--- Day 13: Mine Cart Madness ---

A crop of this size requires significant logistics to transport produce, soil, fertilizer, and so on. The Elves are very busy pushing things around in carts on some kind of rudimentary system of tracks they've come up with.

Seeing as how cart-and-track systems don't appear in recorded history for another 1000 years, the Elves seem to be making this up as they go along. They haven't even figured out how to avoid collisions yet.

You map out the tracks (your puzzle input) and see where you can help.

Tracks consist of straight paths (`|` and `-`), curves (`/` and `\`), and intersections (`+`). Curves connect exactly two perpendicular pieces of track; for example, this is a closed loop:

```
/----\
|    |
|    |
\----/
```

Intersections occur when two perpendicular paths cross. At an intersection, a cart is capable of turning left, turning right, or continuing straight. Here are two loops connected by two intersections:

```
/-----\
|     |
|  /--+--\
|  |  |  |
\--+--/  |
   |     |
   \-----/
```

Several carts are also on the tracks. Carts always face either up (`^`), down (`v`), left (`<`), or right (`>`). (On your initial map, the track under each cart is a straight path matching the direction the cart is facing.)

Each time a cart has the option to turn (by arriving at any intersection), it turns left the first time, goes straight the second time, turns right the third time, and then repeats those directions starting again with left the fourth time, straight the fifth time, and so on. This process is independent of the particular intersection at which the cart has arrived - that is, the cart has no per-intersection memory.

Carts all move at the same speed; they take turns moving a single step at a time. They do this based on their current location: carts on the top row move first (acting from left to right), then carts on the second row move (again from left to right), then carts on the third row, and so on. Once each cart has moved one step, the process repeats; each of these loops is called a tick.

For example, suppose there are two carts on a straight track:

```
|  |  |  |  |
v  |  |  |  |
|  v  v  |  |
|  |  |  v  X
|  |  ^  ^  |
^  ^  |  |  |
|  |  |  |  |
```

First, the top cart moves. It is facing down (v), so it moves down one square. Second, the bottom cart moves. It is facing up (^), so it moves up one square. Because all carts have moved, the first tick ends. Then, the process repeats, starting with the first cart. The first cart moves down, then the second cart moves up - right into the first cart, colliding with it! (The location of the crash is marked with an X.) This ends the second and last tick.

Here is a longer example:

```
/->-\
|   |  /----\
| /-+--+-\  |
| | |  | v  |
\-+-/  \-+--/
  \------/

/-->\
|   |  /----\
| /-+--+-\  |
| | |  | |  |
\-+-/  \->--/
  \------/

/---v
|   |  /----\
| /-+--+-\  |
| | |  | |  |
\-+-/  \-+>-/
  \------/

/---\
|   v  /----\
| /-+--+-\  |
| | |  | |  |
\-+-/  \-+->/
  \------/

/---\
|   |  /----\
| /->--+-\  |
| | |  | |  |
\-+-/  \-+--^
  \------/

/---\
|   |  /----\
| /-+>-+-\  |
| | |  | |  ^
\-+-/  \-+--/
  \------/

/---\
|   |  /----\
| /-+->+-\  ^
| | |  | |  |
\-+-/  \-+--/
  \------/

/---\
|   |  /----<
| /-+-->-\  |
| | |  | |  |
\-+-/  \-+--/
```

```
  \------/

/---\
|   |  /---<\
| /-+--+>\  |
| | |  | |  |
\-+-/  \-+--/
  \------/

/---\
|   |  /--<-\
| /-+--+-v  |
| | |  | |  |
\-+-/  \-+--/
  \------/

/---\
|   |  /-<--\
| /-+--+-\  |
| | |  | v  |
\-+-/  \-+--/
  \------/

/---\
|   |  /<---\
| /-+--+-\  |
| | |  | |  |
\-+-/  \-<--/
  \------/

/---\
|   |  v----\
| /-+--+-\  |
| | |  | |  |
\-+-/  \<+--/
  \------/

/---\
|   |  /----\
| /-+--v-\  |
| | |  | |  |
\-+-/  ^-+--/
  \------/

/---\
|   |  /----\
| /-+--+-\  |
| | |  X |  |
\-+-/  \-+--/
  \------/
```

After following their respective paths for a while, the carts eventually crash. To help prevent crashes, you'd like to know the location of the first crash. Locations are given in `X,Y` coordinates, where the furthest left column is `X=0` and the furthest top row is `Y=0`:

```
           111
 0123456789012
0/---\
1|   |  /----\
2| /-+--+-\  |
3| | |  X |  |
4\-+-/  \-+--/
5  \------/
```

In this example, the location of the first crash is `7,3`.

Your puzzle answer was `91,69`.

--- Part Two ---

There isn't much you can do to prevent crashes in this ridiculous system. However, by predicting the crashes, the Elves know where to be in advance and instantly remove the two crashing carts the moment any crash occurs.

They can proceed like this for a while, but eventually, they're going to run out of carts. It could be useful to figure out where the last cart that hasn't crashed will end up.

For example:

```
/>-<\
|   |
| /<+-\
| | | v
\>+</ |
  |   ^
  \<->/

/---\
|   |
| v-+-\
| | | |
\-+-/ |
  |   |
  ^---^

/---\
|   |
| /-+-\
| v | |
\-+-/ |
  ^   ^
  \---/

/---\
|   |
| /-+-\
| | | |
\-+-/ ^
  |   |
  \---/
```

After four very expensive crashes, a tick ends with only one cart remaining; its final location is `6,4`.

What is the location of the last cart at the end of the first tick where it is the only cart left?

Your puzzle answer was `44,87`.

Both parts of this puzzle are complete! They provide two gold stars: **

At this point, you should return to your advent calendar and try another puzzle.

If you still want to see it, you can get your puzzle input.

You can also [Share] this puzzle.