

[Advent of Code](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[Jarek 6★](#)  
[/^2022\\$/](#)
[\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)

--- Day 4: Camp Cleanup ---

Space needs to be cleared before the last supplies can be unloaded from the ships, and so several Elves have been assigned the job of cleaning up sections of the camp. Every section has a unique ID number, and each Elf is assigned a range of section IDs.

However, as some of the Elves compare their section assignments with each other, they've noticed that many of the assignments overlap. To try to quickly find overlaps and reduce duplicated effort, the Elves pair up and make a big list of the section assignments for each pair (your puzzle input).

For example, consider the following list of section assignment pairs:

```
2-4,6-8
2-3,4-5
5-7,7-9
2-8,3-7
6-6,4-6
2-6,4-8
```

For the first few pairs, this list means:

- Within the first pair of Elves, the first Elf was assigned sections `2-4` (sections `2`, `3`, and `4`), while the second Elf was assigned sections `6-8` (sections `6`, `7`, `8`).
- The Elves in the second pair were each assigned two sections.
- The Elves in the third pair were each assigned three sections: one got sections `5`, `6`, and `7`, while the other also got `7`, plus `8` and `9`.

This example list uses single-digit section IDs to make it easier to draw; your actual list might contain larger numbers. Visually, these pairs of section assignments look like this:

```
.234..... 2-4
.....678. 6-8

.23..... 2-3
...45.... 4-5

....567.. 5-7
.....789 7-9

.2345678. 2-8
..34567.. 3-7

.....6... 6-6
...456... 4-6

.23456... 2-6
...45678. 4-8
```

Some of the pairs have noticed that one of their assignments fully contains the other. For example, `2-8` fully contains `3-7`, and `6-6` is fully contained by `4-6`. In pairs where one assignment fully contains the other, one Elf in the pair would be exclusively cleaning sections their partner will already be cleaning, so these seem like the most in need of reconsideration. In this example, there are `2` such pairs.

In how many assignment pairs does one range fully contain the other?

Our [sponsors](#) help make Advent of Code possible:

[Ahrefs](#) - Work on the next general purpose search engine, a world-class crawler, and real big data. Leveraging bleeding-edge hardware and advanced programming technologies. From anywhere in the world. OCaml, ReasonML, Dlang, C++

To begin, [get your puzzle input](#).

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

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