

[Advent of Code](#) [\[About\]](#) [\[Events\]](#) [\[Shop\]](#) [\[Settings\]](#) [\[Log Out\]](#) (anonymous user #1056941) 14
[//2021](#) [\[Calendar\]](#) [\[AoC++\]](#) [\[Sponsors\]](#) [\[Leaderboard\]](#) [\[Stats\]](#)

--- Day 7: The Treachery of Whales ---

A giant **whale** has decided your submarine is its next meal, and it's much faster than you are. There's nowhere to run!

Suddenly, a swarm of crabs (each in its own tiny submarine - it's too deep for them otherwise) zooms in to rescue you! They seem to be preparing to blast a hole in the ocean floor; sensors indicate a **massive underground cave system** just beyond where they're aiming!

The crab submarines all need to be aligned before they'll have enough power to blast a large enough hole for your submarine to get through. However, it doesn't look like they'll be aligned before the whale catches you! Maybe you can help?

There's one major catch - crab submarines can only move horizontally.

You quickly make a list of the **horizontal position of each crab** (your puzzle input). Crab submarines have limited fuel, so you need to find a way to make all of their horizontal positions match while requiring them to spend as little fuel as possible.

For example, consider the following horizontal positions:

```
16,1,2,0,4,2,7,1,2,14
```

This means there's a crab with horizontal position **16**, a crab with horizontal position **1**, and so on.

Each change of 1 step in horizontal position of a single crab costs 1 fuel. You could choose any horizontal position to align them all on, but the one that costs the least fuel is horizontal position **2**:

- Move from **16** to **2**: **14** fuel
- Move from **1** to **2**: **1** fuel
- Move from **2** to **2**: **0** fuel
- Move from **0** to **2**: **2** fuel
- Move from **4** to **2**: **2** fuel
- Move from **2** to **2**: **0** fuel
- Move from **7** to **2**: **5** fuel
- Move from **1** to **2**: **1** fuel
- Move from **2** to **2**: **0** fuel
- Move from **14** to **2**: **12** fuel

This costs a total of **37** fuel. This is the cheapest possible outcome; more expensive outcomes include aligning at position **1** (**41** fuel), position **3** (**39** fuel), or position **10** (**71** fuel).

Determine the horizontal position that the crabs can align to using the least fuel possible. **How much fuel must they spend to align to that position?**

Your puzzle answer was **355989**.

--- Part Two ---

The crabs don't seem interested in your proposed solution. Perhaps you misunderstand crab engineering?

As it turns out, crab submarine engines don't burn fuel at a constant rate. Instead, each change of 1 step in horizontal position costs 1 more unit of fuel than the last: the first step costs **1**, the second step costs **2**, the third step costs **3**, and so on.

Our **sponsors** help make Advent of Code possible:

Honeycomb - You like performing correct code. So do we. Distributed systems should be easy to understand. Use Honeycomb for free to debug your distributed systems and get a free shirt. Download our white papers and watch our demo.

As each crab moves, moving further becomes more expensive. This changes the best horizontal position to align them all on; in the example above, this becomes 5:

- Move from 16 to 5: 66 fuel
- Move from 1 to 5: 10 fuel
- Move from 2 to 5: 6 fuel
- Move from 0 to 5: 15 fuel
- Move from 4 to 5: 1 fuel
- Move from 2 to 5: 6 fuel
- Move from 7 to 5: 3 fuel
- Move from 1 to 5: 10 fuel
- Move from 2 to 5: 6 fuel
- Move from 14 to 5: 45 fuel

This costs a total of 168 fuel. This is the new cheapest possible outcome; the old alignment position (2) now costs 206 fuel instead.

Determine the horizontal position that the crabs can align to using the least fuel possible so they can make you an escape route! How much fuel must they spend to align to that position?

Your puzzle answer was 102245489.

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.