# Advent of Code

December 2025

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | **3** | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | | | | |

## Answers

### Part 1

Executed on the example input.
- Expected answer: **357**
- Computed answer: **357**

This seems correct

---

Executed on the evaluation input.
- Computed answer: **17412**

### Part 2

Executed on the example input.
- Expected answer: **3121910778619**
- Computed answer: **3121910778619**

This seems correct

---

Executed on the evaluation input.
- Computed answer: **172681562473501**

# Logs

## Parsing

### Part 1

```
(
  (9, 8, 7, 6, 5, 4, 3, 2, 1, 1, 1, 1, 1, 1, 1),
  (8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 9),
  (2, 3, 4, 2, 3, 4, 2, 3, 4, 2, 3, 4, 2, 7, 8),
  (8, 1, 8, 1, 8, 1, 9, 1, 1, 1, 1, 2, 1, 1, 1),
)
```

### Part 2

(Same)

## Part 1

## Part 2

```
Battery bank: 987654321111111
biggest value remaining in 987 is 9 at 0
biggest value remaining in 876 is 8 at 0
biggest value remaining in 765 is 7 at 0
biggest value remaining in 654 is 6 at 0
biggest value remaining in 543 is 5 at 0
biggest value remaining in 432 is 4 at 0
biggest value remaining in 321 is 3 at 0
biggest value remaining in 211 is 2 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
Produces 987654321111
Battery bank: 811111111111119
biggest value remaining in 811 is 8 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
```

```
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 1 at 0
biggest value remaining in 111 is 9 at 3
Produces 811111111119
Battery bank: 234234234234278
biggest value remaining in 234 is 4 at 2
biggest value remaining in 2 is 3 at 1
biggest value remaining in is 4 at 0
biggest value remaining in is 2 at 0
biggest value remaining in is 3 at 0
biggest value remaining in is 4 at 0
biggest value remaining in is 2 at 0
biggest value remaining in is 3 at 0
biggest value remaining in is 4 at 0
biggest value remaining in is 2 at 0
biggest value remaining in is 7 at 0
biggest value remaining in is 8 at 0
Produces 434234234278
Battery bank: 818181911112111
biggest value remaining in 818 is 8 at 0
biggest value remaining in 181 is 8 at 1
biggest value remaining in 18 is 8 at 1
biggest value remaining in 1 is 9 at 1
biggest value remaining in is 1 at 0
biggest value remaining in is 1 at 0
biggest value remaining in is 1 at 0
biggest value remaining in is 1 at 0
biggest value remaining in is 2 at 0
biggest value remaining in is 1 at 0
biggest value remaining in is 1 at 0
biggest value remaining in is 1 at 0
Produces 888911112111
```

# Source code

## Preliminaries

```
#import "/template/aot.typ"

#show: aot.format

#aot.parser(input => {
  input.trim("\n").split("\n").map(s => {
    s.split("").filter(s => s != "").map(int)
  })
})
```

## Part 1

```
#aot.solve(input => {
  let total = 0
  for bank in input {
    let i1 = aot.utils.argmax(bank.slice(0, -1))
    let i2 = aot.utils.argmax(bank.slice(i1 + 1))
    let v1 = bank.at(i1)
    let v2 = bank.at(i1 + i2 + 1)
    total += v1 * 10 + v2
  }
  aot.answer(total)
})
```

## Part 2

```
#aot.solve(input => {
  let total = 0
  for bank in input {
    let jolts = 0
    for i in range(12) {
      let end-idx = if i < 11 { i - 11 } else { none }
      let idx = aot.utils.argmax(bank.slice(0, end-idx))
      let val = bank.at(idx)
      jolts = jolts * 10 + val
      bank = bank.slice(idx + 1)
    }
    total += jolts
  }
  aot.answer(total)
})
```