

[Advent of Code](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[otodreas 4\\*](#)  
[/2025/](#)
[\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboards\]](#)
[\[Stats\]](#)

--- Day 2: Gift Shop ---

You get inside and take the elevator to its only other stop: the gift shop. "Thank you for visiting the North Pole!" gleefully exclaims a nearby sign. You aren't sure who is even allowed to visit the North Pole, but you know you can access the lobby through here, and from there you can access the rest of the North Pole base.

As you make your way through the surprisingly extensive selection, one of the clerks recognizes you and asks for your help.

As it turns out, one of the younger Elves was playing on a gift shop computer and managed to add a whole bunch of invalid product IDs to their gift shop database! Surely, it would be no trouble for you to identify the invalid product IDs for them, right?

They've even checked most of the product ID ranges already; they only have a few product ID ranges (your puzzle input) that you'll need to check. For example:

```
11-22,95-115,998-1012,1188511880-1188511890,222220-222224,
1698522-1698528,446443-446449,38593856-38593862,565653-565659,
824824821-824824827,2121212118-2121212124
```

(The ID ranges are wrapped here for legibility; in your input, they appear on a single long line.)

The ranges are separated by commas (`,`); each range gives its first ID and last ID separated by a dash (`-`).

Since the young Elf was just doing silly patterns, you can find the invalid IDs by looking for any ID which is made only of some sequence of digits repeated twice. So, `55` (`5` twice), `6464` (`64` twice), and `123123` (`123` twice) would all be invalid IDs.

None of the numbers have leading zeroes; `0101` isn't an ID at all. `101` is a valid ID that you would ignore.)

Your job is to find all of the invalid IDs that appear in the given ranges. In the above example:

- `11-22` has two invalid IDs, `11` and `22`.
- `95-115` has one invalid ID, `99`.
- `998-1012` has one invalid ID, `1010`.
- `1188511880-1188511890` has one invalid ID, `1188511885`.
- `222220-222224` has one invalid ID, `222222`.
- `1698522-1698528` contains no invalid IDs.
- `446443-446449` has one invalid ID, `446446`.
- `38593856-38593862` has one invalid ID, `38593859`.
- The rest of the ranges contain no invalid IDs.

Adding up all the invalid IDs in this example produces `1227775554`.

What do you get if you add up all of the invalid IDs?

Your puzzle answer was `19386344315`.

--- Part Two ---

The clerk quickly discovers that there are still invalid IDs in the ranges in your list. Maybe the young Elf was doing other silly patterns as well?

Our sponsors help  
make Advent of  
Code possible:

Axis - 404: Your application not found! <embedded, cloud, Machine learning, fullstack> Save our Christmas - apply now and join our team!

Now, an ID is invalid if it is made only of some sequence of digits repeated at least twice. So, `12341234` (`1234` two times), `123123123` (`123` three times), `1212121212` (`12` five times), and `1111111` (`1` seven times) are all invalid IDs.

From the same example as before:

- `11-22` still has two invalid IDs, `11` and `22`.
- `95-115` now has two invalid IDs, `99` and `111`.
- `998-1012` now has two invalid IDs, `999` and `1010`.
- `1188511880-1188511890` still has one invalid ID, `1188511885`.
- `222220-222224` still has one invalid ID, `222222`.
- `1698522-1698528` still contains no invalid IDs.
- `446443-446449` still has one invalid ID, `446446`.
- `38593856-38593862` still has one invalid ID, `38593859`.
- `565653-565659` now has one invalid ID, `565656`.
- `824824821-824824827` now has one invalid ID, `824824824`.
- `2121212118-2121212124` now has one invalid ID, `2121212121`.

Adding up all the invalid IDs in this example produces `4174379265`.

What do you get if you add up all of the invalid IDs using these new rules?

Your puzzle answer was `34421651192`.

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.