

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] mehaase 42*
 \$year=2018; [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

--- Day 1: Chronal Calibration ---

"We've detected some temporal anomalies," one of Santa's Elves at the Temporal Anomaly Research and Detection Instrument Station tells you. She sounded pretty worried when she called you down here. "At 500-year intervals into the past, someone has been changing Santa's history!"

"The good news is that the changes won't propagate to our time stream for another 25 days, and we have a device" - she attaches something to your wrist - "that will let you fix the changes with no such propagation delay. It's configured to send you 500 years further into the past every few days; that was the best we could do on such short notice."

"The bad news is that we are detecting roughly fifty anomalies throughout time; the device will indicate fixed anomalies with stars. The other bad news is that we only have one device and you're the best person for the job! Good lu--" She taps a button on the device and you suddenly feel like you're falling. To save Christmas, you need to get all fifty stars by December 25th.

Collect stars by solving puzzles. Two puzzles will be made available on each day in the advent calendar; the second puzzle is unlocked when you complete the first. Each puzzle grants one star. Good luck!

After feeling like you've been falling for a few minutes, you look at the device's tiny screen. "Error: Device must be calibrated before first use. Frequency drift detected. Cannot maintain destination lock." Below the message, the device shows a sequence of changes in frequency (your puzzle input). A value like +6 means the current frequency increases by 6; a value like -3 means the current frequency decreases by 3.

For example, if the device displays frequency changes of +1, -2, +3, +1, then starting from a frequency of zero, the following changes would occur:

- Current frequency 0, change of +1; resulting frequency 1.
- Current frequency 1, change of -2; resulting frequency -1.
- Current frequency -1, change of +3; resulting frequency 2.
- Current frequency 2, change of +1; resulting frequency 3.

In this example, the resulting frequency is 3.

Here are other example situations:

- +1, +1, +1 results in 3
- +1, +1, -2 results in 0
- -1, -2, -3 results in -6

Starting with a frequency of zero, what is the resulting frequency after all of the changes in frequency have been applied?

Your puzzle answer was 592.

--- Part Two ---

You notice that the device repeats the same frequency change list over and over. To calibrate the device, you need to find the first frequency it reaches twice.

For example, using the same list of changes above, the device would loop as follows:

- Current frequency 0, change of +1; resulting frequency 1.

Our sponsors help
make Advent of
Code possible:

.ai/io Domains -
Want a domain
like coffee.ai or
eth.io? Hundreds
for sale!

- Current frequency `1`, change of `-2`; resulting frequency `-1`.
- Current frequency `-1`, change of `+3`; resulting frequency `2`.
- Current frequency `2`, change of `+1`; resulting frequency `3`.
- (At this point, the device continues from the start of the list.)
- Current frequency `3`, change of `+1`; resulting frequency `4`.
- Current frequency `4`, change of `-2`; resulting frequency `2`, which has already been seen.

In this example, the first frequency reached twice is `2`. Note that your device might need to repeat its list of frequency changes many times before a duplicate frequency is found, and that duplicates might be found while in the middle of processing the list.

Here are other examples:

- `+1, -1` first reaches `0` twice.
- `+3, +3, +4, -2, -4` first reaches `10` twice.
- `-6, +3, +8, +5, -6` first reaches `5` twice.
- `+7, +7, -2, -7, -4` first reaches `14` twice.

What is the first frequency your device reaches twice?

Your puzzle answer was `241`.

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.