

Our **sponsors** help
make Advent of
Code possible:

Sixfold - The first purpose-built AI solution for underwriters. Our engineers solve real-world insurance challenges at scale with LLMs.

just to make copy-pasting easier. Newlines within the move sequence should be ignored.)

Here is a smaller example to get started:

```
#####  
#..0.0.#  
##@.0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####  
  
<^^>>>vv<v>>v<<
```

Were the robot to attempt the given sequence of moves, it would push around the boxes as follows:

Initial state:

```
#####  
#..0.0.#  
##@.0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move <:

```
#####  
#..0.0.#  
##@.0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move ^:

```
#####  
#.@0.0.#  
##..0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move ^:

```
#####  
#.@0.0.#  
##..0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move >:

```
#####  
#..@00.#  
##..0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move >:

```
#####  
#...@00#  
##..0..#  
#...0..#  
#.#.0..#  
#...0..#  
#.....#  
#####
```

Move >:

```
#####  
#...@00#  
##..0..#
```

#...0..#
#.#.0..#
#...0..#
#.....#
#####

Move v:

#...00#
##..@..#
#...0..#
#.#.0..#
#...0..#
#...0..#
#####

Move v:

#...00#
##..@..#
#...0..#
#.#.0..#
#...0..#
#...0..#
#####

Move <:

#...00#
##.@...#
#...0..#
#.#.0..#
#...0..#
#...0..#
#####

Move v:

#...00#
#.....#
#..@0..#
#.#.0..#
#...0..#
#...0..#
#####

Move >:

#...00#
#.....#
#...@0.#
#.#.0..#
#...0..#
#...0..#
#####

Move >:

#...00#
#.....#
#....@0#
#.#.0..#
#...0..#
#...0..#

```
#####
Move v:
#####
#...00#
##.....#
#.....0#
#.#.0@.#
#...0..#
#...0..#
#####

Move <:
#####
#...00#
##.....#
#.....0#
#.#0@..#
#...0..#
#...0..#
#####

Move <:
#####
#...00#
##.....#
#.....0#
#.#0@..#
#...0..#
#...0..#
#####
```

The larger example has many more moves; after the robot has finished those moves, the warehouse would look like this:

```
#####
#.0.0.000#
#.....#
#00.....#
#00@.....#
#0#.....0#
#0.....00#
#0.....00#
#00....00#
#####
```

The lanternfish use their own custom Goods Positioning System (GPS for short) to track the locations of the boxes. The GPS coordinate of a box is equal to 100 times its distance from the top edge of the map plus its distance from the left edge of the map. (This process does not stop at wall tiles; measure all the way to the edges of the map.)

So, the box shown below has a distance of 1 from the top edge of the map and 4 from the left edge of the map, resulting in a GPS coordinate of $100 * 1 + 4 = 104$.

```
#####
#...0..
#.....
```

The lanternfish would like to know the sum of all boxes' GPS coordinates after the robot finishes moving. In the larger example, the sum of all boxes' GPS coordinates is 10092. In the smaller example, the sum is 2028.

Predict the motion of the robot and boxes in the warehouse. After the robot is finished moving, what is the sum of all boxes' GPS coordinates?

Your puzzle answer was `1371036`.

--- Part Two ---

The lanternfish use your information to find a safe moment to swim in and turn off the malfunctioning robot! Just as they start preparing a festival in your honor, reports start coming in that a second warehouse's robot is also malfunctioning.

This warehouse's layout is surprisingly similar to the one you just helped. There is one key difference: everything except the robot is twice as wide! The robot's list of movements doesn't change.

To get the wider warehouse's map, start with your original map and, for each tile, make the following changes:

- If the tile is `#`, the new map contains `##` instead.
- If the tile is `O`, the new map contains `OO` instead.
- If the tile is `.`, the new map contains `..` instead.
- If the tile is `@`, the new map contains `@.` instead.

This will produce a new warehouse map which is twice as wide and with wide boxes that are represented by `[]`. (The robot does not change size.)

The larger example from before would now look like this:

```
#####
##....[]....[]..[]##
##.....[]..##
##..[] []....[]..[]##
##....[]@.....[]..##
##[]##....[].....##
##[]....[]....[]..##
##..[] []..[]..[] []##
##.....[].....##
#####
```

Because boxes are now twice as wide but the robot is still the same size and speed, boxes can be aligned such that they directly push two other boxes at once. For example, consider this situation:

```
#####
#...#.#
#.....#
#..OO@#
#..O..#
#.....#
#####
<VV<<^^<<^^
```

After appropriately resizing this map, the robot would push around these boxes as follows:

```
Initial state:
#####
##.....##..##
##.....##
##...[] []@.##
##...[] .....##
##.....##
#####

Move <:
#####
##.....##..##
##.....##
##...[] []@.##
##...[] .....##
##.....##
#####

Move v:
#####
##.....##..##
##.....##
##...[] [] .....##
##...[] .@.##
##.....##
#####

Move v:
#####
##.....##..##
##.....##
##...[] [] .....##
##...[] .....##
##.....@.##
#####

Move <:
#####
##.....##..##
##.....##
##...[] [] .....##
##...[] .....##
##.....@.##
#####

Move <:
#####
##.....##..##
##.....##
##...[] [] .....##
##...[] .....##
##.....@.##
#####

Move ^:
#####
##.....##..##
##...[] [] .....##
##...[] .....##
##.....@.##
##.....##
#####

Move ^:
```

```
#####
##.....##.##
##...[] []...##
##...[] .....##
##.....@.....##
##.....##
#####

Move <:
#####
##.....##.##
##...[] []...##
##...[] .....##
##...@.....##
##.....##
#####

Move <:
#####
##.....##.##
##...[] []...##
##...[] .....##
##...@.....##
##.....##
#####

Move ^:
#####
##.....##.##
##...[] []...##
##...@[] .....##
##.....##
##.....##
#####

Move ^:
#####
##...[] .##.##
##...@.[] .....##
##...[] .....##
##.....##
##.....##
#####
```

This warehouse also uses GPS to locate the boxes. For these larger boxes, distances are measured from the edge of the map to the closest edge of the box in question. So, the box shown below has a distance of 1 from the top edge of the map and 5 from the left edge of the map, resulting in a GPS coordinate of $100 * 1 + 5 = 105$.

```
#####
##...[]...
##.....
```

In the scaled-up version of the larger example from above, after the robot has finished all of its moves, the warehouse would look like this:


```
#####
##[] .....[] [] []##
##[] .....[] .##
##[] .....[] [] []##
##[] .....[] ....[]##
##..## .....[] ....##
##..[] .....##
##..@ .....[] [] []##
##.....[] [] ..[] ..##
#####
```

The sum of these boxes' GPS coordinates is 9021.

Predict the motion of the robot and boxes in this new, scaled-up warehouse.
What is the sum of all boxes' final GPS coordinates?

Your puzzle answer was 1392847.

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.