

[Advent of Code](#)
[\[About\]](#)
[\[AoC++\]](#)
[\[Events\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
 Roland Tritsch ([AoC++](#)) 30\*  
[{'year':2017}](#)
[\[Calendar\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)
[\[Sponsors\]](#)

--- Day 22: Sporifica Virus ---

Diagnostics indicate that the local grid computing cluster has been contaminated with the Sporifica Virus. The grid computing cluster is a seemingly-infinite two-dimensional grid of compute nodes. Each node is either clean or infected by the virus.

To **prevent overloading** the nodes (which would render them useless to the virus) or detection by system administrators, exactly one virus carrier moves through the network, infecting or cleaning nodes as it moves. The virus carrier is always located on a single node in the network (the current node) and keeps track of the direction it is facing.

To avoid detection, the virus carrier works in bursts; in each burst, it wakes up, does some work, and goes back to sleep. The following steps are all executed in order one time each burst:

- If the current node is infected, it turns to its right. Otherwise, it turns to its left. (Turning is done in-place; the current node does not change.)
- If the current node is clean, it becomes infected. Otherwise, it becomes cleaned. (This is done after the node is considered for the purposes of changing direction.)
- The virus carrier **moves** forward one node in the direction it is facing.

Diagnostics have also provided a map of the node infection status (your puzzle input). Clean nodes are shown as `.`; infected nodes are shown as `#`. This map only shows the center of the grid; there are many more nodes beyond those shown, but none of them are currently infected.

The virus carrier begins in the middle of the map facing up.

For example, suppose you are given a map like this:

```

..#
#..
...
  
```

Then, the middle of the infinite grid looks like this, with the virus carrier's position marked with `[ ]`:

```

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

The virus carrier is on a clean node, so it turns left, infects the node, and moves left:

```

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

The virus carrier is on an infected node, so it turns right, cleans the node, and moves up:

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

**Cheppers** -  
 xor(Pz0pQUI7Ch  
 cmER8YDAEYAh4L  
 GwEP,  
 ↑↑↓↓←→←→BA)

By popular demand, there a now AoC-themed objects availab (until Jan. 3rd Get them shippe from the US or from Europe.

```

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

```

Four times in a row, the virus carrier finds a clean, infects it, turns left, and moves forward, ending in the same place and still facing up:

```

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

```

Now on the same node as before, it sees an infection, which causes it to turn right, clean the node, and move forward:

```

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

```

After the above actions, a total of `7` bursts of activity had taken place. Of them, `5` bursts of activity caused an infection.

After a total of `70`, the grid looks like this, with the virus carrier facing up:

```

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

```

By this time, `41` bursts of activity caused an infection (though most of those nodes have since been cleaned).

After a total of `10000` bursts of activity, `5587` bursts will have caused an infection.

Given your actual map, after `10000` bursts of activity, how many bursts cause a node to become infected? (Do not count nodes that begin infected.)

Your puzzle answer was `5305`.

The first half of this puzzle is complete! It provides one gold star: \*

--- Part Two ---

As you go to remove the virus from the infected nodes, it evolves to resist your attempt.

Now, before it infects a clean node, it will weaken it to disable your defenses. If it encounters an infected node, it will instead flag the node to be cleaned in the future. So:

- Clean nodes become weakened.
- Weakened nodes become infected.
- Infected nodes become flagged.
- Flagged nodes become clean.

Every node is always in exactly one of the above states.

The virus carrier still functions in a similar way, but now uses the following logic during its bursts of action:

- Decide which way to turn based on the current node:
  - If it is clean, it turns left.
  - If it is weakened, it does not turn, and will continue moving in the same direction.
  - If it is infected, it turns right.
  - If it is flagged, it reverses direction, and will go back the way it came.
- Modify the state of the current node, as described above.
- The virus carrier moves forward one node in the direction it is facing.

Start with the same map (still using `.` for clean and `#` for infected) and still with the virus carrier starting in the middle and facing up.

Using the same initial state as the previous example, and drawing weakened as `W` and flagged as `F`, the middle of the infinite grid looks like this, with the virus carrier's position again marked with `[.]`:

```

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

```

This is the same as before, since no initial nodes are weakened or flagged. The virus carrier is on a clean node, so it still turns left, instead weakens the node, and moves left:

```

. . . . .
. . . . .
. . . # .
. . . [.] W .
. . . . .
. . . . .
. . . . .

```

The virus carrier is on an infected node, so it still turns right, instead flags the node, and moves up:

```

. . . . .
. . . . .
. . . [.] # .
. . . F W .
. . . . .
. . . . .
. . . . .

```

This process repeats three more times, ending on the previously-flagged node and facing right:

```

. . . . .
. . . . .
. . W W . # . . .
. . W[F]W . . . .
. . . . .
. . . . .
. . . . .

```

Finding a flagged node, it reverses direction and cleans the node:

```

. . . . .
. . . . .
. . W W . # . . .
. . [W]. W . . . .
. . . . .
. . . . .
. . . . .

```

The weakened node becomes infected, and it continues in the same direction:

```

. . . . .
. . . . .
. . W W . # . . .
. [. ]# . W . . . .
. . . . .
. . . . .
. . . . .

```

Of the first `100` bursts, `26` will result in infection. Unfortunately, another feature of this evolved virus is speed; of the first `100000000` bursts, `2511944` will result in infection.

Given your actual map, after `100000000` bursts of activity, how many bursts cause a node to become infected? (Do not count nodes that begin infected.)

Although it hasn't changed, you can still [get your puzzle input](#).

Answer:  [\[Submit\]](#)

You can also [\[Share\]](#) this puzzle.