Container Scheduling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In a busy port, a large number of containers need to be stacked on the deck of cargo ships every day. For convenience, you can consider both the deck and the containers as rectangles parallel to the coordinate axes. The deck is a large rectangular area with the lower left corner at $(0, 0)$ and the upper right corner at (l, h).

You are a container scheduler and need to arrange the position of each container on the deck. In theory, you need to schedule it reasonably to maximize space utilization. However, you feel that you are underpaid, so you want to be lazy in the scheduling method. The scheduling method you provide is as follows:

- Due to height restrictions, containers can only be stacked in a single layer without overlapping.
- Since you don't have the budget to buy rotating cranes, containers cannot be rotated when placed.
- Containers follow the first-come, first-served order, meaning that containers at the front are placed first, and containers at the back are placed later, without changing the order of the containers. If a container cannot be placed during the process, it is discarded.
- When selecting a position to place a container, prioritize the position with the smallest x coordinate that can accommodate it. If there are multiple candidate positions, choose the one with the smallest y coordinate.

There are a total of n containers, and for each container, the length along the x axis and the width along the y axis are given in order. Your task is to determine whether each container can be placed, and if so, provide the position of its lower left corner.

Input

The first line contains three integers n, l, h ($1 \leq n \leq 50$, $1 \leq l, h \leq 10^9$), representing the number of containers and the length and width of the deck area, respectively.

The next n lines each contain two integers x, y ($1 \leq x, y \leq 10^9$), representing the length along the x axis and the width along the y axis of each container.

Output

For each container in order, output the coordinates of its lower left corner separated by a space if the container can be placed. Otherwise, output -1.

Example

standard input	standard output
4 10 10 5 5 6 6 4 7 10 2	0 0 -1 5 0 0 7