--- Day 15: Lens Library ---

The newly-focused parabolic reflector dish is sending all of the collected light to a point on the side of yet another mountain - the largest mountain on Lava Island. As you approach the mountain, you find that the light is being collected by the wall of a large facility embedded in the mountainside.

You find a door under a large sign that says "Lava Production Facility" and next to a smaller sign that says "Danger - Personal Protective Equipment required beyond this point".

As you step inside, you are immediately greeted by a somewhat panicked reindeer wearing goggles and a loose-fitting hard hat. The reindeer leads you to a shelf of goggles and hard hats (you quickly find some that fit) and then further into the facility. At one point, you pass a button with a faint snout mark and the label "PUSH FOR HELP". No wonder you were loaded into that trebuchet so quickly!

You pass through a final set of doors surrounded with even more warning signs and into what must be the room that collects all of the light from outside. As you admire the large assortment of lenses available to further focus the light, the reindeer brings you a book titled "Initialization Manual".

"Hello!", the book cheerfully begins, apparently unaware of the concerned reindeer reading over your shoulder. "This procedure will let you bring the Lava Production Facility online - all without burning or melting anything unintended!"

"Before you begin, please be prepared to use the Holiday ASCII String Helper algorithm (appendix 1A)." You turn to appendix 1A. The reindeer leans closer with interest.

The HASH algorithm is a way to turn any string of characters into a single number in the range 0 to 255. To run the HASH algorithm on a string, start with a current value of 0. Then, for each character in the string starting from the beginning:

- Determine the ASCII code for the current character of the string.
- Increase the current value by the ASCII code you just determined.
- Set the current value to itself multiplied by 17.
- Set the current value to the remainder of dividing itself by 256.

After following these steps for each character in the string in order, the current value is the output of the HASH algorithm.

So, to find the result of running the HASH algorithm on the string HASH:

- The current value starts at 0.
- The first character is H; its ASCII code is 72.
- The current value increases to 72.
- The current value is multiplied by 17 to become 1224.
- The current value becomes 200 (the remainder of 1224 divided by 256).
- The next character is A; its ASCII code is 65.
- The current value increases to 265.
- The current value is multiplied by 17 to become 4505.
- The current value becomes 153 (the remainder of 4505 divided by 256).
- The next character is S; its ASCII code is 83.
- The current value increases to 236.
- The current value is multiplied by 17 to become 4012.
- The current value becomes 172 (the remainder of 4012 divided by 256).
- The next character is H; its ASCII code is 72.
- The current value increases to 244.
- The current value is multiplied by 17 to become 4148.
- The current value becomes 52 (the remainder of 4148 divided by 256).

So, the result of running the HASH algorithm on the string `HASH` is `52`.

The **initialization sequence** (your puzzle input) is a comma-separated list of steps to start the Lava Production Facility. **Ignore newline characters** when parsing the initialization sequence. To verify that your HASH algorithm is working, the book offers the sum of the result of running the HASH algorithm on each step in the initialization sequence.

For example:

```
rn=1,cm-,qp=3,cm=2,qp-,pc=4,ot=9,ab=5,pc-,pc=6,ot=7
```

This initialization sequence specifies 11 individual steps; the result of running the HASH algorithm on each of the steps is as follows:

- `rn=1` becomes `30`.
- `cm-` becomes `253`.
- `qp=3` becomes `97`.
- `cm=2` becomes `47`.
- `qp-` becomes `14`.
- `pc=4` becomes `180`.
- `ot=9` becomes `9`.
- `ab=5` becomes `197`.
- `pc-` becomes `48`.
- `pc=6` becomes `214`.
- `ot=7` becomes `231`.

In this example, the sum of these results is `1320`. Unfortunately, the reindeer has stolen the page containing the expected verification number and is currently running around the facility with it excitedly.

Run the HASH algorithm on each step in the initialization sequence. **What is the sum of the results?** (The initialization sequence is one long line; be careful when copy-pasting it.)

Your puzzle answer was `508498`.

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

--- Part Two ---

You convince the reindeer to bring you the page; the page confirms that your HASH algorithm is working.

