--- Day 13: Distress Signal ---

You climb the hill and again try contacting the Elves. However, you instead receive a signal you weren't expecting: a distress signal.

Your handheld device must still not be working properly; the packets from the distress signal got decoded out of order. You'll need to re-order the list of received packets (your puzzle input) to decode the message.

Your list consists of pairs of packets; pairs are separated by a blank line. You need to identify how many pairs of packets are in the right order.

For example:

```
[1,1,3,1,1]
[1,1,5,1,1]

[[1],[2,3,4]]
[[1],4]

[9]
[[8,7,6]]

[[4,4],4,4]
[[4,4],4,4,4]

[7,7,7,7]
[7,7,7]

[]
[3]

[[[]]]
[[]]

[1,[2,[3,[4,[5,6,7]]]],8,9]
[1,[2,[3,[4,[5,6,0]]]],8,9]
```

Packet data consists of lists and integers. Each list starts with `[`, ends with `]`, and contains zero or more comma-separated values (either integers or other lists). Each packet is always a list and appears on its own line.

When comparing two values, the first value is called left and the second value is called right. Then:

- If both values are integers, the lower integer should come first. If the left integer is lower than the right integer, the inputs are in the right order. If the left integer is higher than the right integer, the inputs are not in the right order. Otherwise, the inputs are the same integer; continue checking the next part of the input.
- If both values are lists, compare the first value of each list, then the second value, and so on. If the left list runs out of items first, the inputs are in the right order. If the right list runs out of items first, the inputs are not in the right order. If the lists are the same length and no comparison makes a decision about the order, continue checking the next part of the input.
- If exactly one value is an integer, convert the integer to a list which contains that integer as its only value, then retry the comparison. For example, if comparing `[0,0,0]` and `2`, convert the right

value to `[2]` (a list containing `2`); the result is then found by instead comparing `[0,0,0]` and `[2]`.

Using these rules, you can determine which of the pairs in the example are in the right order:

```
== Pair 1 ==
- Compare [1,1,3,1,1] vs [1,1,5,1,1]
  - Compare 1 vs 1
  - Compare 1 vs 1
  - Compare 3 vs 5
    - Left side is smaller, so inputs are in the right order

== Pair 2 ==
- Compare [[1],[2,3,4]] vs [[1],4]
  - Compare [1] vs [1]
    - Compare 1 vs 1
  - Compare [2,3,4] vs 4
    - Mixed types; convert right to [4] and retry comparison
    - Compare [2,3,4] vs [4]
      - Compare 2 vs 4
        - Left side is smaller, so inputs are in the right order

== Pair 3 ==
- Compare [9] vs [[8,7,6]]
  - Compare 9 vs [8,7,6]
    - Mixed types; convert left to [9] and retry comparison
    - Compare [9] vs [8,7,6]
      - Compare 9 vs 8
        - Right side is smaller, so inputs are not in the right order

== Pair 4 ==
- Compare [[4,4],4,4] vs [[4,4],4,4,4]
  - Compare [4,4] vs [4,4]
    - Compare 4 vs 4
    - Compare 4 vs 4
  - Compare 4 vs 4
  - Compare 4 vs 4
  - Left side ran out of items, so inputs are in the right order

== Pair 5 ==
- Compare [7,7,7,7] vs [7,7,7]
  - Compare 7 vs 7
  - Compare 7 vs 7
  - Compare 7 vs 7
  - Right side ran out of items, so inputs are not in the right order

== Pair 6 ==
- Compare [] vs [3]
  - Left side ran out of items, so inputs are in the right order

== Pair 7 ==
- Compare [[[]]] vs [[]]
  - Compare [[]] vs []
    - Right side ran out of items, so inputs are not in the right order

== Pair 8 ==
- Compare [1,[2,[3,[4,[5,6,7]]]],8,9] vs [1,[2,[3,[4,[5,6,0]]]],8,9]
  - Compare 1 vs 1
  - Compare [2,[3,[4,[5,6,7]]]] vs [2,[3,[4,[5,6,0]]]]
    - Compare 2 vs 2
    - Compare [3,[4,[5,6,7]]] vs [3,[4,[5,6,0]]]
      - Compare 3 vs 3
      - Compare [4,[5,6,7]] vs [4,[5,6,0]]
        - Compare 4 vs 4
        - Compare [5,6,7] vs [5,6,0]
          - Compare 5 vs 5
          - Compare 6 vs 6
          - Compare 7 vs 0
            - Right side is smaller, so inputs are not in the right order
```

What are the indices of the pairs that are already in the right order? (The first pair has index 1, the second pair has index 2, and so on.) In the above example, the pairs in the right order are 1, 2, 4, and 6; the sum of these indices is 13.

Determine which pairs of packets are already in the right order. What is the sum of the indices of those pairs?

To begin, get your puzzle input.

Answer: [Submit]

You can also [Share] this puzzle.