

[Advent of Code](#) [\[About\]](#) [\[Events\]](#) [\[Shop\]](#) [\[Settings\]](#) [\[Log Out\]](#) Florescu Eduard 20*
[\\$year=2021;](#) [\[Calendar\]](#) [\[AoC++\]](#) [\[Sponsors\]](#) [\[Leaderboard\]](#) [\[Stats\]](#)

--- Day 10: Syntax Scoring ---

You ask the submarine to determine the best route out of the deep-sea cave, but it only replies:

Syntax error in navigation subsystem on line: all of them

All of them?! The damage is worse than you thought. You bring up a copy of the navigation subsystem (your puzzle input).

The navigation subsystem syntax is made of several lines containing **chunks**. There are one or more chunks on each line, and chunks contain zero or more other chunks. Adjacent chunks are not separated by any delimiter; if one chunk stops, the next chunk (if any) can immediately start. Every chunk must **open** and **close** with one of four legal pairs of matching characters:

- If a chunk opens with `[`, it must close with `]`.
- If a chunk opens with `{`, it must close with `}`.
- If a chunk opens with `(`, it must close with `)`.
- If a chunk opens with `<`, it must close with `>`.

So, `[]` is a legal chunk that contains no other chunks, as is `[][]`. More complex but valid chunks include `()[()]`, `{()()()}`, `<([{}])>`, `[<>({}){}][()<>]`, and even `(((((((((((()())))))))))))`.

Some lines are **incomplete**, but others are **corrupted**. Find and discard the corrupted lines first.

A corrupted line is one where a chunk **closes with the wrong character** – that is, where the characters it opens and closes with do not form one of the four legal pairs listed above.

Examples of corrupted chunks include `[]`, `{()()()>`, `(((())])`, and `<([]){()}{[]}]`. Such a chunk can appear anywhere within a line, and its presence causes the whole line to be considered corrupted.

For example, consider the following navigation subsystem:

```
[{(<()[])>[[{}{<()<>>
[()<>]]({[<[[]>>(
{[(<{}<>[]]>[[]{[(<()>
((({<>}<{<[<>]}[[]{}
[[<[[]])<[[]{}[()]]]
[{}{({}){}}([{}{({}){}}([
{<[[]]>}<{[[]{}[(){}[[]
[<(<(<(<{}))><[[]([()
<{([([(<>()){}]}><<{
<{([{}]}<[[]<>{}]]>[[]]
```

Some of the lines aren't corrupted, just incomplete; you can ignore these lines for now. The remaining five lines are corrupted:

- `{([(<{}<>[]]>[[]{[(<()>]` - Expected `]`, but found `]` instead.
- `[[<[[]])<[[]{}[()]]]` - Expected `]`, but found `]` instead.
- `[{}{({}){}}([{}{({}){}}([` - Expected `]`, but found `]` instead.
- `[<(<(<(<{}))><[[]([()` - Expected `>`, but found `]` instead.
- `<{([([(<>()){}]}><<{` - Expected `]`, but found `>` instead.

Stop at the first incorrect closing character on each corrupted line.

Did you know that syntax checkers actually have contests to see who can get the high score for syntax errors in a file? It's true! To calculate the

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

[Retool](#) - Build internal apps remarkably fast. Drag and drop a form together, and have it POST back to your API in minutes. Write JavaScript anywhere to customize. Deploy instantly with access controls and audit logs.

- `()}>[]}]` - `5566` total points.
- `}}>}>)))]` - `1480781` total points.
- `[]}]}}[]}]>` - `995444` total points.
- `[]}]>` - `294` total points.

Autocomplete tools are an odd bunch: the winner is found by sorting all of the scores and then taking the middle score. (There will always be an odd number of scores to consider.) In this example, the middle score is `288957` because there are the same number of scores smaller and larger than it.

Find the completion string for each incomplete line, score the completion strings, and sort the scores. What is the middle score?

Your puzzle answer was `1685293086`.

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.