

--- Day 8: Memory Maneuver ---

The sleigh is much easier to pull than you'd expect for something its weight. Unfortunately, neither you nor the Elves know which way the North Pole is from here.

You check your wrist device for anything that might help. It seems to have some kind of navigation system! Activating the navigation system produces more bad news: "Failed to start navigation system. Could not read software license file."

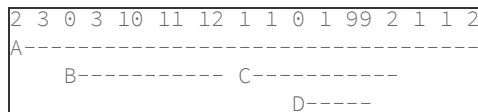
The navigation system's license file consists of a list of numbers (your puzzle input). The numbers define a data structure which, when processed, produces some kind of **tree** that can be used to calculate the license number.

The **tree** is made up of **nodes**; a single, outermost node forms the tree's **root**, and it contains all other nodes in the tree (or contains nodes that contain nodes, and so on).

Specifically, a node consists of:

- A **header**, which is always exactly two numbers:
 - The quantity of child nodes.
 - The quantity of metadata entries.
- Zero or more **child nodes** (as specified in the header).
- One or more **metadata entries** (as specified in the header).

Each child node is itself a node that has its own header, child nodes, and metadata. For example:



In this example, each node of the tree is also marked with an underline starting with a letter for easier identification. In it, there are four nodes:

- A, which has 2 child nodes (B, C) and 3 metadata entries (1, 1, 2).
- B, which has 0 child nodes and 3 metadata entries (10, 11, 12).
- C, which has 1 child node (D) and 1 metadata entry (2).
- D, which has 0 child nodes and 1 metadata entry (99).

The first check done on the license file is to simply add up all of the metadata entries. In this example, that sum is 1+1+2+10+11+12+2+99=138.

What is the sum of all metadata entries?

Your puzzle answer was 35852.

--- Part Two ---

The second check is slightly more complicated: you need to find the value of the root node (A in the example above).

The value of a node depends on whether it has child nodes.

If a node has no child nodes, its value is the sum of its metadata entries. So, the value of node B is 10+11+12=33, and the value of node D is 99.

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

[Catawiki](#) - Online auctions for special objects. Win prizes on our private leaderboard! And obviously, we're hiring!
bit.ly/join-cw

However, if a node does have child nodes, the metadata entries become indexes which refer to those child nodes. A metadata entry of `1` refers to the first child node, `2` to the second, `3` to the third, and so on. The value of this node is the sum of the values of the child nodes referenced by the metadata entries. If a referenced child node does not exist, that reference is skipped. A child node can be referenced multiple time and counts each time it is referenced. A metadata entry of `0` does not refer to any child node.

For example, again using the above nodes:

- Node `C` has one metadata entry, `2`. Because node `C` has only one child node, `2` references a child node which does not exist, and so the value of node `C` is `0`.
- Node `A` has three metadata entries: `1`, `1`, and `2`. The `1` references node `A`'s first child node, `B`, and the `2` references node `A`'s second child node, `C`. Because node `B` has a value of `33` and node `C` has a value of `0`, the value of node `A` is `33+33+0=66`.

So, in this example, the value of the root node is `66`.

What is the value of the root node?

Your puzzle answer was `33422`.

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.