		
		`7 3`
`miss`		
		`5 3`
`hit`		
		`4 3`
`hit`		
		`3 3`
`hit`		
		`2 3`
`hit`		
		`1 3`
`sunk`		
		`6 7`
`miss`		
		`6 7`
`miss`		
		`6 2`
`hit`		
		`6 2`
`miss`		
		`6 4`
`hit`		
		`6 5`
`hit`		
		`6 6`
`sunk`		