Advent of Code [About] [AoC++] [Events] [Settings] [Log Out] peterpepo 24*
int y=2015; [Calendar] [Leaderboard] [Stats] [Sponsors]

--- Day 12: JSAbacusFramework.io ---

Santa's Accounting-Elves need help balancing the books after a recent order. Unfortunately, their accounting software uses a peculiar storage format. That's where you come in.

They have a JSON document which contains a variety of things: arrays (`[1,2,3]`), objects (`{"a":1, "b":2}`), numbers, and strings. Your first job is to simply find all of the numbers throughout the document and add them together.

For example:

- `[1,2,3]` and `{"a":2,"b":4}` both have a sum of `6`.
- `[[[3]]]` and `{"a":{"b":4},"c":-1}` both have a sum of `3`.
- `{"a":[-1,1]}` and `[-1,{"a":1}]` both have a sum of `0`.
- `[]` and `{}` both have a sum of `0`.

You will not encounter any strings containing numbers.

What is the sum of all numbers in the document?

Your puzzle answer was `156366`.

--- Part Two ---

Uh oh - the Accounting-Elves have realized that they double-counted everything red.

Ignore any object (and all of its children) which has any property with the value `"red"`. Do this only for objects (`{...}`), not arrays (`[...]`).

- `[1,2,3]` still has a sum of `6`.
- `[1,{"c":"red","b":2},3]` now has a sum of `4`, because the middle object is ignored.
- `{"d":"red","e":[1,2,3,4],"f":5}` now has a sum of `0`, because the entire structure is ignored.
- `[1,"red",5]` has a sum of `6`, because `"red"` in an array has no effect.

Your puzzle answer was `96852`.

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.