

[Advent of Code](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[josianed 5*](#)
[/2023/](#)
[\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)

--- Day 12: Hot Springs ---

You finally reach the hot springs! You can see steam rising from secluded areas attached to the primary, ornate building.

As you turn to enter, the `researcher` stops you. "Wait - I thought you were looking for the hot springs, weren't you?" You indicate that this definitely looks like hot springs to you.

"Oh, sorry, common mistake! This is actually the `onsen`! The hot springs are next door."

You look in the direction the researcher is pointing and suddenly notice the massive metal helixes towering overhead. "This way!"

It only takes you a few more steps to reach the main gate of the massive fenced-off area containing the springs. You go through the gate and into a small administrative building.

"Hello! What brings you to the hot springs today? Sorry they're not very hot right now; we're having a lava shortage at the moment." You ask about the missing machine parts for Desert Island.

"Oh, all of Gear Island is currently offline! Nothing is being manufactured at the moment, not until we get more lava to heat our forges. And our springs. The springs aren't very springy unless they're hot!"

"Say, could you go up and see why the lava stopped flowing? The springs are too cold for normal operation, but we should be able to find one springy enough to launch you up there!"

There's just one problem - many of the springs have fallen into disrepair, so they're not actually sure which springs would even be safe to use! Worse yet, their condition records of which springs are damaged (your puzzle input) are also damaged! You'll need to help them repair the damaged records.

In the giant field just outside, the springs are arranged into rows. For each row, the condition records show every spring and whether it is operational (`.`) or damaged (`#`). This is the part of the condition records that is itself damaged; for some springs, it is simply unknown (`?`) whether the spring is operational or damaged.

However, the engineer that produced the condition records also duplicated some of this information in a different format! After the list of springs for a given row, the size of each contiguous group of damaged springs is listed in the order those groups appear in the row. This list always accounts for every damaged spring, and each number is the entire size of its contiguous group (that is, groups are always separated by at least one operational spring: `####` would always be `4`, never `2,2`).

So, condition records with no unknown spring conditions might look like this:

```
#.##### 1,1,3
.#...#....###. 1,1,3
.#####.##### 1,3,1,6
####.#...#... 4,1,1
#.....#####. 1,6,5
.#####.###...# 3,2,1
```

However, the condition records are partially damaged; some of the springs'

Our sponsors help
make Advent of
Code possible:

[Boot.dev](#) - Ready
to become a
backend
developer? If you
like AoC, you
might be like us.
We think smartest
way to learn to
code is to ensure
you're never
bored. Try the
most captivating,
addictive way to
learn to code on
[Boot.dev](#).

```
??.?## 1,1,3  
.?..??...?##. 1,1,3  
#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?#?  
????.#...#... 4,1,1  
????.#####.#####. 1,6,5  
?###??????? 3,2,1
```

In the first line (`???### 1,1,3`), there is exactly one way separate groups of one, one, and three broken springs (in that order) can appear in that row: the first three unknown springs must be broken, then operational, then broken (`#.#`), making the whole row `#.###`.

The last line is actually consistent with ten different arrangements! Because the first number is **3**, the first and second **?** must both be **.** (if either were **#**, the first number would have to be **4** or higher). However, the remaining run of unknown spring conditions have many different ways they could hold groups of two and one broken springs:

?###??????? 3,2,1
.###.##.#...
.###.##..#.
.###.##..#. .
.###.##...#
.###..##.##.
.###..##..#.
.###..##...#
.###...##.##.
.###...##...#
.###...##.##

- ???## 1,1,3 - 1 arrangement
- .??..??...?##. 1,1,3 - 4 arrangements
- ?#?#?#?#?#?#?#?#?# 1,3,1,6 - 1 arrangement
- ????.#...#... 4,1,1 - 1 arrangement
- ????.######...##### 1,6,5 - 4 arrangements
- ?####?#?#?#?#?#?#?#?#?# 3,2,1 - 10 arrangements

To begin, **get your puzzle input.**

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