B. Warehouse

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: input.txt

output: output.txt

Once upon a time, when the world was more beautiful, the sun shone brighter, the grass was greener and the sausages tasted better Arlandia was the most powerful country. And its capital was the place where our hero DravDe worked. He couldn't program or make up problems (in fact, few people saw a computer those days) but he was nevertheless happy. He worked in a warehouse where a magical but non-alcoholic drink Ogudar-Olok was kept. We won't describe his work in detail and take a better look at a simplified version of the warehouse.

The warehouse has one set of shelving. It has n shelves, each of which is divided into m sections. The shelves are numbered from top to bottom starting from 1 and the sections of each shelf are numbered from left to right also starting from 1. Each section can contain exactly one box of the drink, and try as he might, DravDe can never put a box in a section that already has one. In the course of his work DravDe frequently notices that he has to put a box in a filled section. In that case his solution is simple. DravDe ignores that section and looks at the next one to the right. If it is empty, he puts the box there. Otherwise he keeps looking for the first empty section to the right. If no empty section is found by the end of the shelf, he looks at the shelf which is under it, then the next one, etc. Also each time he looks at a new shelf he starts from the shelf's beginning. If DravDe still can't find an empty section for the box, he immediately drinks it all up and throws the empty bottles away not to be caught.

After one great party with a lot of Ogudar-Olok drunk DravDe asked you to help him. Unlike him, you can program and therefore modeling the process of counting the boxes in the warehouse will be easy work for you.

The process of counting contains two types of query messages:

- «`+1 x y id`» (where x, y are integers, $1 \le x \le n$, $1 \le y \le m$, and id is a string of lower case Latin letters — from 1 to 10 characters long). That query means that the warehouse got a box identified as id, which should be put in the section y on the shelf x. If the section is full, use the rules described above. It is guaranteed that every moment of the process the identifiers of all the boxes in the warehouse are different. You don't have to answer this query.
- «`-1 id`» (where id is a string of lower case Latin letters — from 1 to 10 characters long). That query means that a box identified as id is removed from the warehouse. You have to answer this query (see output format).

Input

The first input line contains integers n, m and k ($1 \le n, m \le 30$, $1 \le k \le 2000$) — the height, the width of shelving and the amount of the operations in the warehouse that you need to analyze. In the following k lines the queries are given in the order of appearance in the format described above.

Output

For each query of the «`-1 id`» type output two numbers in a separate line — index of the shelf and index of the section where the box with this identifier lay. If there was no such box in the warehouse when the query was made, output «`-1 -1`» without quotes.

Examples

input

```
2 2 9
+1 1 1 cola
+1 1 1 fanta
+1 1 1 sevenup
+1 1 1 whitekey
-1 cola
-1 fanta
-1 sevenup
-1 whitekey
-1 cola
```

output

```
1 1
1 2
2 1
2 2
-1 -1
```

input

```
2 2 8
+1 1 1 cola
-1 cola
+1 1 1 fanta
-1 fanta
+1 1 1 sevenup
-1 sevenup
+1 1 1 whitekey
-1 whitekey
```

output

```
1 1
1 1
1 1
1 1
```