The book goes on to describe a series of 256 **boxes** numbered `0` through `255`. The boxes are arranged in a line starting from the point where light enters the facility. The boxes have holes that allow light to pass from one box to the next all the way down the line.

```
      +-----+  +-----+         +-----+
Light | Box |  | Box |   ...   | Box |
----------------------------------------->
      |  0  |  |  1  |   ...   | 255 |
      +-----+  +-----+         +-----+
```

Inside each box, there are several **lens slots** that will keep a lens correctly positioned to focus light passing through the box. The side of each box has a panel that opens to allow you to insert or remove lenses as necessary.

Along the wall running parallel to the boxes is a large library containing lenses organized by **focal length** ranging from `1` through `9`. The reindeer also brings you a small handheld label printer.

The book goes on to explain how to perform each step in the initialization sequence, a process it calls the Holiday ASCII String Helper Manual Arrangement Procedure, or `HASHMAP` for short.

Each step begins with a sequence of letters that indicate the **label** of the lens on which the step operates. The result of running the HASH algorithm on the label indicates the correct box for that step.

The label will be immediately followed by a character that indicates the **operation** to perform: either an equals sign (`=`) or a dash (`-`).

If the operation character is a **dash** (`-`), go to the relevant box and remove the lens with the given label if it is present in the box. Then, move any remaining lenses as far forward in the box as they can go without changing their order, filling any space made by removing the indicated lens. (If no lens in that box has the given label, nothing happens.)

If the operation character is an **equals sign** (`=`), it will be followed by a number indicating the **focal length** of the lens that needs to go into the relevant box; be sure to use the label maker to mark the lens with the label given in the beginning of the step so you can find it later. There are two possible situations:

- If there is already a lens in the box with the same label, **replace the old lens** with the new lens: remove the old lens and put the new lens in its place, not moving any other lenses in the box.
- If there is **not** already a lens in the box with the same label, add the lens to the box immediately behind any lenses already in the box. Don't move any of the other lenses when you do this. If there aren't any lenses in the box, the new lens goes all the way to the front of the box.

Here is the contents of every box after each step in the example initialization sequence above:

```
After "rn=1":
Box 0: [rn 1]

After "cm-":
Box 0: [rn 1]

After "qp=3":
Box 0: [rn 1]
Box 1: [qp 3]

After "cm=2":
Box 0: [rn 1] [cm 2]
Box 1: [qp 3]

After "qp-":
Box 0: [rn 1] [cm 2]

After "pc=4":
Box 0: [rn 1] [cm 2]
Box 3: [pc 4]

After "ot=9":
Box 0: [rn 1] [cm 2]
Box 3: [pc 4] [ot 9]

After "ab=5":
Box 0: [rn 1] [cm 2]
Box 3: [pc 4] [ot 9] [ab 5]

After "pc-":
Box 0: [rn 1] [cm 2]
Box 3: [ot 9] [ab 5]

After "pc=6":
Box 0: [rn 1] [cm 2]
Box 3: [ot 9] [ab 5] [pc 6]

After "ot=7":
Box 0: [rn 1] [cm 2]
Box 3: [ot 7] [ab 5] [pc 6]
```

All 256 boxes are always present; only the boxes that contain any lenses are shown here. Within each box, lenses are listed from front to back; each lens is shown as its label and focal length in square brackets.

To confirm that all of the lenses are installed correctly, add up the **focusing power** of all of the lenses. The focusing power of a single lens is the result of multiplying together:

- One plus the box number of the lens in question.
- The slot number of the lens within the box: `1` for the first lens, `2` for the second lens, and so on.
- The focal length of the lens.

At the end of the above example, the focusing power of each lens is as follows:

- `rn`: `1` (box 0) * `1` (first slot) * `1` (focal length) = `1`
- `cm`: `1` (box 0) * `2` (second slot) * `2` (focal length) = `4`
- `ot`: `4` (box 3) * `1` (first slot) * `7` (focal length) = `28`
- `ab`: `4` (box 3) * `2` (second slot) * `5` (focal length) = `40`
- `pc`: `4` (box 3) * `3` (third slot) * `6` (focal length) = `72`

So, the above example ends up with a total focusing power of `145`.

With the help of an over-enthusiastic reindeer in a hard hat, follow the initialization sequence. **What is the focusing power of the resulting lens configuration?**

Answer: [Submit]

Although it hasn't changed, you can still get your puzzle input.

You can also [Share] this puzzle.