

[Advent of Code](#) [\[About\]](#) [\[Events\]](#) [\[Shop\]](#) [\[Log In\]](#)
[0x0000|2021](#) [\[Calendar\]](#) [\[AoC++\]](#) [\[Sponsors\]](#) [\[Leaderboard\]](#) [\[Stats\]](#)

--- Day 6: Lanternfish ---

The sea floor is getting steeper. Maybe the sleigh keys got carried this way?

A massive school of glowing `lanternfish` swims past. They must spawn quickly to reach such large numbers - maybe exponentially quickly? You should model their growth rate to be sure.

Although you know nothing about this specific species of lanternfish, you make some guesses about their attributes. Surely, each lanternfish creates a new lanternfish once every 7 days.

However, this process isn't necessarily synchronized between every lanternfish - one lanternfish might have 2 days left until it creates another lanternfish, while another might have 4. So, you can model each fish as a single number that represents the number of days until it creates a new lanternfish.

Furthermore, you reason, a new lanternfish would surely need slightly longer before it's capable of producing more lanternfish: two more days for its first cycle.

So, suppose you have a lanternfish with an internal timer value of `3`:

- After one day, its internal timer would become `2`.
- After another day, its internal timer would become `1`.
- After another day, its internal timer would become `0`.
- After another day, its internal timer would reset to `6`, and it would create a new lanternfish with an internal timer of `8`.
- After another day, the first lanternfish would have an internal timer of `5`, and the second lanternfish would have an internal timer of `7`.

A lanternfish that creates a new fish resets its timer to `6`, not `7` (because `0` is included as a valid timer value). The new lanternfish starts with an internal timer of `8` and does not start counting down until the next day.

Realizing what you're trying to do, the submarine automatically produces a list of the ages of several hundred nearby lanternfish (your puzzle input). For example, suppose you were given the following list:

`3,4,3,1,2`

This list means that the first fish has an internal timer of `3`, the second fish has an internal timer of `4`, and so on until the fifth fish, which has an internal timer of `2`. Simulating these fish over several days would proceed as follows:

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

[Catawiki](#) - Online auctions for special objects.
We're hiring!
bit.ly/join-cw

```
Initial state: 3,4,3,1,2
After 1 day: 2,3,2,0,1
After 2 days: 1,2,1,6,0,8
After 3 days: 0,1,0,5,6,7,8
After 4 days: 6,0,6,4,5,6,7,8,8
After 5 days: 5,6,5,3,4,5,6,7,7,8
After 6 days: 4,5,4,2,3,4,5,6,6,7
After 7 days: 3,4,3,1,2,3,4,5,5,6
After 8 days: 2,3,2,0,1,2,3,4,4,5
After 9 days: 1,2,1,6,0,1,2,3,3,4,8
After 10 days: 0,1,0,5,6,0,1,2,2,3,7,8
After 11 days: 6,0,6,4,5,6,0,1,1,2,6,7,8,8,8
After 12 days: 5,6,5,3,4,5,6,0,0,1,5,6,7,7,7,8,8
After 13 days: 4,5,4,2,3,4,5,6,6,0,4,5,6,6,6,7,7,8,8
After 14 days: 3,4,3,1,2,3,4,5,5,6,3,4,5,5,5,6,6,7,7,8
After 15 days: 2,3,2,0,1,2,3,4,4,5,2,3,4,4,4,5,5,6,6,7
After 16 days: 1,2,1,6,0,1,2,3,3,4,1,2,3,3,3,4,4,5,5,6,8
After 17 days: 0,1,0,5,6,0,1,2,2,3,0,1,2,2,2,3,3,4,4,5,7,8
After 18 days: 6,0,6,4,5,6,0,1,1,2,6,0,1,1,1,2,2,3,3,4,6,7,8,8,8,8
```

Each day, a `0` becomes a `6` and adds a new `8` to the end of the list, while each other number decreases by 1 if it was present at the start of the day.

In this example, after 18 days, there are a total of `26` fish. After 80 days, there would be a total of `5934`.

Find a way to simulate lanternfish. How many lanternfish would there be after 80 days?

To play, please identify yourself via one of these services:

[\[GitHub\]](#) [\[Google\]](#) [\[Twitter\]](#) [\[Reddit\]](#) - [\[How Does Auth Work?\]](#)