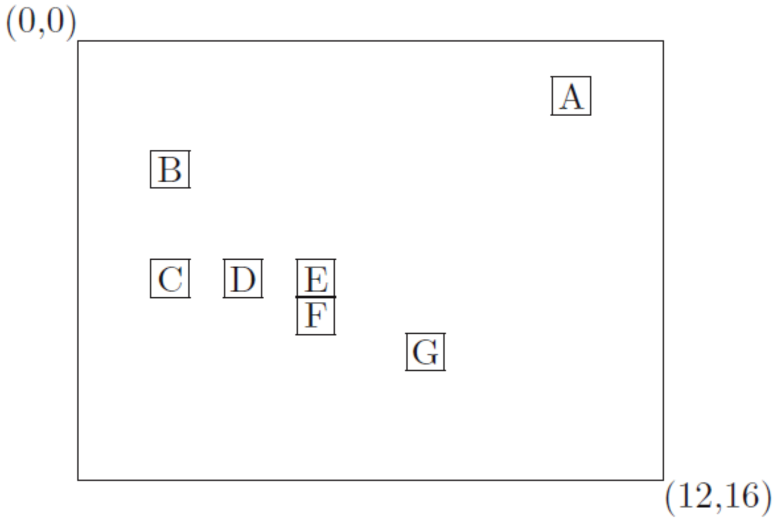
I: Pushing Boxes

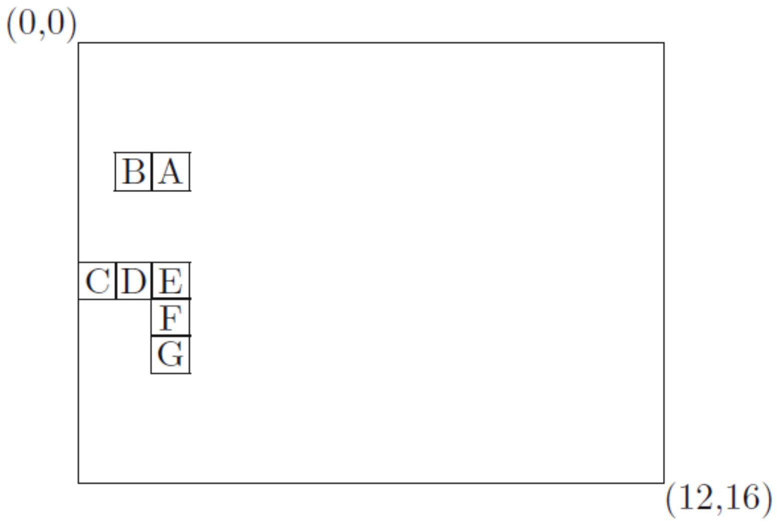
Time Limit: 1 second(s)

Scrap cars in a junk yard are crushed in a device that pushes the car in from the sides, from the front and back, and from the top and bottom. The result is a compact little chunk of metal. In this problem you're going to model a device that works in a similar manner, but doesn't crush anything, only pushes boxes around in two dimensions. The boxes are all square with unit length on a side and are situated on the floor of a room. Each wall of the room can be programmed to move inward a certain amount, pushing any boxes it may bump into. Unlike the car-crusher, this device is sensitive and if it senses that boxes are stacked up against a wall and that it might crush them if pressed any farther, it will stop.

For example, suppose we have boxes arranged in a 12-by-16 room as shown below. The upper left-hand corners of the boxes (which is how we will locate them in this problem) are at coordinates (1,13) (box A below), (3,2), (6,2), (6,4), (6,6), (7,6) and (8,9) (box G), where the first coordinate indicates distance from the top wall and the second coordinate indicates distance from the left wall.



Suppose the top wall is programmed to move down 3 units (then retreats, as the walls always will) and then the right wall is programmed to move left 14 units. The first operation can be performed with no problem, but the second one can not be carried out without crushing some boxes. Therefore, the right wall will move only 13 units, the maximum distance it can move until boxes are packed tightly between it and the left wall. The boxes will then be in the configuration shown in the following figure. The locations of the boxes are (3,1), (3,2), (6,0), (6,1), (6,2), (7,2), (8,2).



Input

The input consists of one test instance. The first line of the data set will be two integers giving the height and width of the room. (We will visualize the room as if on a piece of paper, as drawn above.) Each dimension will be no more than 20. The next line will contain an integer $n (0 < n \leq 10)$ followed by n pairs of integers, each pair giving the location of a box as the distances from the top and the left walls of the room. The following lines will be of the form `direction` m, where `direction` is either `down`, `left`, `up`, `right`, or `done` and m is a positive integer. For example, `left 2` would mean to try to move the right wall 2 spaces to the left. The "direction" `done` indicates that you are finished pushing this set of boxes around. There will be no integer m following the direction `done`, of course.

Output

You are to produce one line of output of the form:

`Boxes are at locations` (r_1, c_1) (r_2, c_2) $\ldots$ (r_n, c_n).

where the (r_i, c_i) are the locations of the boxes given from top-to-bottom, left-to-right, (separated by one space).

Sample Input and Output

Sample Input 1

```
12 16
7 1 13 3 2 6 2 6 4 6 6 7 6 8 9
down 3
left 14
done
```

Output for Sample Input

```
Boxes are at locations (3,1) (3,2) (6,0) (6,1) (6,2) (7,2) (8,2).
```

Sample Input 2

```
4 4
3 1 0 2 1 2 3
right 3
up 2
left 1
done
```

Output for Sample Input

```
Boxes are at locations (0,2) (1,1) (1,2).
```