

Advent of Code 2025

Jump to problem

Advent of Code 2025	1
Day 1	2
Part 1	2
Part 2	2

Day 1

Today was fairly difficult as far as Day 1's go (or maybe I was tired from the 4 hour drive through a rainstorm I had just hours earlier). As in the past, I did the problems in Kotlin.

Part 1

I started part 1 off wonderfully, misreading it, and (attempting to) solve part 2. When I was getting a higher answer than expected, I went back, reread, and stashed my code for later, (correctly) assuming it may be useful for part 2. Other than that, my code wasn't that interesting, just a check for zero, and repeatedly incrementing my dial variable by 100 until it was positive, since modulo arithmetic wouldn't help me there.

Total time: **00:06:03**

Part 2

Part 2 is where I really began to struggle. At first, I attempted the "smarter" solution. It looked something like this (unfortunately my 12:30am brain decided not to save it for when I want to come back later to fix it):

```
dial += value * direction
if (dial == 0) {
    zeroCounter++
}

while (dial < 100) {
    dial += 100
    zeroCounter++
}

while (dial > 100) {
    dial -= 100
    zeroCounter++
}
```

The main issue (at least as far as I could tell at the time) was my code would "double count" the dial if it was initially zero before an instruction, and moved to the left. I added a check to cover this case, but it didn't seem to work. This solution is definitely one I would like to revisit after the calendar is over, as it seems like one that could be optimized heavily.

After this attempt, I moved onto the "brute-force" method. I ran a loop from 1 to the instruction's "value", counting every time the dial passed zero, and ensuring it stayed within the range required. This ended up working, and giving me the correct solution after many incorrect submissions.

Total time: **00:38:10**