

Advent of Code [About] [AoC++] [Events] [Settings] [Log Out] Roland Tritsch (AoC++) 17*
 \$year=2017; [Calendar] [Leaderboard] [Stats] [Sponsors]

--- Day 9: Stream Processing ---

A large stream blocks your path. According to the locals, it's not safe to cross the stream at the moment because it's full of garbage. You look down at the stream; rather than water, you discover that it's a stream of characters.

You sit for a while and record part of the stream (your puzzle input). The characters represent groups - sequences that begin with `[` and end with `]`. Within a group, there are zero or more other things, separated by commas: either another group or garbage. Since groups can contain other groups, a `]` only closes the most-recently-opened unclosed group - that is, they are nestable. Your puzzle input represents a single, large group which itself contains many smaller ones.

Sometimes, instead of a group, you will find garbage. Garbage begins with `<` and ends with `>`. Between those angle brackets, almost any character can appear, including `[` and `]`. Within garbage, `<` has no special meaning.

In a futile attempt to clean up the garbage, some program has canceled some of the characters within it using `!`: inside garbage, any character that comes after `!` should be ignored, including `<`, `>`, and even another `!`.

You don't see any characters that deviate from these rules. Outside garbage, you only find well-formed groups, and garbage always terminates according to the rules above.

Here are some self-contained pieces of garbage:

- `<>`, empty garbage.
- `<random characters>`, garbage containing random characters.
- `<<<<>`, because the extra `<` are ignored.
- `<{!}>`, because the first `>` is canceled.
- `<!!>`, because the second `!` is canceled, allowing the `>` to terminate the garbage.
- `<!!!!>`, because the second `!` and the first `>` are canceled.
- `<{o"!a,<{i<a>`, which ends at the first `>`.

Here are some examples of whole streams and the number of groups they contain:

- `[]`, 1 group.
- `[[[]]]`, 3 groups.
- `{{},{}}`, also 3 groups.
- `{{[],[],[]}}`, 6 groups.
- `<[],[],[]>`, 1 group (which itself contains garbage).
- `<a>,<a>,<a>,<a>`, 1 group.
- `{{<a>},{<a>},{<a>},{<a>}}`, 5 groups.
- `{{<!!>},{<!!>},{<!!>},{<a>}}`, 2 groups (since all but the last `>` are canceled).

Your goal is to find the total score for all groups in your input. Each group is assigned a score which is one more than the score of the group that immediately contains it. (The outermost group gets a score of 1.)

- `[]`, score of 1.
- `[[[]]]`, score of $1 + 2 + 3 = 6$.
- `{{},{}}`, score of $1 + 2 + 2 = 5$.
- `{{[],[],[]}}`, score of $1 + 2 + 3 + 3 + 3 + 4 = 16$.
- `<a>,<a>,<a>,<a>`, score of 1.
- `{{<ab>},{<ab>},{<ab>},{<ab>}}`, score of $1 + 2 + 2 + 2 + 2 = 9$.
- `{{<!!>},{<!!>},{<!!>},{<!!>}}`, score of $1 + 2 + 2 + 2 + 2 = 9$.
- `{{<a!>},{<a!>},{<a!>},{<ab>}}`, score of $1 + 2 = 3$.

What is the total score for all groups in your input?

Our sponsors help make Advent of Code possible:

SmartyStreets -
 U2VuZGluZyBDaH
 Jpc3RtYXMGY2Fy
 ZHMgdG8gYmFkIG
 FkZHJlc3Nlc38K

Your puzzle answer was .

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

--- Part Two ---

Now, you're ready to remove the garbage.

To prove you've removed it, you need to count all of the characters within the garbage. The leading and trailing `<` and `>` don't count, nor do any canceled characters or the `!` doing the canceling.

- `<>`, 0 characters.
- `<random characters>`, 17 characters.
- `<<<<>`, 3 characters.
- `<{!>}>`, 2 characters.
- `<!!>`, 0 characters.
- `<!!!!>`, 0 characters.
- `<{"i!a,<i<a>`, 10 characters.

How many non-canceled characters are within the garbage in your puzzle input?

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

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

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